Multiple-source Domain Adaptation in Rule-based Neural Network

1st Hua Zuo
Centre for Artificial Intelligence
University of Technology Sydney
Sydney, Australia
Hua.Zuo@uts.edu.au

2th Jie Lu
Centre for Artificial Intelligence
University of Technology Sydney
Sydney, Australia
Jie.Lu@uts.edu.au

3th Guangquan Zhang
Centre for Artificial Intelligence
University of Technology Sydney
Sydney, Australia
Guangquan.Zhang@uts.edu.au

***Abstract*—Domain adaptation uses the previously acquired knowledge (source domain) to support predicted tasks in the current domain without sufficient labeled data (target domain). Although many methods have been developed in domain adaptation, one issue hasn't been solved: how to implement knowledge transfer when more than one source domain is available. In this paper we present a neural network-based method which extracts domain knowledge in the form of rules to facilitate knowledge transfer, merge rules from all source domains and further select related rules for target domain and clip redundant rules. The method presented is validated on datasets that simulate the multi-source scenario and the experimental results verify the superiority of our method in handling multi-source domain adaptation problems.**

***Index Terms*—domain adaptation, transfer learning, neural network, machine learning, regression**

I. Introduction

The great achievement of machine learning [1] in various fields of data science benefits from the explosion of data, and most famous machine learning methods, such as neural network, SVM and Bayesian network, are supervised learning processes that rely heavily on a large amount of labeled data. This means the accuracy and universality of a model cannot be guaranteed without sufficient labeled data. However, for some cold start problems, only limited data might be available and, especially in the recently emerging areas, accessing massive labeled data for supervised learning is impossible.

Domain adaptation methods [2] are developed due to their ability to transfer knowledge from one domain with enough labeled data (the source domain) to a domain with little or no labeled data (the target domain). The existing domain adaptation methods can be categorized as: domain adaptation [3], cross-domain adaptation [4], multi-task learning [5]. Based on the amount of labeled data in the target domain, the work can be divided into semi-supervised domain adaptation area [6]–[8], where only part of the target data are labeled, and unsupervised domain adaptation [9], [10], a more challenging problem, where no labeled target data is accessible. The increasing demand for solving this data deficiency issue and leveraging related data from auxiliary domains, has prompted an increase in domain adaptation work, some of which begin to focus on precise areas: computational intelligence [11], visualization [12], activity recognition [13], and reinforcement learning [14]. Despite the theoretical work, domain adaptation methods have also been used in many applications, such as medical diagnosis [15], fault diagnosis [16], face recognition [17], sentiment analysis [18], [19] and recommendation systems [20], [21].

Although domain adaptation has seen great advancement and achievement, most work deals with the situation of only one source domain; but in many practical problems, multiple source domains are available for supporting tasks in the current domain. How to merge knowledge from all the source domains and avoid redundant information resulting in negative transfer is a challenging problem, and few works concentrate on the issue of applying multiple source domains. Yao et al. [22] transfer knowledge from multiple sources by extending the boosting framework and develop two algorithms: MultiSource-TrAdaBoost and TaskTraAdaBoost, which increase the number of sources to avoid negative transfer. In order to reduce the distribution discrepancy across different domains, Tan et al. [23] propose a new method that uses knowledge from different sources and views collaboratively, acquiring knowledge of different sources from different views to build a co-training framework. Zhuang et al. [24] transfer source data by learning new feature representation to facilitate knowledge transfer. All source original data are used to train autoencoders to obtain the hidden representation. Simultaneously, multiple classifiers are trained when learning shared feature representation. However, these approaches focus on classification tasks and can't handle the regression tasks in a multi-source scenario. This paper proposes a neural network-based method that is specific for transferring multiple sources in regression tasks.

The main contribution of this work is twofold: abstract and high-level knowledge is extracted using a network work-based model to facilitate knowledge transfer between domains; knowledge from multiple source domains are merged in the form of rules and modified by changing the input space to fit the target domain.

This paper is structured as follows. Section II details the proposed multi-source domain adaptation methods, including extracting abstract knowledge from domains using a neural network-based structure, selecting related knowledge from combined domains and transferring selected rules to the target domain. Section III validates the presented method in real-

world datasets with multi-source setting. The final section concludes the paper and outlines future work.

II. Multi-source Domain Adaptation Method

We propose a new domain adaptation method to deal with multiple sources situations. Prior to presenting the method, some basic concepts about domain adaptation are given. Then the key idea and main steps of our new method are outlined and details of each step formalized and elaborated clearly.

A. Definitions

First, we introduce some basic concepts that are related to domain adaptation to give readers a clear idea of our work, including Domain, Task, Transfer Learning, and Domain Adaptation.

Definition 1 (Domain) [2]: A domain is denoted by $D = \{F, P(X)\}$, where F is a feature space, and $P(X), X = \{x_1, x_2, ..., x_n\}$ are the probability distributions of the instances.

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

Definition 3 (Transfer Learning) [2]: 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 the learning of the target predictive function $f_t(\cdot)$ in D_t using the knowledge in D_s an T_s, where $D_s \neq D_t$, or $T_s \neq T_t$.

In brief, transfer learning aims to use knowledge of a domain (from a source domain) to support the construction of prediction model in a new, but related domain (the target domain). Domain adaptation is one category of transfer learning, where source and target domains share the same feature space, and being distinguished with domain adaptation, cross-domain adaptation is another important category, where the feature spaces in source and target domains are different. In this work, we focus on the domain adaptation problem with a multi-source scenario.

B. Framework of Multi-source Domain Adaptation Method

Our first step is to formalize the problem we aim to solve: considering that we have h source domains, $\boldsymbol{S}_1$, $\boldsymbol{S}_2$,...,$\boldsymbol{S}_h$, where massive labeled data are available in each domain, and a target domain $\boldsymbol{T}$,, with only few labeled data.

The data in each source domain $\boldsymbol{S}_j (j \in \{1, ..., h\})$ are represented as:

$$\boldsymbol{S}_j = \{(\boldsymbol{x}_1^{sj}, y_1^{sj}), (\boldsymbol{x}_2^{sj}, y_2^{sj}), ..., (\boldsymbol{x}_{cj}^{sj}, y_{cj}^{sj})\} \quad j \in \{1, ..., h\} \tag{1}$$

where $(\boldsymbol{x}_k^{sj}, y_k^{sj})$ represents the kth labeled data pair in the jth ($j \in \{1, ..., h\}$) source domain, $\boldsymbol{x}_k^{sj} \in R^n (k = 1, ..., N_{sj})$ is an n-dimensional input variable, the label $y_k^{sj} \in R$ is a continuous variable and N_{sj} is the number of labeled data pairs.

In contrast to source domains, not all data in target domain $\boldsymbol{T}$ are labeled:

$$\boldsymbol{T} = \{\boldsymbol{T}_L, \boldsymbol{T}_U\} = \{\{(\boldsymbol{x}_1^t, y_1^t), ..., (\boldsymbol{x}_{N_{t1}}^t, y_{N_{t1}}^t)\}, \{\boldsymbol{x}_{N_{t1}+1}^t, ..., \boldsymbol{x}_{N_t}^t\}\} \tag{2}$$

Target domain $\boldsymbol{T}$ is consist of one subset $\boldsymbol{T}_L$ with labels and one subsets $\boldsymbol{T}_U$ without labels, where $\boldsymbol{x}_k^t \in R^n (i = 1, ..., N_t)$ is an n-dimensional input variable, $y_k^t \in R$ is the label only accessible for the first N_{t1} data. The numbers of data in $\boldsymbol{T}_L$ and $\boldsymbol{T}_U$ are N_{t1} and $N_t - N_{t1}$, respectively, and satisfy $N_{t1} << N_t$, $N_{t1} << N_{s1}$,..., $N_{t1} << N_{sh}$.

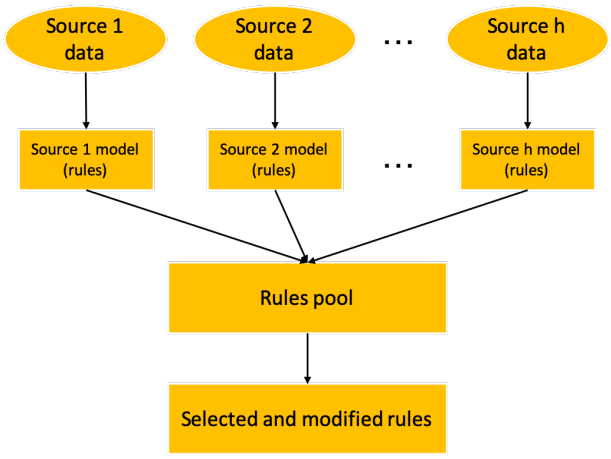

Fig. 1. Framework of multi-source domain adaptation method

For each source domain, a well-performed prediction model could be built because of sufficient labeled data but these models show poor accuracy on target tasks due to the distribution discrepancy between domains. In order to fill this gap and effectively merge all source domains, we propose a new multi-source domain adaptation method, the framework of which is illustrated in Fig 1 , which contains three key elements:

a) A prediction model with neural network structure is used to extract abstract knowledge from one domain and the high-level knowledge is expressed as the form of rules. The abstract rules could facilitate exploring knowledge shared between domains and the usage of rules could avoid using original data, thereby protecting privacy, such as medical data.

b) Combining rules from all source domains is a natural and simple way to deal with multiple source domains but the redundancy of the combined rules will harm the knowledge transfer and may lead to negative transfer. So, we select some appropriate rules based on the defined principle and adapt them to target domain.

c) The selected rules of source domains cannot be used directly to target domain due to the domain distributions gap. So, they need to be modified to fit the target domain. We develop a method of changing the input space using mappings with neural network, with the mappings optimized to make the rules more compatible with target tasks.

C. Four Steps of Implementing Multi-source Domain Adaptation Method

In this section, more details and formulae will be provided to indicate how to implement the four steps of multi-source domain adaptation method.

Step 1: Build a prediction model for each source domain and combine all the learned rules.

A prediction model is built for each source domain using the labeled data $\boldsymbol{S}_j$, and the structure of the model is based on a neural network with four layers, as shown in Fig 2. The first layer is for the input variables and the number of neurons is the dimension of the feature space. The second layer is the clustering results of the former layer which grasps the geometrical structure of input data and the third layer gives the actions that define each cluster formed from the second layer. The final layer is the output that summarises the former three.

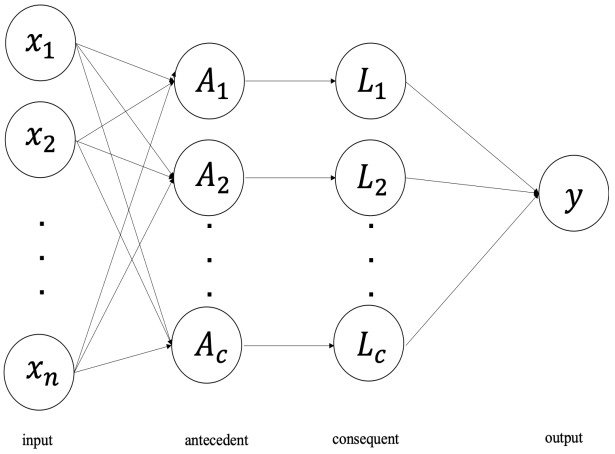

Fig. 2. Structure of prediction model

The construction of the prediction model in Fig 2 shows how it could learn abstract knowledge in the form of rules which are represented as:

$$r(\boldsymbol{v}_i, \boldsymbol{a}_i):$$
$$\text{if } \boldsymbol{x} \text{ is } A_i(\boldsymbol{x},\ \boldsymbol{v}_i),\ \text{then } y \text{ is } L_i(\boldsymbol{x}, \boldsymbol{a}_i) \qquad i = 1, 2, \cdots, c \tag{3}$$

Each rule $r(\boldsymbol{v}_i, \boldsymbol{a}_i)$ contains two parts: $A_i(\boldsymbol{x},\ \boldsymbol{v}_i)$ defines the antecedent of a rule, where $\boldsymbol{v}_i$ is the cluster centers learned in the second layer and $L_i(\boldsymbol{x}, \boldsymbol{a}_i)$ is the consequent of a rule, where $\boldsymbol{a}_i$ is the coefficient of the linear function. From this it follows that the learning process of this model consists mainly of two steps: 1) use a clustering algorithm to cluster the input data and get the cluster centers, which is an unsupervised learning process using $\boldsymbol{x}_k^{sj}$; 2) calculate the coefficients of linear functions, which could be obtained by an optimization applying labeled data $(\boldsymbol{x}_k^{sj}, y_k^{sj})$.

Summarizing all the rules, the output of the prediction is $y = \sum_{i=1}^{c} A_i(\boldsymbol{x}, \boldsymbol{v}_i) L_i(\boldsymbol{x}, \boldsymbol{a}_i)$, where $A_i(\boldsymbol{x},\ \boldsymbol{v}_i) = 1/\sum_{j=1}^{c} \frac{\|\boldsymbol{x}-\boldsymbol{v}_i\|}{\|\boldsymbol{x}-\boldsymbol{v}_j\|}^{\frac{2}{m-1}}$, and $L_i(\boldsymbol{x}, \boldsymbol{a}_i) = a_{i0} + a_{i1}x_1 + \cdots + a_{in}x_n$.

The result is that, for each source domain, a prediction model is built and a set of rules are obtained, denoted as $\boldsymbol{R}^{s1}, \boldsymbol{R}^{s2}, ...,$ and $\boldsymbol{R}^{sh}$:

$$\boldsymbol{R}^{sj} = \{r(\boldsymbol{v}_1^{sj}, \boldsymbol{a}_1^{sj}), r(\boldsymbol{v}_2^{sj}, \boldsymbol{a}_2^{sj}), ...r(\boldsymbol{v}_{cj}^{sj}, \boldsymbol{a}_{cj}^{sj})\} \tag{4}$$

where $r(\boldsymbol{v}_i^{sj}, \boldsymbol{a}_i^{sj})$ represents a rule in the jth source domain with the center $\boldsymbol{v}_i^{sj}$ and coefficients of linear functions $\boldsymbol{a}_i^{sj}$.

Step 2: Determine the target data structure to assist the ensuing rules selection process.

The structure of target data needs to be explored before selecting rules for the target domain. Here we use the Infinite Gaussian Mixture model (IGMM) [25] which explores the structure of data based on distributions, to determine the number of clusters, i.e. the number of rules in the target domain.

Step 3: Select appropriate rules for the target domain.

A pool of rules is obtained after Step 1, denoted as $\boldsymbol{R}^s$:

$$\boldsymbol{R}^s = \boldsymbol{R}^{s1} \cup \boldsymbol{R}^{s2} \cup \cdots \cup \boldsymbol{R}^{sh} \tag{5}$$

Determining how to choose the appropriate rules for the current target domain is crucial. Since the discrepancy of source and target domains is reflected on the distribution, we use the input data structure to select rules. The rules are recognized and separated using a clustering algorithm. So, we apply the cluster centers to seek appropriate rules for target domain.

The cluster centers of all rules in $\boldsymbol{R}^s$ are denoted as $\boldsymbol{V}^s$:

$$\boldsymbol{V}^s = \{\boldsymbol{v}_1^s, \boldsymbol{v}_2^s, ..., \boldsymbol{v}_{cs}^s\} \tag{6}$$

where $cs = c1 + \cdots + ch$ represents the total number of rules in all source domains.

Although rules cannot be obtained for the target domain, we could apply the clustering algorithm to the target input data to get the cluster centers, denoted as $\boldsymbol{V}^t$:

$$\boldsymbol{V}^t = \{\boldsymbol{v}_1^t, \boldsymbol{v}_2^t, ..., \boldsymbol{v}_{ct}^t\} \tag{7}$$

In order to select rules from $\boldsymbol{R}^s$ for target domain, we define the close degree of the source rules with target domain:

$$CD(\boldsymbol{v}_i^{sj}, \boldsymbol{T}) = -\sum_{k=1}^{ct} dis(\boldsymbol{v}_i^{sj} - \boldsymbol{v}_k^t) \tag{8}$$

where $\boldsymbol{v}_i^{sj}$ and $\boldsymbol{v}_k^t$ are the centers of clusters in the source and target domains, separately. $CD(\boldsymbol{v}_i^{sj}, \boldsymbol{T})$ defines the close degree of one rule $r(\boldsymbol{v}_i^{sj}, \boldsymbol{a}_i^{sj})$, the ith rule from jth source domain, with the target domain by calculating the distance of $\boldsymbol{v}_i^{sj}$ with all cluster centers in the target domain. Then we sort all the calculation results, and select the first ct source rules with the greatest close degree with the target domain.

Step 4: Modify the selected rules and make them adaptable for the target domain.

The selected rules from source domains are not adaptive for solving regression tasks in the target domain due to the distribution gap and different functions in subspace. As an alternative, we modify the existing rules by changing the input space using mappings. The mappings are constructed using shallow networks in conjunction with the prediction model. The resulting model for the target domain is shown in Fig. 3.

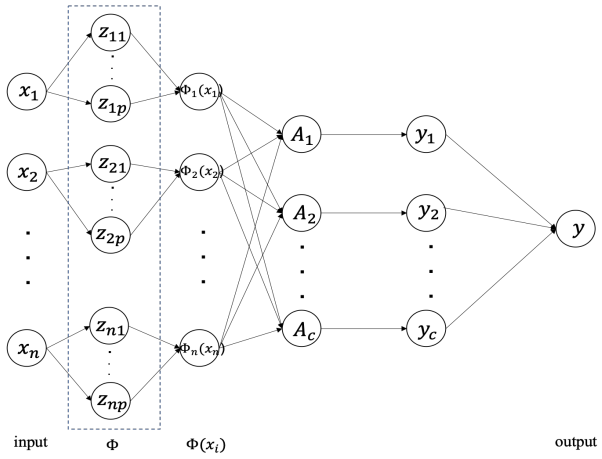

Fig. 3. Structure of prediction model

From Fig. 3, we can see that each input variable is modified by using a small network which forms a nonlinear transformation Φ to the input space. The active functions in the hidden layer of mappings are the modified sigmoid function: $\Phi(x) = 1/(1 + exp(-\alpha(x - \beta)))$. Then the output of this network becomes:

$$y^t = \sum_{i=1}^{ct} A_i(\Phi(\boldsymbol{x}^t), \Phi(\boldsymbol{v}_i^s))L_i(\Phi(\boldsymbol{x}^t), \boldsymbol{a}_i^s) \tag{9}$$

Since the mapping Φ is applying to the input space, it acts on not only the instance $\Phi(\boldsymbol{x}^t)$, but also the cluster centers $\Phi(\boldsymbol{v}_i^s)$. To obtain the best performance, the parameter of the mappings are obtained by optimizing 10 using the labeled target data. Although the amount of labeled target data is low, it still could guide the optimized parameters to fit the target data.

$$Cost = \sqrt{\frac{1}{N_{t1}} \sum_{k=1}^{N_{t1}} \sum_{i=1}^{ct} (A_i(\Phi(\boldsymbol{x}_k^t), \Phi(\boldsymbol{v}_i^s))(L_i(\Phi(\boldsymbol{x}_k^t), \boldsymbol{a}_i^s) - y_k^t)^2} \tag{10}$$

The quality of the labeled target data greatly impacts the performance of the model and the labeled data that covers all clusters will guarantee the transfer efficiency. To make this achievable, active learning is applied to select the labeled target data or label them where no labels are available.

III. Experiments

A set of experiments is set to evaluate and analyze the proposed method in handling domain adaptation problems when multiple source domains are available. We begin by explaining the datasets and the experimental setting that simulates the multi-source environment in domain adaptation; then the experimental results compared with some baselines are shown and analyzed.

We use the dataset "PM2.5 Data of Five Chinese Cities Data Set" from "UCI Machine Learning Repository". This dataset includes PM2.5 data and related meteorological data in five big cities Gangzhou (GZ), Shanghai (SH), Chengdu (CD), Beijing (BJ) and Shenyang (SY) in China across years 2013 to 2015. Thirteen related meteorological attributes are used to predict the PM2.5 values: year, month, day, hour, season, dew point, temperature, humidity, pressure, combined wind direction, cumulated wind speed, hourly precipitation and cumulated precipitation.

In order to simulate the multi-source domain adaptation environment, we design four groups of experiments to implement knowledge transfer from different cities and different years. The performance of these four groups of experiments are shown in Tables I, II, III, and IV. The first three groups of experiments use the data in years 2013, 2014, and 2015 separately, with data of two cities selected as source domains and one remaining city is as target domain. The fourth group of experiments transfers the data from years 2013 and 2014 in one city to support the PM2.5 value prediction in year 2015.

The experimental comparison setting in four groups experiments is the same, consequently we only give Table I as an illustration. The second and third columns list the domains setting in each experiment, where column two gives the two selected source domains and column three shows the target domain. For example, for experiment 1, data from cities Shanghai and Beijing are source domains, denoted as Source 1 and Source 2, to predict PM2.5 value in Guangzhou (target domain). Two types of domain adaptation are compared with our method to validate its transferring ability in handling multiple source domains: single transfer and multiple transfer. Two models are contained in "Single transfer": one is using data only from Source 1 to target domain, and the other is transferring data from Source 2 to support target domain. The "Multiple transfer" has three ways of implementing knowledge transfer from multiple source domains: a) by combining all the data across the source domains; b) by combining the rules from the source domains and c) selecting rules from the source domains using our proposed method. Similarly, experiments on Tables III, II and IV have the same comparison factors.

RMSE is used to measure the performance of each method, and five-fold cross validation is applied to all the experiments, so the results are shown in the form of "mean $\pm$ variance".

Analyzing the results in four tables, we acquire the following conclusions:

a) The performance of our proposed method is superior to both the "Single transfer" and "Multiple transfer" methods.

b) The method of "Combined rules" has a poor accuracy in all experiments, implying that simply combining all the rules would contain redundant information that will lead to negative transfer.

TABLE I
TRANSFER BETWEEN DIFFERENT CITIES IN YEAR 2013

Exps	Datasets setting		Single transfer		Multiple transfer		
	Source domains	Target	Source 1	Source 2	Combined data	Combined rules	Selected rules
1	SH, BJ	GZ	14.23±0.09	15.40±6.94	13.05±1.54	18.11±0.24	12.56±0.00
2	SH, SY	GZ	17.50±15.40	17.25±3.44	15.21±5.75	18.18±13.64	14.12±0.62
3	SH, GZ	CD	31.43±1.91	30.02±0.57	27.80±24.12	34.42±3.58	25.09± 1.22
4	SH, BJ	CD	25.40±34.57	34.32±4.55	39.38±0.82	35.24±1.04	22.75±15.77
5	SY, BJ	SH	31.79±0.08	30.55±30.86	32.15±0.97	34.90±2.56	22.35±5.44
6	SY, CD	SH	19.01±2.89	23.24±28.39	28.38±0.03	80.10±3389	16.51±13.4
7	GZ, SH	BJ	33.09±55.44	32.41±122.05	27.49±2.16	44.40±0.04	27.27±1.80
8	SH, SY	BJ	35.61±12.7	31.85±45.58	30.07±9.61	51.65±90.62	29.41±1.43
9	SH, CD	SY	25.92±5.70	28.24±4.33	33.66±1.51	30.92±45.08	24.12±0.62
10	BJ, GZ	SY	26.94±27.3	25.59±2.71	31.37±19.32	35.26±2.83	24.72±4.24

IV. CONCLUSION AND FUTURE STUDY

This work explores the domain adaptation problem when multiple source domains are available and proposes a method that could combine and select related knowledge for a target domain to ensure effective transfer and avoid corruption from redundant information. The method presented is validated on a dataset for predicting PM2.5 values between different cities in different years and our method performs better than the other two baselines.

The method presented in this paper aims to solve the multiple source domains transfer problem, especially where the source domains and target domain have the same feature space, i.e. same feature dimension and attributes significance. In future, we will consider more complex problems – for example, where the source and target domains have different feature spaces and even the multiple source domains don't share the same features.

ACKNOWLEDGMENT

This work is supported by the Australia Research Council under DP170101632 and FL190100149.

REFERENCES

[1] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

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

[3] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[4] 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*. ACM, 2010, pp. 751–760.

[5] A.-A. Liu, Y.-T. Su, W.-Z. Nie, and M. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 102–114, 2016.

[6] 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.

[7] 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.

[8] 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.

[9] F. Liu, J. Lu, and G. Zhang, "Unsupervised heterogeneous domain adaptation via shared fuzzy equivalence relations," *IEEE Transactions on Fuzzy Systems*, 2018.

[10] F. Liu, G. Zhang, and J. Lu, "Unconstrained fuzzy feature fusion for heterogeneous unsupervised domain adaptation," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2018, pp. 1–8.

[11] 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.

[12] 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, 2014.

[13] 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.

[14] 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.

[15] R. K. Samala, H.-P. Chan, L. M. Hadjiiski, M. A. Helvie, K. H. Cha, and C. D. Richter, "Multi-task transfer learning deep convolutional neural network: application to computer-aided diagnosis of breast cancer on mammograms," *Physics in Medicine & Biology*, vol. 62, no. 23, p. 8894, 2017.

[16] 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*, vol. 49, no. 1, pp. 136–144, 2017.

[17] C.-X. Ren, D.-Q. Dai, K.-K. Huang, and Z.-R. Lai, "Transfer learning of structured representation for face recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5440–5454, 2014.

[18] Y. Yoshida, T. Hirao, T. Iwata, M. Nagata, and Y. Matsumoto, "Transfer learning for multiple-domain sentiment analysis—identifying domain dependent/independent word polarity," in *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[19] Q. You, J. Luo, H. Jin, and J. Yang, "Robust image sentiment analysis using progressively trained and domain transferred deep networks," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[20] W. Pan, "A survey of transfer learning for collaborative recommendation with auxiliary data," *Neurocomputing*, vol. 177, pp. 447–453, 2016.

[21] H. Kanagawa, H. Kobayashi, N. Shimizu, Y. Tagami, and T. Suzuki, "Cross-domain recommendation via deep domain adaptation," in *European Conference on Information Retrieval*. Springer, 2019, pp. 20–29.

[22] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1855–1862.

[23] 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*. SIAM, 2013, pp. 243–251.

[24] 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*. Springer, 2014, pp. 417–431.

[25] C. E. Rasmussen, "The infinite gaussian mixture model," in *Advances in Neural Information Processing Systems*, 2000, pp. 554–560.

TABLE II
TRANSFER BETWEEN DIFFERENT CITIES IN YEAR 2014

Exps	Datasets setting		Single transfer		Multiple transfer		
	Source domains	Target	Source 1	Source 2	Combined data	Combined rules	Selected rules
1	CD, GZ	SH	13.31±5.21	15.61±0.98	23.12±56.98	33.51±18.39	12.68±16.49
2	BJ, CD	SH	15.69±0.15	13.77±9.89	12.76±2.66	28.87±3.13	9.13±0.55
3	BJ, SH	CD	32.98±0.00	31.02±6.34	35.83±85.74	63.08±169.20	28.15±37.27
4	BJ, SY	CD	30.31±2.61	31.7±2.58	46.79±2.71	39.15±0.35	26.47±123.24
5	BJ, CD	SY	27.81±0.03	31.61±49.83	38.63±91.15	48.97±73.04	27.67±1.48
6	SH,CD	SY	51.83±8.47	43.32±1.63	51.44±5.85	43.53±0.16	40.83±23.53
7	BJ, CD	GZ	16.98±0.00	16.88±0.16	12.74±2.83	25.76±4.86	13.82±4.85
8	SY, BJ	GZ	16.75±29.82	16.29±50.92	16.52±2.85	23.94±7.25	16.14±0.00
9	SH, CD	BJ	34.20±3.92	20.31±2.21	31.90±7.63	28.27±55.71	18.31±0.09
10	SH, GZ	BJ	35.51±0.71	29.44±50.09	27.43±9.15	36.04±22.16	21.75±2.13

TABLE III
TRANSFER BETWEEN DIFFERENT CITIES IN YEAR 2015

Exps	Datasets setting		Single transfer		Multiple transfer		
	Source domains	Target	Source 1	Source 2	Combined data	Combined rules	Selected rules
1	CD, GZ	SH	17.57±4.56	14.70±0.89	18.18±1.34	25.52±13.33	10.83±0.49
2	CD, SY	SH	11.37±0.10	20.46±1.18	19.11±24.78	18.32±1.45	7.82±0.60
3	SH, CD	BJ	36.36±3.33	33.69±97.52	31.91±26.98	42.08±33.35	29.06±3.29
4	SH, SY	BJ	42.40±23.73	30.11±59.75	40.00±98.00	50.71±7.34	33.78±97.85
5	BJ, GZ	CD	18.66±21.59	15.70±0.23	23.08±32.17	17.92±3.90	15.00±4.70
6	BJ,SH	CD	13.73±2.71	15.62±2.06	16.43±0.72	17.47±9.04	11.14±0.18
7	SY,BJ	GZ	15.70±0.15	15.36±44.98	19.52±21.23	22.56±27.56	9.63±1.01
8	SY,SH	GZ	14.94±27.52	12.99±7.56	20.53±123	21.48±0.00	7.07±0.01
9	SH, BJ	SY	63.31±789	229±1570	511±3900	55.06±33.14	43.42±411
10	SH,GZ	SY	68.49±195	88.64±78.65	118±504	87.67±893	32.03±30.67

TABLE IV
TRANSFER OF CITIES FROM DIFFERENT YEARS

Exps	Datasets setting		Single transfer		Multiple transfer		
	Source domains	Target	Source 1	Source 2	Combined data	Combined rules	Selected rules
SH	2013, 2014	2015	14.36±9.89	13.89±3.85	15.58±28.09	23.27±30.84	13.91±7.65
SY	2013, 2014	2015	46.64±10.51	44.97±5.45	42.86±182	56.47±193	35.55±1.17
CD	2013, 2014	2015	13.56±3.29	12.26±0.48	16.50±0.24	22.39±18.05	11.25±0.81
GZ	2013, 2014	2015	19.13±7.69	13.27±21.40	15.54±0.07	16.28±2.74	11.78±4.33
BJ	2013, 2014	2015	39.85±14.69	51.54±9.82	45.25±30.79	76.15±379	32.59±2.86