

“©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

# Fuzzy Multiple-source Transfer Learning

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

**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 systems and particularly fuzzy rule-based models, was developed due to its capacity to deal with uncertainty. However, one issue with fuzzy transfer learning, even in the area of general transfer learning, has not been resolved: how to combine and then use knowledge when multiple source domains are available. This study presents new methods for merging fuzzy rules from multiple domains for regression tasks. Two different settings are separately explored: homogeneous and heterogeneous space. In homogeneous situations, knowledge from the source domains is merged in the form of fuzzy rules. In heterogeneous situations, knowledge is merged in the form of both data and fuzzy rules. Experiments on both synthetic and real-world datasets provide insights into the scope of applications suitable for the proposed methods and validate their effectiveness through comparisons with other state-of-the-art transfer learning methods. An analysis of parameter sensitivity is also included.

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

## I. INTRODUCTION

Machine learning [1] has deeply affected the great achievements gained in many areas of data science, including computer vision [2], biology [3], medical imaging [4], and business management [5]. However, fundamentally, many well-known machine-learning algorithms, such as neural networks, SVM, and Bayesian network, are supervised processes, which means the performance and generalizability of the resulting models tend 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 train a model properly is difficult, even impossible. Without enough labeled data, the accuracy and generalizability of a model suffers. Thus, transfer learning [6] has emerged as a potential solution.

Transfer learning, in general, addresses the problem of how to leverage previously acquired knowledge to improve the efficiency and accuracy of learning in one domain that in some way relates to the original domain. The first survey, which is state-of-the-art, on transfer learning [6] provides important definitions in transfer learning. As part of this review, transfer learning studies are categorized into multi-task learning [7], domain adaptation [8], and cross-domain learning [9].

However, as this area has attracted many researchers, more methods are being developed to handle transfer learning problems, and survey papers are beginning to focus on precise areas, e.g., visualization [10], reinforcement learning [11], activity recognition [12], computational intelligence [13], and collaborative recommendation [14]. The current applications for transfer learning techniques are extensive – from image processing [15] to text categorization [16] to natural language processing [17] to fault diagnosis [18] and beyond.

Yet, while these existing methods have had some success in handling domain adaptation issues, most ignore the inherent phenomenon of uncertainty – a crucial factor during the knowledge transfer process [19]. There is a clear co-dependency between the level of certainty in learning a task and the amount of information that is available. Problems with too little information have a high degree of uncertainty. If there are too few data with labels in the target domain, only a finite amount of information can be extracted, and this leads to a high degree of uncertainty. However, the emergence of fuzzy systems has shown promising results in overcoming this problem [20].

The integration of fuzzy logic with transfer learning has drawn considerable attention in the literature. For example, researchers have applied fuzzy sets to represent linguistic variables when feature values cannot be precisely described numerically, while fuzzy distances assist the retrieval of similar cases [21]. Transferring implicit and explicit knowledge from similar domains is hidden and uncertain by nature, so using fuzzy logic and fuzzy rule theory to handle the associated vagueness and uncertainty is apt and can improve transfer accuracy. Hence, many scholars have turned to fuzzy systems as a solution to transfer learning problems with promising results. Deng et al. [22, 23] proposed a series of transfer learning methods using a Mamdani-Larsen-type fuzzy system and a Takagi-Sugeno-Kang (TSK) fuzzy model coupled with novel fuzzy logic algorithms that include definitions for two new objective functions. Further, they applied their methods to scenarios with insufficient data, such as recognizing electroencephalogram signals in environments with a data shortage. Behbood et al. [24, 25] proposed a fuzzy-based transfer learning approach to long-term bank failure prediction models with source and target domains that have different data distributions. Liu et al. [26] focused on unsupervised heterogeneous domain adaptation problems, presenting a novel transfer learning model that incorporates n-dimensional fuzzy

geometry and fuzzy equivalence relations. A metric based on  $n$ -dimensional fuzzy geometry is defined to measure the similarity of features between domains. Shared fuzzy equivalence relations then force the same number of clustering categories given the same value of  $\alpha$ , which means knowledge can be transferred from the source domain to the target domain in heterogeneous space through the clustering categories.

Despite these advancements in fuzzy system-based transfer learning methods, there is still the main issue that has not been solved: how to merge and transfer knowledge when multiple source domains are available. This case is quite common in the real world. For example, a company needs to determine the price for a new type of computer entering the Australian market but has very little data on consumer responses to the product. However, data for two other types of computers sold in Australia are available. So, how might these two datasets (source domains) be used to support the pricing decision at hand (target domain)?

There have already been some studies on multiple-source domain adaptation problems. Yao et al. [27] proposed two new algorithms, MultiSource-TrAdaBoost and TaskTraAdaBoost, which extend the boosting framework for transferring knowledge from multiple sources. These algorithms reduce negative transfers by increasing the number of sources. Tan et al. [28] presented a novel algorithm to leverage knowledge from different views and sources collaboratively by letting different views from different sources complement each other through a co-training style framework to reduce the differences in distribution across different domains. Beyond transferring the source data, Zhuang et al. [29] discovered a more powerful feature representation of the data when transferring knowledge from multiple source domains to the target domain. Here, autoencoders are used to construct a feature mapping from an original instance to a hidden representation, and multiple classifiers from the source domain data are jointly trained to learn the hidden representation and classifiers simultaneously. However, these approaches were developed for classification tasks and, thus far, combining knowledge from multiple sources cannot be translated into fuzzy systems, which are superior at handling uncertainty in domain adaptation problems.

Some of our own previous research has focused on developing the domain adaptation ability of fuzzy rule-based models with regression tasks [30, 31]. We proposed a set of algorithms for two different scenarios, where the datasets for the source domain and target domain were homogeneous [32] and heterogeneous [33]. In this paper, we explore the ability of fuzzy systems to deal with transfer learning problems when multiple source domains are available based on these previous works.

The specific contribution of this paper is to advance the domain adaptation ability of fuzzy rule-based systems in multiple-source environments for regression tasks. The current transfer learning methods cannot deal with the regression tasks with multiple sources. In principle, sometimes a single source transfer is better than multi-domains and sometimes multi-domain transfer is better than single domain, which determines the “similarity” between source domains and the target domain.

This paper, in fact, aims to identify which source domains(s) are more suitable than others to transfer knowledge to a given target domain. We propose two algorithms to handle knowledge transfer from multiple source domains to one target domain for regression tasks, and the use of fuzzy systems confers the model the capacity to handle uncertainty in an information insufficient environment and improves the prediction accuracy.

The remainder of this paper is structured as follows. Section II presents the preliminaries of this work, including some important definitions in transfer learning, and the main prediction model applied, i.e., the Takagi-Sugeno fuzzy model. Section III presents a 4-step algorithm for the domain adaptation process using multiple source domains in homogeneous space. Section IV presents an algorithm for multiple source domain knowledge transfer in heterogeneous situations, which includes two approaches implemented simultaneously – one with 4 steps and one with 5 steps. Sections V and VI present the validation tests of the two proposed algorithms using both 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 (TS) fuzzy model, which is the basic prediction model used in our multiple-source domain adaptation method.

### A. Definitions

Definition 1 (Domain) [6]: A domain is denoted as  $D = \{F, P(X)\}$ , where  $F$  is a feature space, and  $P(X)$ ,  $X = \{x_1, \dots, x_n\}$  are the probability distributions of the instances.

Definition 2 (Task) [6]: A task is denoted as  $T = \{Y, f(\cdot)\}$ , where  $Y \in R$  is the output, and  $f(\cdot)$  is an objective predictive function.

Definition 3 (Transfer Learning) [6]: 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 (TS) Fuzzy Model

A fuzzy system, in this case a TS model, comprises a set of IF-THEN fuzzy rules in the following form:

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

where  $\mathbf{v}_i$  are the prototypes, and  $\mathbf{a}_i$  are the coefficients of the linear function.

This TS fuzzy model is built using a set of instances  $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$  using a sequence of two procedures [34]: from the conditions  $A_1, \dots, A_c$  through fuzzy clustering, and from the optimized parameters of the linear functions  $L_i(\mathbf{x}, \mathbf{a}_i)$ .

The TS fuzzy model could also be rewritten in the form of a neural network with the structure illustrated in Fig.1. The first layer represents the input data, and each neuron in the second layer represents a cluster, which also represents the condition of a fuzzy rule. The third layer contains the corresponding consequences of the fuzzy rules, i.e., the output.

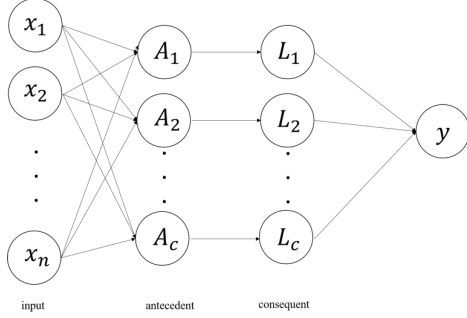


Fig. 1. The TS fuzzy model in a neural network structure

The output of the TS fuzzy model is calculated by

$$y = \sum_{i=1}^c \frac{A_i(\mathbf{x}, \mathbf{v}_i)}{\sum_{j=1}^c A_j(\mathbf{x}, \mathbf{v}_j)} \cdot L_i(\mathbf{x}, \mathbf{a}_i) \quad (2)$$

### III. FUZZY TRANSFER LEARNING USING MULTIPLE SOURCES IN HOMOGENEOUS SPACE

This section presents a method for transferring knowledge from multiple source domains to the target domain in homogeneous space. The multiple-domain adaptation problem with a fuzzy rule-based model is outlined first with formulas. And the specific challenge with implementing knowledge transfer in such cases is described. Second, the procedures in the proposed method are described in detail.

#### A. Problem Statement

Consider there are  $h$  source domains with large amounts of labeled data and a target domain with very little labeled data. The datasets in multiple source domains are denoted as  $\mathcal{S}_1, \dots, \mathcal{S}_h$ :

$$\begin{aligned} \mathcal{S}_1 &= \{(\mathbf{x}_1^{s1}, y_1^{s1}), \dots, (\mathbf{x}_{N_{s1}}^{s1}, y_{N_{s1}}^{s1})\} \\ &\dots \\ \mathcal{S}_h &= \{(\mathbf{x}_1^{sh}, y_1^{sh}), \dots, (\mathbf{x}_{N_{sh}}^{sh}, y_{N_{sh}}^{sh})\} \end{aligned} \quad (3)$$

$(\mathbf{x}_k^{sj}, y_k^{sj})$  is the  $k$ th input-output data pairs in the  $j$ th source domain, where  $\mathbf{x}_k^{sj} \in R^n$  ( $k = 1, \dots, N_{sj}, j = 1, \dots, h$ ) is an  $n$ -dimensional input variable, the label  $y_k^{sj} \in R$  is a continuous variable, and  $N_{sj}$  indicates the number of data pairs.

The dataset in the target domain  $\mathcal{T}$  consists of two subsets: one with labels and one without:

$$\mathcal{T} = \{\mathcal{T}_L, \mathcal{T}_U\} = \{(\mathbf{x}_1^t, y_1^t), \dots, (\mathbf{x}_{N_{t1}}^t, y_{N_{t1}}^t)\}, \{\mathbf{x}_{N_{t1}+1}^t, \dots, \mathbf{x}_{N_t}^t\} \quad (4)$$

where  $\mathbf{x}_k^t \in R^n$  ( $k = 1, \dots, N_t$ ) is the  $n$ -dimensional input variable,  $y_k^t \in R$  is a label that is only accessible for the first  $N_{t1}$  data.  $\mathcal{T}_L$  contains the instances with labels, and  $\mathcal{T}_U$  contains the instances without labels. The numbers of instances in  $\mathcal{T}_L$

and  $\mathcal{T}_U$  are  $N_{t1}$  and  $N_t - N_{t1}$ , respectively, and satisfy  $N_{t1} \ll N_t, N_{t1} \ll N_{s1}, \dots, N_{t1} \ll N_{sh}$ .

In each source domain, a well-performed prediction model could be built since there is a sufficient amount of labeled data and, hence, corresponding sets of fuzzy rules can be obtained.

In homogeneous space, the dimensionality of the input space in the  $h$  source domains and the target domain are the same, and the input variables have exactly the same meanings. The datasets  $\mathcal{S}_1, \dots, \mathcal{S}_h$ , and  $\mathcal{T}$  are distinguished from each other due to discrepancies in their distributions. Therefore, the rules in the source domains cannot be directly used to solve prediction problems in the target domain.

Many general transfer learning methods are easily able to solve single-domain transfer problems, but there are two main issues when using fuzzy rule-based models to solve problems that involve multiple source domains. First, brute force methods that combine all the rules in the source domains lead to redundancy. Additionally, accumulating all the rules increases the number of parameters to optimize, which in turn increases the computational complexity. However, a fuzzy rule-based multiple-source transfer learning method can overcome these problems. The details are proposed in the next section.

#### B. Multiple-source Domain Adaptation in Homogeneous Space

The method of transferring knowledge from multiple source domains to a target domain can be summarized in four steps:

**Step 1:** Combine all the rules in the source domains.

Given  $h$  source domains  $\mathcal{S}_1, \dots, \mathcal{S}_h$ ,  $h$  sets of fuzzy rules would be obtained, denoted as  $R^{s1}, \dots, R^{sh}$ :

$$\begin{aligned} R^{s1} &= \{r(\mathbf{v}_1^{s1}, \mathbf{a}_1^{s1}), r(\mathbf{v}_2^{s1}, \mathbf{a}_2^{s1}), \dots, r(\mathbf{v}_{c1}^{s1}, \mathbf{a}_{c1}^{s1})\} \\ &\dots \\ R^{sh} &= \{r(\mathbf{v}_1^{sh}, \mathbf{a}_1^{sh}), r(\mathbf{v}_2^{sh}, \mathbf{a}_2^{sh}), \dots, r(\mathbf{v}_{ch}^{sh}, \mathbf{a}_{ch}^{sh})\} \end{aligned} \quad (5)$$

where  $r(\mathbf{v}_i^{sj}, \mathbf{a}_i^{sj})$ , ( $i = 1, \dots, cj$ ), represents a rule in the source domain  $\mathcal{S}_j$ ,  $j = 1, \dots, h$ ,  $\mathbf{v}_k^{sj}$  is the prototype, i.e., the center of the data clusters, and  $\mathbf{a}_k^{sj}$  are the coefficients of the corresponding linear functions. The rule  $r(\mathbf{v}_k^{sj}, \mathbf{a}_k^{sj})$  is represented as:

$$\text{If } \mathbf{x}_k^{sj} \text{ is } A_i(\mathbf{x}_k^{sj}, \mathbf{v}_i^{sj}), \text{ then } y_k^{sj} \text{ is } L_i(\mathbf{x}_k^{sj}, \mathbf{a}_k^{sj}) \quad i = 1, \dots, cj \quad (6)$$

In homogeneous cases, there is an assumption that the source data is not accessible after constructing the model, and only the rules are available. This could preserve the privacy of the source data.

The rules in  $\mathcal{S}_1, \dots, \mathcal{S}_h$  are combined and denoted as  $R^s$ :

$$R^s = \{r(\mathbf{v}_1^{s1}, \mathbf{a}_1^{s1}), \dots, r(\mathbf{v}_{c1}^{s1}, \mathbf{a}_{c1}^{s1}), \dots, r(\mathbf{v}_1^{sh}, \mathbf{a}_1^{sh}), \dots, r(\mathbf{v}_{ch}^{sh}, \mathbf{a}_{ch}^{sh})\} \quad (7)$$

which can be rewritten as:

$$R^s = \{r(\mathbf{v}_1^s, \mathbf{a}_1^s), r(\mathbf{v}_2^s, \mathbf{a}_2^s), \dots, r(\mathbf{v}_{cs}^s, \mathbf{a}_{cs}^s)\} \quad (8)$$

where  $cs = c1 + \dots + ch$ .

Due to the different distributions of the source and target data, the rules in  $R^s$  will have poor prediction accuracy for the target data.

**Step 2:** Determine the number of fuzzy rules in the target domain.

To effectively use the rules in the source domain, it is important if not crucial to determine the number of clusters or the number of fuzzy rules in target domain to inform how many (and which) rules to select for the target domain in the next step. Here, an infinite Gaussian mixture model (IGMM) is used to explore the data structure of the target domain and determine the number of fuzzy rules.

IGMM simulates the distribution of the target data by using  $\{\mathbf{x}_1^t, \dots, \mathbf{x}_{N_t}^t\}$  in an unsupervised learning manner. Fig. 2 illustrates the probability of finding 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 the dataset has been 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).

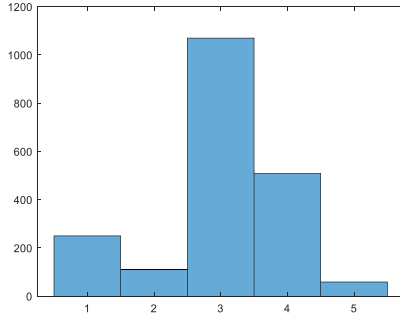


Fig. 2. Example of the results for IGMM

Applying the technique IGMM, the number of rules in the target domain is determined based on  $\{\mathbf{x}_1^t, \dots, \mathbf{x}_{N_t}^t\}$ , and denoted as  $ct$ . Additionally, this step lays the basis for the next step where the number of clusters needs to be provided in advance.

The unlabeled target data are used in this step, and the labeled target data are used in Step 4 to optimize the mappings by modifying the existing rules.

**Step 3:** Select the appropriate rules for the target domain.

How the source rules in  $R^s$  are merged is the key step in implementing domain adaptation with multiple sources. We have adopted the method of selecting the most appropriate rules from  $R^s$  based on the center of the clusters in the target data.

First, fuzzy C-means is applied to  $\{\mathbf{x}_1^t, \dots, \mathbf{x}_{N_t}^t\}$  to find the center of the clusters in the target data, denoted as:

$$\mathbf{v}_1^t, \mathbf{v}_2^t, \dots, \mathbf{v}_{ct}^t \quad (9)$$

Then based on the obtained centers, the  $ct$  rules that satisfy

$$R^t = \{r(\mathbf{v}_i^s, \mathbf{a}_i^s) \in R^s | \text{dist}(\mathbf{v}_i^s, \mathbf{v}_k^t) \leq \text{dist}(\mathbf{v}_j^s, \mathbf{v}_k^t), \forall j \in \{1, \dots, cs\}, k = 1, \dots, ct\} \quad (10)$$

are selected from  $R^s$ .

The distances between each element in  $\{\mathbf{v}_1^t, \mathbf{v}_2^t, \dots, \mathbf{v}_{ct}^t\}$  and all the centers in the source domains  $\{\mathbf{v}_1^s, \mathbf{v}_2^s, \dots, \mathbf{v}_{cs}^s\}$  are measured, and the corresponding rule with the smallest distance is selected. Therefore, the  $ct$  rules in  $R^s$  for the target domain are selected after calculating the distances for all the elements in  $\{\mathbf{v}_1^t, \mathbf{v}_2^t, \dots, \mathbf{v}_{ct}^t\}$ . For simplicity, assume the first few  $ct$  rules in  $R^s$  are the ones selected, denoted as:

$$R^t = \{R(\mathbf{v}_1^s, \mathbf{a}_1^s), \dots, R(\mathbf{v}_{ct}^s, \mathbf{a}_{ct}^s)\} \quad (11)$$

**Step 4:** Modify the selected rules to fit the target data.

The selected rules in  $R^t$  cannot be used to solve the regression tasks in the target domain because they have different data distributions. Thus, some techniques we presented in a previous paper [32] are used to modify the fuzzy rules by changing the input/output spaces through mappings. Fig.3 shows the modifications to the TS model.

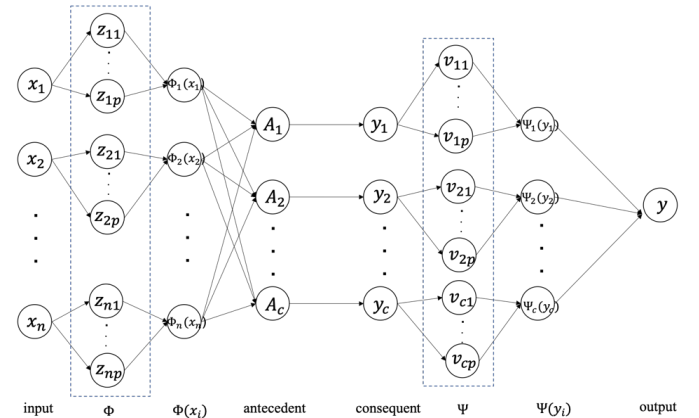


Fig. 3. Modified TS model with changed inputs and outputs

Comparing the structure of the TS models shown in Fig.1 versus Fig. 3, two modifications have been made, i.e., the input space and output space use the mappings  $\Phi$  and  $\Psi$ , respectively, for the dotted lines.

The idea of changing the input space is supported by the notion that each input variable is assumed to be determined by some hidden compared features. As such, the different distributions of the input variables in the two domains must be due to either different hidden features or different weights of those features. Therefore, changing the input variables effectively adjusts the number and weight of these hidden features, so the input distribution is more compatible with the target data.

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 a TS 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 not feasible for regression tasks where only unlabeled data are available. Thus, changing the output space as a consequence of the fuzzy rules in regression tasks is both essential and effective.

The method to construct mappings for the input and output space are the same. The nonlinear functions are used to build the mappings. The nonlinear function is constructed through a network that is composed of  $P$  nodes in the hidden layer and a single node in the output layer. The transformation of the  $j$ th input variable of data  $\mathbf{x}_k^t$  is shown in Fig. 4 as an example of the nonlinear mapping for each input variable.

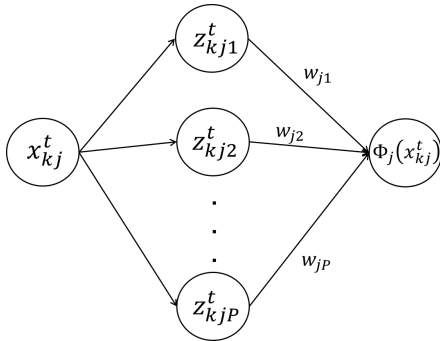


Fig. 4. Nonlinear mappings structure

The active functions of the nodes in the hidden layer are sigmoid functions, which are dominated by two parameters. Therefore, as shown in Fig. 4, the graphical representation of the transformed  $j$ th input variable of data  $\mathbf{x}_k^t$  is:

$$\Phi_j(\mathbf{x}_{kj}^t) = \sum_{p=1}^P w_{jp} * z_{kj}^t \quad (12)$$

where  $w_{jp}$  indicates the weights of the  $p$ th node's contribution to the output, and  $z_{kj}^t = \frac{1}{1+e^{-\alpha_{jp}(\mathbf{x}_{kj}^t - \beta_{jp})}}$ ,  $j = 1, \dots, n$ ,  $p = 1, \dots, P$ ,  $\alpha_{jp} > 0$ .

There are three ways to change the TS model: changing the input space, changing the output space, and changing both the input and output spaces. The application of these methods is discussed in our previous paper [32]. But, in summary, we have found that using one specific method does not always produce the best performance. Rather, specific datasets are generated to simulate different cases of fuzzy rule-based domain adaptation, and the corresponding approach is applied to modify the TS model. For example, one may generate two datasets that have the same input distributions but different linear functions that change the output space. Overall, changing the input space is superior in cases where the source and target data have different input distributions and linear functions due to the optimization process for modifying an input or output space. But sometimes optimizing the mapping parameters for the output space can cover the gap between the input data. Therefore, selecting the

best method for modifying the TS model is problem-oriented and depends heavily on the datasets.

Following our previous findings, we suggest the strategy of trying the three methods and choosing the one with the best performance. The small number of parameters in  $\Phi$  and  $\Psi$  make optimizing and constructing the transformation mappings highly efficient, even when the three approaches are implemented simultaneously.

After the transformation mapping, the rules in  $R^t$  will take on a new representation:

$$\text{if } \mathbf{x}_k^t \text{ is } \Phi(A_i(\mathbf{x}_k^t, \mathbf{v}_i^s)), \text{ then } y_k^t \text{ is } \Psi(L_i(\mathbf{x}_k^t, \mathbf{a}_i^s)) \quad i = 1, \dots, ct \quad (13)$$

The overall algorithm for domain adaptation using multiple sources in homogeneous space is provided in Algorithm 1.

---

**Algorithm 1.** Homogenous Domain Adaptation using Multiple Sources

---

**Input:**  $R^{s1}, \dots, R^{sh}$ , and  $T$

**Output:**  $Y_U$  for  $T_U$

---

1. Combine all the rules in  $R^{s1}, \dots, R^{sh}$  to get  $R^s$ .
  2. Determine the number of rules  $ct$  in  $T$  using IGMM.
  3. Select the rules from  $R^s$  to get  $R^t$  for target domain.
    - 3.1 Find the centers of clusters in  $T$ :  $\{\mathbf{v}_1^t, \dots, \mathbf{v}_{ct}^t\}$
    - 3.2 For each  $\mathbf{v}_k^t$ , calculate  $dist(\mathbf{v}_j^s, \mathbf{v}_k^t)$ , find the smallest distance and get the corresponding rule
    - 3.3 Obtain the selected rules  $R^t$
  4. Modify the selected fuzzy rules to fit target data.
    - 4.1 Change the input space
    - 4.2 Change the output space
    - 4.3 Change the input and output spaces
    - 4.4 Compare the above three models and choose one with the best performance
  5. Use the modified and optimized rules to predict labels  $Y_U$  for  $T_U$
- 

#### IV. MULTIPLE-SOURCE DOMAIN ADAPTATION IN HETEROGENEOUS SPACE

This section discusses domain adaptation problems involving multiple sources in heterogeneous space. The symbol representations of the heterogeneous domain adaptation problem are provided first to illustrate the method more clearly in the following discussion. Then, the proposed method of implementing multiple domains knowledge transfer is discussed with detailed procedures.

##### A. Problem Statement

Suppose the datasets in  $h$  source domains and a target domain are  $S_1, \dots, S_h$ , and  $T$ :

$$\begin{aligned} S_1 &= \{(\mathbf{x}_1^{s1}, \mathbf{y}_1^{s1}), \dots, (\mathbf{x}_{N_{s1}}^{s1}, \mathbf{y}_{N_{s1}}^{s1})\} \\ &\dots \\ S_h &= \{(\mathbf{x}_1^{sh}, \mathbf{y}_1^{sh}), \dots, (\mathbf{x}_{N_{sh}}^{sh}, \mathbf{y}_{N_{sh}}^{sh})\} \\ T &= \{T_L, T_U\} = \{(\mathbf{x}_1^t, \mathbf{y}_1^t), \dots, (\mathbf{x}_{N_{t1}}^t, \mathbf{y}_{N_{t1}}^t)\}, \{\mathbf{x}_{N_{t1}+1}^t, \dots, \mathbf{x}_{N_t}^t\} \end{aligned} \quad (14)$$

Unlike homogeneous cases, the number of features in  $\mathcal{S}_1, \dots, \mathcal{S}_h$  are different from those in  $\mathcal{T}$ , i.e., the dimensions of the input data in  $\{\mathbf{x}_1^{s1}, \dots, \mathbf{x}_{N_{s1}}^{s1}\}, \dots, \{\mathbf{x}_1^{sh}, \dots, \mathbf{x}_{N_{sh}}^{sh}\}$  are not identical to  $\{\mathbf{x}_1^t, \dots, \mathbf{x}_{N_t}^t\}$ . In this work, particularly, we have concentrated on cases where the  $h$  source domains share the same feature space, but the feature distributions are different.

Since the dimensions of the input data in the source domains are different from the target domain, it is impossible to apply the models built for the source domains to solve regression tasks in the target domain. Further, merging and transferring knowledge from multiple domains is a more challenging problem. This section presents the multiple-source transfer learning method for heterogeneous space. The specific procedures are described in detail in the next subsection.

### B. Transfer Learning with Multiple Source Domains in Heterogeneous Space

Since the knowledge transfer in heterogeneous space is much more complicated and challenging, knowledge in the source domains needs to be transferred in more than one way to guarantee optimal results. We have incorporated two different forms of knowledge transfer in our method: data transfer and rules transfer.

The processes for using knowledge from the source domains in the form of data and rules are shown in Figs. 5 and 6, respectively. For simplicity,  $h$  is equal to two as an example.

The approach for transferring knowledge in the form of data comprises four steps.

**Step A.1** Combine all the data in  $h$  source domains indistinguishably, denoted as  $\mathcal{S}$ :

$$\mathcal{S} = \{(\mathbf{x}_1^s, \mathbf{y}_1^s), \dots, (\mathbf{x}_{N_{s1}+\dots+N_{sh}}^s, \mathbf{y}_{N_{s1}+\dots+N_{sh}}^s)\} \quad (15)$$

**Step A.2** Extract the latent feature space  $\mathcal{L}_s$  of the combined source and target domains using Canonical Correlation Analysis (CCA). CCA connects two sets of variables by finding linear combinations of variables that maximally correlate. Typically, CCA has two purposes: data reduction by explaining the covariation between two sets of variables using a small number of linear combinations; and data interpretation by finding features (canonical variates) that are important for explaining covariation between sets of variables. Therefore, we have applied CCA here to extract a latent feature space by using the source and target data with unsupervised learning. Two mappings are then learned to map the source and target data to a new latent feature space with data distributions from all domains that are quite similar. The data takes on new representations as follows:

$$\begin{aligned} \bar{\mathcal{S}} &= \{(\bar{\mathbf{x}}_1^s, \mathbf{y}_1^s), \dots, (\bar{\mathbf{x}}_{N_{s1}+\dots+N_{sh}}^s, \mathbf{y}_{N_{s1}+\dots+N_{sh}}^s)\} \\ \bar{\mathcal{T}} &= \{\bar{\mathcal{T}}_L, \bar{\mathcal{T}}_U\} = \{(\bar{\mathbf{x}}_1^t, \mathbf{y}_1^t), \dots, (\bar{\mathbf{x}}_{N_{t1}}^t, \mathbf{y}_{N_{t1}}^t)\}, \{\bar{\mathbf{x}}_{N_{t1}+1}^t, \dots, \bar{\mathbf{x}}_{N_t}^t\} \end{aligned} \quad (16)$$

**Step A.3** Construct the source model using the source data with the new representation.

Based on the combined source dataset  $\bar{\mathcal{S}}$ , a TS fuzzy model

is built and a set of fuzzy rules is obtained:

$$\text{if } \bar{\mathbf{x}}_k^s \text{ is } A_i(\bar{\mathbf{x}}_k^s, \bar{\mathbf{v}}_i^s), \text{ then } \mathbf{y}_k^s \text{ is } L_i(\bar{\mathbf{x}}_k^s, \bar{\mathbf{a}}_i^s) \quad i = 1, \dots, ct \quad (17)$$

where  $ct$  is the number of clusters in the target data (IGMM applied).

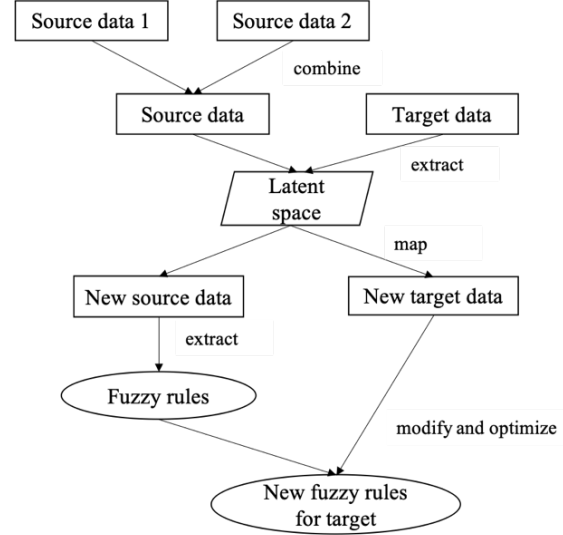


Fig. 5. Transfer learning based on combined data

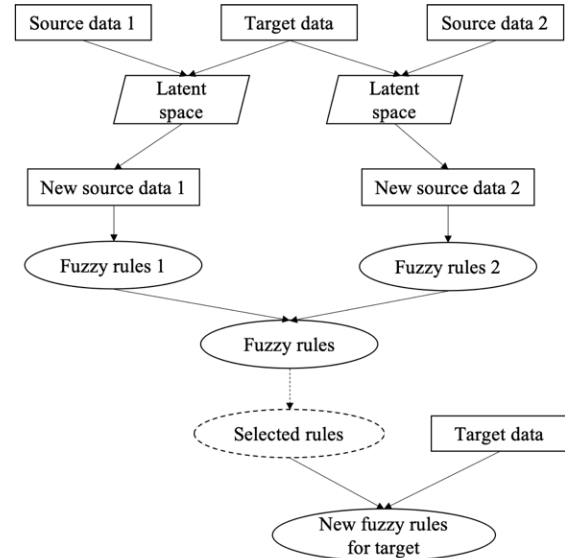


Fig. 6. Transfer learning based on combined rules

**Step A.4** Transfer the fuzzy rules from the combined source domain to the target domain.

The fuzzy rules of the source domain obtained from Step A.3 are modified by changing the input or output space, and the transformation mapping parameters are optimized using  $\bar{\mathcal{T}}_L$ .

The rules are transferred to fit the target data.

$$\text{if } \mathbf{x}_k^t \text{ is } \Phi(A_i(\mathbf{x}_k^t, \bar{\mathbf{v}}_i^s)), \text{ then } \mathbf{y}_k^t \text{ is } \Psi(L_i(\mathbf{x}_k^t, \bar{\mathbf{a}}_i^s)) \quad i = 1, \dots, ct \quad (18)$$

The approach for knowledge transfer in the form of rules contains five steps:

**Step B.1** Extract the latent feature space from the source domains separately.

Based on each source data  $\{\mathbf{x}_1^{sj}, \dots, \mathbf{x}_{N_{sj}}^{sj}\}$  ( $j=1, \dots, h$ ) and target data  $\{\mathbf{x}_1^t, \dots, \mathbf{x}_{N_t}^t\}$ , apply CCA and extract a latent feature space  $\mathbf{L}_{sj}$ , and the data in  $\mathbf{S}_j$  and  $\mathbf{T}$  are converted to

$$\begin{aligned} \tilde{\mathbf{S}}_j &= \{(\tilde{\mathbf{x}}_1^{sj}, y_1^{sj}), \dots, (\tilde{\mathbf{x}}_{N_{sj}}^{sj}, y_{N_{sj}}^{sj})\} \\ \tilde{\mathbf{T}}_j &= \{\tilde{\mathbf{T}}_L^j, \tilde{\mathbf{T}}_U^j\} = \{(\tilde{\mathbf{x}}_1^{tj}, y_1^{tj}), \dots, (\tilde{\mathbf{x}}_{N_{tj}}^{tj}, y_{N_{tj}}^{tj})\}, \{\tilde{\mathbf{x}}_{N_{tj}+1}^{tj}, \dots, \tilde{\mathbf{x}}_{N_j}^{tj}\} \end{aligned} \quad (19)$$

Note that, with this technique, the dimensions of the  $h$  latent feature spaces must be the same.

**Step B.2** The TS fuzzy models for the  $h$  source domains are built in the new latent feature space separately. And,  $h$  sets of fuzzy rules are constructed correspondingly. Suppose the obtained fuzzy rules are:

$$\begin{aligned} \mathbf{R}\tilde{\mathbf{S}}_1 &= \{\tilde{\mathbf{r}}_1^{s1}, \tilde{\mathbf{r}}_2^{s1}, \dots, \tilde{\mathbf{r}}_{c1}^{s1}\} \\ &\dots \\ \mathbf{R}\tilde{\mathbf{S}}_h &= \{\tilde{\mathbf{r}}_1^{sh}, \tilde{\mathbf{r}}_2^{sh}, \dots, \tilde{\mathbf{r}}_{ch}^{sh}\} \end{aligned} \quad (20)$$

The rules in  $\mathbf{R}\tilde{\mathbf{S}}_j$  ( $j=1, \dots, h$ ) are represented as:

$$\text{if } \tilde{\mathbf{x}}_k^{sj} \text{ is } A_i(\tilde{\mathbf{x}}_k^{sj}, \tilde{\mathbf{v}}_i^{sj}), \text{ then } y_k^{sj} \text{ is } L_i(\tilde{\mathbf{x}}_k^{sj}, \tilde{\mathbf{a}}_i^{sj}) \quad i = 1, \dots, cj \quad (21)$$

where  $cj$  is numbers of clusters in the  $j$ th source domains, respectively.

**Step B.3** Combine the fuzzy rules in the source domains.

Since the dimensions of the  $h$  latent feature spaces are identical, the fuzzy rules for the source domains can be easily combined, denoted as  $\mathbf{RS} = \{\mathbf{r}_1^s, \mathbf{r}_2^s, \dots, \mathbf{r}_{c1+\dots+ch}^s\}$ . Each rule in  $\mathbf{RS}$  is subsequently represented as:

$$\text{if } \tilde{\mathbf{x}}_k^s \text{ is } A_i(\tilde{\mathbf{x}}_k^s, \tilde{\mathbf{v}}_i^s), \text{ then } y_k^s \text{ is } L_i(\tilde{\mathbf{x}}_k^s, \tilde{\mathbf{a}}_i^s) \quad i = 1, \dots, c1 + \dots + ch \quad (22)$$

**Step B.4** Modify the fuzzy rules in  $\mathbf{RS}$  to suit the target data using the same process as in Step A.4.

**Step B.5** Select the  $ct$  rules from  $\mathbf{RS}$  using formula (10) and modify the rules.

In homogeneous situations, not all rules in the combined set are used in the knowledge transfer process. Rather only some rules are selected and then modified for the target domain. In heterogeneous situations, transferring features from the original space into the new latent feature space will inevitably result in some information loss, giving rise to uncertainty in the domain adaptation process. Therefore, to guarantee the best quality transfer, a strategy that uses all the combined rules and a strategy that only relies on selected rules should be tested, and the strategy with the best results should be chosen.

In the two approaches, Step A.1 to Step A.4 and Step B.1 to Step B.5, a latent feature space is used to transform all the data into a unified dimensional space so as to convert the heterogeneous transfer learning problem into a homogeneous transfer learning problem. Therefore, applying the rules adaptation method in Algorithm 1 to implement the following

rules transfer means this procedure does not have to be repeated in Steps A.4 and B.5.

The algorithm for domain adaptation using multiple sources in heterogeneous space is provided in Algorithm 2.

---

**Algorithm 2.** Heterogeneous Domain Adaptation using Multiple Sources

---

**Input:**  $\mathbf{S}_1, \dots, \mathbf{S}_h$  and  $\mathbf{T}$

**Output:**  $\mathbf{Y}_U$  for  $\mathbf{T}_U$

---

1. Apply data-based multiple-source transfer
    - 1.1 Combine  $\mathbf{S}_1, \dots, \mathbf{S}_h$  to  $\bar{\mathbf{S}}$
    - 1.2 Extract  $\mathbf{L}_s$  using  $\mathbf{S}$  and  $\mathbf{T}$ , and convert  $\mathbf{S}$  and  $\mathbf{T}$  to  $\bar{\mathbf{S}}$  and  $\bar{\mathbf{T}}$
    - 1.3 Train fuzzy rules using  $\bar{\mathbf{S}}$
    - 1.4 Modify rules using  $\bar{\mathbf{T}}_L$
    - 1.5 Get labels for  $\mathbf{T}_L$
  2. Implement rule-based multiple-source transfer
    - 2.1 Extract  $\mathbf{L}_{sj}$  using  $\mathbf{S}_j$  and  $\mathbf{T}$ ,  $j=1, \dots, h$
    - 2.2 Convert  $\mathbf{S}_j$  to  $\tilde{\mathbf{S}}_j$ , and train rules  $\mathbf{R}\tilde{\mathbf{S}}_j$
    - 2.3 Combine  $\mathbf{R}\tilde{\mathbf{S}}_1, \dots, \mathbf{R}\tilde{\mathbf{S}}_h$  to get  $\mathbf{RS}$
    - 2.4 Modify the rules in  $\mathbf{RS}$  by constructing mappings
    - 2.5 Select rules from  $\mathbf{RS}$  and modify them
    - 2.6 Get labels for  $\mathbf{T}_L$
  3. Compare the results in 1.5 and 2.6, and select the better model
  4. Predict labels for  $\mathbf{T}_U$
- 

## V. EXPERIMENTS IN HOMOGENEOUS SPACE

We executed a set of experiments to validate and analyze the presented method in dealing with domain adaptation problems when multiple source domains are available. Section 5.1 explains how the synthetic datasets were designed and implemented to simulate multiple-source scenarios, along with our experiments for verifying the effectiveness of this new method and a discussion on the application scope for multiple sources. The experiments in Section 5.2 involve real-world datasets and compare the performance of our method with several state-of-the-art methods on multiple-source domain adaptation problems. The sensitivity of the parameters is also analyzed with practical cases.

### A. Experiments on Synthetic Datasets

Several synthetic datasets were generated to simulate different multiple-source transfer learning scenarios. Although the real cases may be quite different from the scenarios we created using the synthetic datasets, the results and patterns obtained do provide some guidance in knowledge transfer for practical cases.

There are two intuitive baselines for transfer learning problems with multiple sources. The first baseline is a model that contains all the fuzzy rules from all source domains. The second is a single source domain model. We evaluated the performance of our method with three models: no-transfer models, single-source transfer models, and multiple-source transfer models. A no-transfer model means the source model is used directly to solve the target task. A single-source transfer



model indicates that only one domain has been used as the source, and a multiple-source transfer model obviously means that knowledge is leveraged from multiple source domains to support regression tasks in the target domain.

In our experiments, all models were tested on unlabeled target datasets  $T_U$  to verify the models' ability to solve regression tasks in the target domain.

In this set of experiments, we generated four datasets with two, three, four, and five clusters. More details about how the datasets were generated can be found in our previous paper [35]. For each experiment, we chose two datasets to serve as the source domains and one to serve as the target domain. Table I lists the datasets used for each experiment with each dataset denoted by the number of clusters it contains. For example, in Experiments 1-3 the dataset with two clusters is selected as the target domain, and two of the remaining three datasets are selected as the source domains, which results in three configurations. Similar operations were conducted for Experiments 4-6 and 7-9. The reason that the dataset with five clusters was not chosen as the target domain is that the aim is to construct an environment where there are less rules in the target domain than in the combined source domains. A sufficient number of rules is beneficial in the selection process and also creates a guarantee of model performance in the target domain. As such, nine different experimental configurations are shown in Table I, and nine groups of results are shown in Table II.

Table I. Datasets used in each of the nine experiments

Domain	Experiment No.								
	1	2	3	4	5	6	7	8	9
Source 1	4	5	5	2	5	5	5	2	5
Source 2	3	3	4	4	2	4	2	3	3
Target	2	2	2	3	3	3	4	4	4

We tested these nine dataset configurations in Table II with three types of models – no transfer, single transfer, and multiple transfer. The “No-transfer model” actually contains two models, one prediction model for each of the two source domains. Similarly, the “Single-source transfer” involves two models. The “Multiple-source transfer” models also contain two models: one is the TS model with all the fuzzy rules from both source domains; the other is the model built using our proposed method. Root mean square error (RMSE) is used to measure the regression performance. All models were constructed using five-cross validation; therefore, the results are shown in the form of “mean±variance”. The results with the best performance are in highlighted in bold.

The results show that the no-transfer method returned high mean values, which represents the gap between the source domains and the target domain. Comparing the two forms of multiple-source transfer, selecting a set of appropriate rules worked better than the brute force method of combining all the rules, which serves as a clear indication that combining every fuzzy rule leads to redundancy and inferior results. For the most

part, the multiple-source method also worked better than single-source transfer. Exp No. 8 and 9 were the exceptions. Upon further analysis, we attribute the success of the single-source method in these two cases to the poor selection of rules, but this does highlight that multiple-source transfer has some limitations.

Hence, in the following part, we explore the scope of applications to provide some guidance for practical uses of multiple-source transfer learning.

Three experiments are implemented in this part. The input datasets in each experiment are shown in Fig. 7-9. Each figure represents a different input data scenario, where the points in blue indicate input data from Source Domain 1, yellow indicates Source Domain 2, and red indicates the target input data. The linear functions used to generate these datasets are the same in the source and target domains.

In Scenario A, the distributions of all three domains are quite dissimilar, and both Source Domain 1 and Source Domain 2 are very different from the target data, as shown in Fig.7. The results in Table III show that our method of combining rules from multiple source domains has the best performance in this case.

Scenarios B and C are two special cases that are more applicable to a single-source transfer method. However, they require a strict condition – the data structures in all the domains must be identified.

In Scenario B, the distributions of all the three domains are quite similar, but the discrepancies between the two source domains and the target domain are different, as shown in Fig. 8. The results show that the single-source transfer methods are superior to the multi-source transfer methods, and that the single-source transfer methods based on Source Domain 2 performed the best. This is because the data in both source domains have similar distributions to the target domain, and Source Domain 2 is more similar to the target domain than Source Domain 1.

In Scenario C, only one source domain, Source Domain 1, has a similar data structure to the target data, while Source Domain 2 has a dissimilar data structure, as shown in Fig. 9. Thus, the single-transfer method with Source Domain 1 would be superior to the other methods.

The results of implementing transfer learning with different models are shown in Table III. The results with the best performance are indicated in bold.

Analyzing the results from the above three scenarios, we can draw two conclusions. First, if all the source domains have a dissimilar structure to the target domain or the relationships of data structures between domains are implicit, then selecting an appropriate subset of rules from multiple source domains would be the optimal choice. Second, if a source domain exists with a data structure that is similar to the target data, the single transfer method will likely provide the best performance. Further, the closer the source domain to the target domain, the better the transfer result. Since, in real-world applications, it is difficult to identify the structure, especially in high-dimensional datasets, our method could play a significant role in practical multiple-source cases.

Table II. RMSE of three types transfer learning (no, single, and multiple)

Exp No.	No-transfer		Single-source transfer		Multiple-source transfer	
	Source 1	Source 2	Source 1	Source 2	All rules	Our method
1	4.16±0.00	2.46±0.00	0.85±0.30	0.74±0.00	1.51±0.00	<b>0.47±0.00</b>
2	6.46±0.00	1.89±0.00	0.86±0.00	0.57±0.00	1.15±0.00	<b>0.45±0.02</b>
3	7.12±0.00	3.45±0.00	0.73±0.00	1.35±0.03	1.43±0.00	<b>0.35±0.01</b>
4	1.51±0.00	1.63±0.00	0.91±0.01	1.38±0.01	1.37±0.01	<b>0.65±0.00</b>
5	5.80±0.00	1.97±0.00	0.78±0.00	0.84±0.01	2.11±1.06	<b>0.62±0.00</b>
6	6.67±0.00	1.64±0.00	1.52±0.00	0.75±0.02	1.60±0.00	<b>0.61±0.00</b>
7	4.29±0.00	3.65±0.00	0.84±0.00	0.93±0.00	1.10±0.03	<b>0.84±0.03</b>
8	3.23±0.00	1.75±0.00	<b>0.98±0.01</b>	1.07±0.00	1.02±0.02	1.08±0.02
9	3.80±0.00	2.56±0.00	1.73±0.45	<b>0.64±0.01</b>	1.49±2.40	0.88±0.00

Table III. Transfer performances of three scenarios in Figures 7 to 9

Exp. No.	No-transfer		Single-source transfer		Multiple-source transfer	
	Source 1	Source 2	Source 1	Source 2	All rules	Our method
A	3.06±0.00	2.81±0.00	1.52±0.33	1.16±0.08	1.50±0.41	<b>0.67±0.03</b>
B	0.75± 0.00	0.48± 0.00	0.39± 0.01	<b>0.33± 0.00</b>	1.53± 0.06	0.60± 0.05
C	1.35± 0.00	3.63± 0.00	<b>0.29±0.00</b>	0.55± 0.00	0.86± 0.18	0.57± 0.00

In all the above experiments, for simplicity, the source domains are designed to have the same number of data. Please note that unbalanced data in multiple source domains will not affect the performance of the method, since the performance of the source model can be guaranteed as long as the source data covers all the clusters.

B. Experiments on Real-world Datasets

In this section, we used real-world datasets to validate the effectiveness of the proposed multiple-source transfer method. Since the studies on domain adaptation with regression problems are scarce, there are no public datasets for these scenarios. We, therefore, turned to five datasets from the UCI Machine Learning Repository and modified them to simulate a range of multiple-source transfer learning scenarios. Since how the datasets were modified is crucial, a detailed description follows using two datasets as examples.

The ‘‘Condition-based maintenance of naval propulsion plants’’ (CBM) dataset contains 14 features, such as ship speed, gas turbine shaft torque, and so on. These features were used to predict gas turbine decay state coefficients. We split the data according to ship speed; speeds greater than 10 knots formed the source domains (7500 instances), and the remaining 3500 instances were used as the target domain. The source instances were further divided into 4000 for Source 1 and 3500 for Source 2. All instances in the source domains were labeled with only 10 labeled instances in the target domain.

The ‘‘Combined cycle power plant’’ (CCPP) dataset contains four attributes: temperature, ambient pressure, relative humidity, and exhaust vacuum, which were used to predict the net hourly electrical energy output. 6800 instances with a temperature of not greater than 25 degrees formed Source Domains 1 and 2, evenly split into groups of 3400. The remaining 2500 instances formed the target domain. Again, all source instances were labeled, and 10 target instances were labeled.

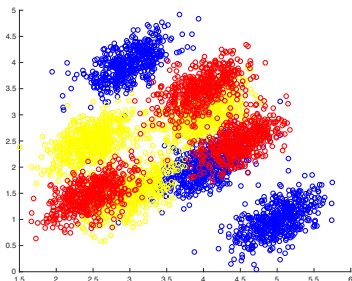


Fig. 7. Scenario A: different distributions in three domains

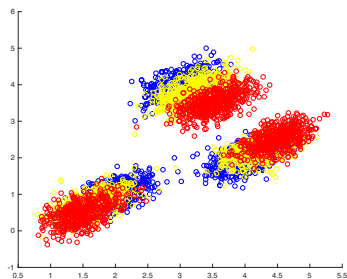


Fig. 8. Scenario B: similar distributions in three domains

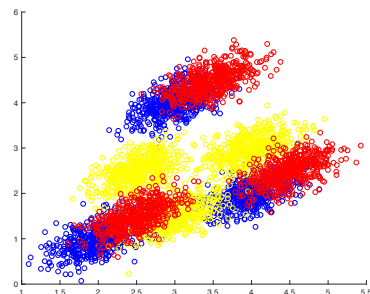


Fig. 9. Scenario C: similar distributions in Source 1 and Target, but different in Source 2

Table IV. Comparison of our method with TSK, TCA, SA, and GFK

Dataset	TSK	TCA	SA	GFK	Our method
Istanbul	2.85±37.72	0.08±0.00	0.08±0.00	0.09±0.00	<b>0.07±0.00</b>
Air quality	2.43±28.47	0.20±0.00	0.19±0.00	0.20±0.00	<b>0.09±0.00</b>
CCPP	0.10±0.00	0.21±0.00	0.21±0.00	0.22±0.00	<b>0.07±0.00</b>
CBM	3.61±0.54	3.69±0.04	12.54±0.07	5.51±1.80	<b>0.31±0.00</b>
Airfoil	0.28±0.00	0.35±0.00	0.34±0.00	0.38±0.00	<b>0.17±0.00</b>

Table V. Performance with varying numbers of clusters (rules) in target domain

clusters	Istanbul	Air quality	CCPP	CBM	Airfoil
2	0.08±0.00	0.09±0.00	0.07±0.00	0.31±0.00	0.18±0.00
3	0.07±0.00	0.10±0.00	0.07±0.00	0.31±0.00	0.18±0.00
4	0.07±0.00	0.09±0.00	0.07±0.00	0.31±0.00	0.27±0.02
5	0.07±0.00	0.10±0.00	0.07±0.00	0.60±0.04	0.23±0.00
6	0.07±0.00	0.12±0.00	0.07±0.00	0.85±0.16	0.17±0.00
7	0.07±0.00	0.11±0.00	0.07±0.00	1.24±0.52	0.47±0.02

The other three datasets are “Istanbul stock exchange dataset”, “Air quality dataset”, and “Airfoil self-noise dataset”. For more details, please refer to the UCI Machine Learning Repository.

We performed two groups of experiments to both validate our method and analyze the impact of the number of clusters. In the first set of experiments, we compared our method with some state-of-the-art methods in transfer learning, i.e., TSK [36], TCA [37], SA [38], and GFK [39]. All these methods are able to solve both classification and regression tasks but have not been presented as solutions for multiple-source situations. To be fair, we used these methods with combined data from all source domains for knowledge transfer.

Although there are some methods and heuristic algorithms for determining the number of clusters, it is often difficult to identify the number of clusters to use when constructing a TS model with real-world data – especially those with high-dimensional data. Hence, in the second set of experiments, we treated the number of clusters as a hyperparameter and discuss its impact on the transfer learning results.

The results of the two groups of experiments are shown in Tables IV and V. The results show superior performance by our proposed method on all five datasets. Table V shows there is no obvious impact on prediction accuracy with a different number of clusters. However, in most cases, the best performance appeared with fewer clusters.

Beside the five datasets, we also applied a new, large and more complex problem of predicting PM 2.5 concentration in different cities to further validate our transfer learning method in multiple-source scenario. The dataset contains PM 2.5 data and related meteorological data in five big cities (Beijing (BJ), Shanghai (SH), Guangzhou (GZ), Chengdu (CD) and Shenyang (SY)) in China from year 2013 to 2015. Beside the values of PM 2.5, there are thirteen main attributes to describe the data: year, month, day, hour, season, dew point, temperature, humidity, pressure, combined wind direction, cumulated wind speed, hourly precipitation, and cumulated precipitation. The thirteen attributes are used as the inputs to predict PM 2.5 concentration.

To simulate the multi-source transfer learning scenario, four groups of experiments have been designed in the new version to implement and validate our method in handling multiple

sources. The transfer performance of these four groups experiments are shown in Tables VI to IX. The first group experiments execute the knowledge transfer among five cities in year 2013, and Table VI displays the transfer performance (RMSE). The third column in Table VI indicates the city that is selected as the target domain, and the second column shows the two cities that are chosen from the remaining cities as the source domains. Two types of transfer learning methods: single-source transfer and multiple-source transfer are implemented. Here, the “Single-source transfer” involves two models: one is transferred with fuzzy rules from Source Domain 1, and the other is transferred with fuzzy rules from Source Domain 2. The “Multiple-source transfer” contains three models: a) combining all the data across the source domains, b) combining the rules from the source domains, and c) selecting rules from the source domains using our proposed method. Similarly, experiments on Tables VII and VIII implement transfer learning in five cities in year 2014 and 2015. The Last group of experiments in Table IX use the data from years 2013 and 2014 to predict the PM 2.5 concentration in year 2015.

The RMSE shown in Tables VI to IX indicates that our proposed method is superior to the single transfer leaning methods and the other two multiple-source transfer learning methods. This further validates the effectiveness of our method.

## VI. EXPERIMENTS IN HETEROGENEOUS SPACE

Our experiments in heterogeneous settings also involved synthetic and real-world datasets.

### A. Synthetic Datasets

We designed two groups of experiments: one with three-dimensional data in the source domains and two-dimensional data in the target domain; the other with four-dimensional data as the source and three-dimensional data as the target. Ten experiments were executed in each group. The settings for one of the ten experiments are provided in Tables X and XI as an example to illustrate the data structures in the three domains.

Table VI. Transfer performance (RMSE) in five cities in year 2013

Exp No.	Datasets setting		Single-source transfer		Multiple transfer		
	Source domains	Target domain	Source 1	Source 2	Combined data	Combined rules	Selected rules (ours)
1	SH, BJ	GZ	14.62±4.06	15.23±0.31	14.13±0.85	16.97±0.32	<b>12.47±1.08</b>
2	SH, GZ	CD	34.11±24.34	30.38±4.99	29.03±6.35	42.85±8.74	<b>24.02±5.58</b>
3	BJ, SY	SH	27.84±11.92	26.67±17.00	28.68±7.27	35.99±6.72	<b>18.17±3.11</b>
4	GZ, CD	BJ	36.68±32.35	33.47±52.27	34.09±8.78	42.62±14.21	<b>29.30±13.38</b>
5	BJ, CD	SY	25.71±3.52	27.81±6.99	27.61±6.35	33.32±12.20	<b>19.80±0.46</b>

Table VII. Transfer performance (RMSE) in five cities in year 2014

Exp No.	Datasets setting		Single transfer		Multiple transfer		
	Source domains	Target domain	Source 1	Source 2	Combined data	Combined rules	Selected rules (ours)
1	CD, GZ	SH	19.08±2.47	17.55±12.39	25.61±27.18	28.13±9.91	<b>14.80±7.06</b>
2	SY, GZ	BJ	23.68±15.04	30.35±16.90	35.53±4.58	41.41±42.58	<b>22.40±0.42</b>
3	BJ, SH	CD	29.47±12.00	31.52±5.36	32.14±13.80	43.32±48.17	<b>28.46±7.59</b>
4	BJ, GZ	SY	45.18±11.56	45.37±5.19	47.52±23.61	48.38±11.86	<b>39.34±8.54</b>
5	BJ, SH	GZ	16.03±3.08	18.89±2.04	17.45±9.63	22.46±15.21	<b>12.65±1.34</b>

Table VIII. Transfer performance (RMSE) in five cities in year 2015

Exp No.	Datasets setting		Single transfer		Multiple transfer		
	Source domains	Target domain	Source 1	Source 2	Combined data	Combined rules	Selected rules (ours)
1	CD, GZ	SH	19.47±5.10	12.91±5.98	21.08±3.43	22.53±5.70	<b>10.57±2.28</b>
2	SH, CD	BJ	42.17±54.56	33.24±7.26	41.05±64.72	77.12±116.80	<b>29.25±4.44</b>
3	BJ, GZ	CD	20.57±2.92	27.46±6.57	22.96±24.84	36.86±110.92	<b>16.17±3.36</b>
4	SY, BJ	GZ	25.30±36.61	16.54±1.10	20.11±10.50	28.47±76.09	<b>12.56±8.69</b>
5	SH, BJ	SY	86.22±532.65	154.0±424.2	523.5±7656	51.32±100.11	<b>27.82±6.76</b>

Table IX. Transfer data from years 2013 and 2014 to 2015

Exp No.	City	Single transfer		Multiple transfer		
		Source 1 (year 2013)	Source 2 (year 2014)	Combined data	Combined rules	Selected rules (ours)
1	SH	11.70±3.19	12.16±4.81	12.75±4.57	15.57±2.34	<b>10.04±1.29</b>
2	SY	34.23±23.81	31.09±35.60	32.34±13.63	44.21±132.23	<b>30.28±14.98</b>
3	CD	17.16±5.66	17.10±8.01	19.84±28.48	24.22±39.53	<b>16.91±16.16</b>
4	GZ	12.16±5.15	10.92±10.35	10.61±7.93	16.11±1.37	<b>6.44±0.09</b>
5	BJ	36.69±24.54	44.69±41.14	47.18±82.94	59.10±60.12	<b>29.37±7.36</b>

Table X. The dataset with 3-D source data and 2-D target data

	Source 1	Source 2	Target
Cluster centers	[1 3 5; 3 2 4; 4 1 3]	[1.5 3 5; 3.5 2 4; 4 1.5 3]	[3 5; 3.5 4; 4 3]
Linear functions	[1 1 1 2; 2 1 3 4; 1 3 1 3];	[1 1 1 2; 2 1 3 4; 1 3 1 3];	[3 3 2; 2 1 5; 4 2 3];

Table XI. The dataset with 4-D source data and 3-D target data

	Source 1	Source 2	Target
Cluster centers	[2 4 1 3; 1 3 5 2; 3 1 4 2];	[2.3 4.1 1.2 3.2; 1.3 3.2 4.8 2.3; 3.1 1.3 3.7 2.4];	[2.5 4.2 1.3; 1.4 3.5 5.2; 3.3 1.1 3.9];
Linear functions	[1 1 1 1 2; 2 1 1 3 4; 1 3 2 1 3];	[2 3 1 4 2; 1 3 4 2 5; 4 1 2 2 3];	[1 1 1 1 1; 2 1 3 4; 1 3 2 3];

The parameters in Tables X and XI show that, although the source and target domains have different dimensions, there are always some shared features with similar values, which represent the relevance between the domains. However, there is always a domain that has quite different linear coefficients than the other two.

Table XII implements the experiments where data in the source domains are three-dimensional and data in the target domain are two-dimensional. And in Table XIII, the source data are four-dimensional and the target data are three-dimensional. Tables XII and XIII show the RMSE of the ten experiments for each group using the single-source transfer method plus three variations of the multiple-source transfer method: combining all the data across the source domains, combining the rules from the source domains, and selecting only some rules from the source domains. The best results are indicated in bold.

Table XII shows the best results in eight of the experiments were obtained by combining all the data from the source domains. Combining all the rules from both source domains worked best in the remaining experiments. However, Table XIII shows that only selecting some rules was advantageous in most of the experiments, while two experiments benefited from combining all the data in the source domains.

The results support our conclusion with the synthetic datasets, i.e., our method, using knowledge from multiple source domains, has good performance in dealing with multiple-source situations. But, the setting of the two groups of experiments cannot cover all cases, and there are some cases where a multiple-source method is not suitable. The following experiments illustrate such inappropriate situations.

Table XII. Heterogeneous transfer with 3-D source data and 2-D target data

Exp No.	Single-source transfer		Multiple-source transfer		
	Source 1	Source 2	Combine data	Combine rules	Selected rules
1	1.66±0.00	1.91±0.01	1.71±0.00	<b>1.62±0.02</b>	2.29±0.02
2	2.61±0.01	2.55±0.01	<b>1.91±0.01</b>	2.15±0.07	2.42±0.01
3	2.51±0.08	2.55±0.01	<b>2.00±0.01</b>	2.21±0.05	2.23±0.02
4	2.61±0.01	2.42±0.02	<b>2.10±0.01</b>	3.24±1.97	2.40±0.15
5	1.89±0.10	1.69±0.01	1.80±0.01	<b>1.57±0.01</b>	2.30±0.01
6	2.51±0.08	2.36±0.01	<b>2.08±0.04</b>	3.30±1.06	2.40±0.14
7	1.75±0.01	1.69±0.01	<b>1.69±0.01</b>	1.81±0.45	2.07±0.02
8	1.75±0.01	1.78±0.02	<b>1.74±0.01</b>	1.90±0.80	2.27±0.00
9	2.51±0.08	2.40±0.01	<b>2.12±0.05</b>	3.58±1.75	2.38±0.09
10	2.84±0.11	2.78±0.05	<b>2.63±0.20</b>	5.35±19.22	5.01±11.58

Table XIII. Heterogeneous transfer with 4-D source data and 3-D target data

Exp No.	Single-domain transfer		Multiple-domain transfer		
	Source 1	Source 2	Combine data	Combine rules	Selected rules
1	3.52±0.00	3.85±0.01	<b>2.37±0.06</b>	4.30±0.34	5.19±0.21
2	2.04±0.09	2.56±0.01	<b>1.88±0.03</b>	4.07±0.19	3.47±0.01
3	3.31±0.00	3.34±0.00	3.52±0.08	3.30±0.00	<b>3.18±0.08</b>
4	4.32±0.10	3.56±0.05	4.20±0.07	4.20±0.67	<b>2.39±0.01</b>
5	4.03±0.07	3.77±0.04	3.98±0.04	3.74±0.12	<b>2.92±0.06</b>
6	4.03±0.07	4.14±0.15	4.22±0.13	5.19±0.39	<b>3.38±1.63</b>
7	5.57±0.12	6.33±0.08	6.11±0.01	7.31±0.16	<b>4.89±0.19</b>
8	4.32±0.10	3.46±0.59	4.08±0.03	4.39±0.30	<b>2.67±0.03</b>
9	6.65±0.17	7.04±0.08	7.23±0.32	7.41±0.29	<b>5.06±0.51</b>
10	4.32±0.10	4.14±0.15	4.32±0.11	4.93±1.21	<b>2.85±0.04</b>

Table XV. Results of experiments for limitation of multiple-source transfer

Exp No.	Single-domain transfer		Multiple-domain transfer		
	Source 1	Source 2	Combine data	Combine rules	Selected rules
1	2.71±0.17	<b>2.64±0.16</b>	2.76±0.21	4.24±1.04	3.37±0.50
2	<b>1.98±0.00</b>	1.98±0.01	2.06±0.01	2.35±0.98	2.02±0.00
3	<b>1.59±0.03</b>	1.67±0.01	1.61±0.00	2.16±0.08	1.87±0.11
4	1.59±0.03	<b>1.53±0.00</b>	1.55±0.00	2.26±0.29	2.08±0.01
5	1.43±0.01	<b>1.35±0.00</b>	1.38±0.01	1.60±0.84	1.67±0.03
6	<b>2.78±0.02</b>	3.17±0.10	2.99±0.67	4.18±1.09	3.80±4.70
7	<b>1.71±0.00</b>	1.82±0.00	1.72±0.00	2.13±1.21	2.90±0.06
8	<b>1.77±0.00</b>	2.01±0.01	1.87±0.00	2.33±0.26	2.39±0.13
9	6.07±0.00	<b>4.71±0.20</b>	5.24±0.41	5.64±0.15	4.85±0.05
10	6.00±0.00	<b>4.40±0.17</b>	5.32±0.00	5.95±0.41	5.41±0.00

Following the same procedures, we designed ten experiments and provide the data structures for one experiment in Table XIV as an example.

The feature values for all three domains are similar, but the most important thing is that the coefficients of linear functions in two source domains are exactly the same, and they are also almost equal to the target domain.

The results of these ten experiments are shown in Table XV.

Table XIV. The dataset for showing limitation of multiple-source transfer

	Source 1	Source 2	Target
Cluster centers	[1 3 5; 3 2 4; 4 1 3];	[1.2 3.2 4.7; 3.2 2.3 3.8; 4.1 1.2 2.8];	[1.2 3; 3.2 2.3; 4 1.2];
Linear functions	[1 1 1 2; 2 1 3 4; 1 3 1 3];	[1 1 1 2; 2 1 3 4; 1 3 1 3];	[1 1 2; 2 1 3; 1 3 2];

As the results show, the multiple-source method was inferior to the single-source method in each of these situations and highlights that proper domain selection is key to producing

good results when drawing on multiple source domains to transfer knowledge. Selecting an appropriate domain is a crucial problem and, therefore, will be studied in future work.

### B. Real-world Datasets

For this set of experiments, we used the “Airfoil self-noise” dataset from UCI, divided according to frequency. Data with a frequency of greater than 800 hertz formed the source domains (900 instances, split equally between Source 1 and Source 2), the remaining 450 instances were used as the target domain. Five attributes – frequency, angle of attack, chord length, free-stream velocity, and suction side displacement thickness, were used to predict scaled sound pressure levels. We removed suction side displacement thickness in the target domain to replicate a heterogeneous setting. All the instances in the source domains were labeled; 10 instances in the target domain were labeled. The results are shown in Table XVI with the best results in bold.

As shown, combining all the data in the source domains gave the best performance in half of the experiments and selecting some of the rules gave the best performance in the remaining

Table XVI. Results for the heterogeneous real-world datasets

Clusters	Single-source transfer		Multiple-source transfer		
	Source 1	Source 2	Combine data	Combine rules	Selected rules
2	0.26±0.00	0.18±0.00	<b>0.167±0.00</b>	0.18±0.00	0.18±0.00
3	0.23±0.00	0.22±0.00	<b>0.22±0.00</b>	0.58±0.55	0.21±0.00
4	0.21±0.00	0.17±0.00	0.16±0.00	0.18±0.00	<b>0.15±0.00</b>
5	0.23±0.00	0.33±0.01	0.18±0.00	0.26±0.00	<b>0.17±0.00</b>
6	0.44±0.00	0.27±0.00	0.36±0.04	1.25±0.81	<b>0.25±0.00</b>
7	0.20±0.00	0.24±0.00	<b>0.18±0.00</b>	0.26±0.00	0.23±0.00

half. Taken overall, we therefore conclude that leveraging multiple domains as sources performs better than using a single source in heterogeneous situations.

### VII. CONCLUSION AND FURTHER STUDY

This work explores transfer learning problems when multiple source domains are available. In homogeneous space, our method is based on combining the fuzzy rules in the source domains, selecting some of those rules, and modifying them to handle regression tasks in a target domain based on labeled target data. We further generalized the idea of using multiple source domains to suit heterogeneous space. Unlike homogeneous cases, where only fuzzy rules are available for transfer, in heterogeneous cases, the source data is also available. This data is used to extract a shared latent feature space for transfer along with the rules. Both methods rely on the same basic procedures. We conducted experiments on synthetic datasets to simulate complex cases of knowledge transfer, and the results validate that our methods have better performance than no-transfer or single-source transfer approaches. Further experiments using real-world datasets support our findings and show that our method of drawing on multiple source domains provides superior performance compared to some state-of-the-art methods in transfer learning.

The methods presented in this paper aim to deal with transfer learning in situations with multiple source domains, especially source domains that share the same feature space. In future studies, we plan to study more complicated cases – for example, where the dimensions of multiple source domains are also different. In addition, we will explore wider applications for these transfer learning techniques, such as activity recognition and robotics. Lastly, these experiments have revealed that source domain selection is a key factor in the methods’ efficacy, which we intend to examine as an effective way to avoid negative transfer in future studies.

### REFERENCES

[1] N. M. Nasrabadi, "Pattern recognition and machine learning," *Journal of Electronic Imaging*, vol. 16, no. 4, p. 049901, 2007.  
 [2] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "Deep learning advances in computer vision with 3d data: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, p. 20, 2017.  
 [3] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle, "Deep learning for computational biology," *Molecular Systems Biology*, vol. 12, no. 7, p. 878, 2016.  
 [4] S. Hoo-Chang *et al.*, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, p. 1285, 2016.

[5] J. Heaton, N. Polson, and J. H. Witte, "Deep learning for finance: deep portfolios," *Applied Stochastic Models in Business and Industry*, vol. 33, no. 1, pp. 3-12, 2017.  
 [6] 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.  
 [7] A. Liu, Y. Su, W. Nie, and M. S. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 102-114, 2017.  
 [8] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Computer Vision and Pattern Recognition (CVPR)*, 2017, vol. 1, no. 2, pp. 7167-7176.  
 [9] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 751-760: ACM.  
 [10] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019-1034, 2015.  
 [11] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1633-1685, 2009.  
 [12] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowledge and Information Systems*, vol. 36, no. 3, pp. 537-556, 2013.  
 [13] 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, 2015.  
 [14] W. Pan, "A survey of transfer learning for collaborative recommendation with auxiliary data," *Neurocomputing*, vol. 177, pp. 447-453, 2016.  
 [15] L. Wen, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on ResNet-50," *Neural Computing and Applications*, pp. 1-14.  
 [16] Z. Lu, Y. Zhu, S. J. Pan, E. W. Xiang, Y. Wang, and Q. Yang, "Source Free Transfer Learning for Text Classification," in *AAAI*, 2014, pp. 122-128.  
 [17] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 160-167: ACM.  
 [18] L. Wen, L. Gao, and X. Li, "A new deep transfer learning based on sparse auto-encoder for fault diagnosis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. 99, pp. 1-9, 2017.  
 [19] R. R. Yager and L. A. Zadeh, *An introduction to fuzzy logic applications in intelligent systems*. Springer Science & Business Media, 2012.  
 [20] J. Shell and S. Coupland, "Fuzzy transfer learning: methodology and application," *Information Sciences*, vol. 293, pp. 59-79, 2015.  
 [21] 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.  
 [22] Z. Deng, Y. Jiang, F.-L. Chung, H. Ishibuchi, and S. Wang, "Knowledge-Leverage-Based Fuzzy System and Its Modeling," *IEEE Trans. Fuzzy Systems*, vol. 21, no. 4, pp. 597-609, 2013.  
 [23] Z. Deng, Y. Jiang, H. Ishibuchi, K.-S. Choi, and S. Wang, "Enhanced knowledge-leverage-based TSK fuzzy system modeling for inductive transfer learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 1, p. 11, 2016.  
 [24] V. Behbood, J. Lu, and G. Zhang, "Fuzzy bridged refinement domain adaptation: Long-term bank failure prediction," *International Journal of Computational Intelligence and Applications*, vol. 12, no. 01, p. 1350003, 2013.  
 [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 Trans. Fuzzy Systems*, vol. 23, no. 6, pp. 1917-1935, 2015.

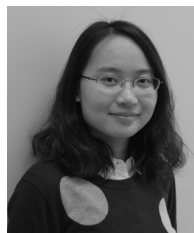
- [26] F. Liu, J. Lu, and G. Zhang, "Unsupervised heterogeneous domain adaptation via shared fuzzy equivalence relations," *IEEE Transactions on Fuzzy Systems*, 2018.
- [27] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, 2010, pp. 1855-1862: IEEE.
- [28] B. Tan, E. Zhong, E. W. Xiang, and Q. Yang, "Multi-transfer: Transfer learning with multiple views and multiple sources," in *Proceedings of the 2013 SIAM International Conference on Data Mining*, 2013, pp. 243-251.
- [29] F. Zhuang, X. Cheng, S. J. Pan, W. Yu, Q. He, and Z. Shi, "Transfer learning with multiple sources via consensus regularized autoencoders," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2014, pp. 417-431: Springer.
- [30] H. Zuo, J. Lu, G. Zhang, and F. Liu, "Fuzzy transfer learning using an infinite Gaussian mixture model and active learning," *IEEE Transactions on Fuzzy Systems*, 2018.
- [31] H. Zuo, G. Zhang, and J. Lu, "Semi-supervised transfer learning in Takagi-Sugeno fuzzy models," in *Data Science And Knowledge Engineering For Sensing Decision Support-Proceedings Of The 13th International Flins Conference*, 2018, vol. 11, p. 316: World Scientific.
- [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*, vol. 26, no. 2, pp. 847-858, 2018.
- [33] H. Zuo, J. Lu, G. Zhang, and W. Pedrycz, "Fuzzy rule-based domain adaptation in homogeneous and heterogeneous spaces," *IEEE Transactions on Fuzzy Systems*, 2018.
- [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] 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, 2017.
- [36] 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.
- [37] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199-210, 2011.
- [38] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2960-2967.
- [39] 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.



**Jie Lu** (F'18) 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 Ph.D. in information systems from the Curtin University of Technology, Australia, in 2000.

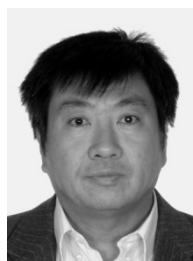
Her main research interests are in the areas of fuzzy transfer learning, concept drift, decision support systems, and recommender systems. She is an IEEE fellow, IFSA fellow and Australian Laureate fellow. She has published six research books and over 450 papers in refereed journals and conference proceedings; has won over 20 ARC Laureate, ARC Discovery Projects, government and industry projects. She serves as Editor-In-Chief for Knowledge-Based Systems (Elsevier) and Editor-In-Chief for International journal of computational intelligence systems. She has delivered over 25 keynote

speeches at international conferences and chaired 15 international conferences. She has received various awards such as the UTS Medal for Research and Teaching Integration (2010), the UTS Medal for Research Excellence (2019), the Computer Journal Wilkes Award (2018), the IEEE Transactions on Fuzzy Systems Outstanding Paper Award (2019), and the Australian Most Innovative Engineer Award (2019).



**Hua Zuo** is a Lecturer 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.



**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 Ph.D. 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 modeling 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 eight Australian Research Council (ARC) Discovery Projects grants and many other research grants. He was awarded an ARC QEII fellowship. He has served 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.