

A FUZZY DOMAIN ADAPTATION METHOD BASED ON SELF-CONSTRUCTING FUZZY NEURAL NETWORK

PENG HAO* and GUANGQUAN ZHANG and VAHID BEHBOOD

*Centre for Quantum Computation and Intelligent Systems
Faculty of Engineering and Information Technology, University of Technology Sydney,
Sydney, NSW 2007, Australia*

**E-mail: Peng.Hao@student.uts.edu.au, Guangquan.Zhang@uts.edu.au,
Vahid.Behbood@uts.edu.au*

ZHENG ZHENG

*School of Automation Science and Electrical Engineering, Bei Hang University,
Beijing, 100191, China*

E-mail: zhengz@buaa.edu.cn

Domain adaptation addresses the problem of how to utilize a model trained in the source domain to make predictions for target domain when the distribution between two domains differs substantially and labeled data in target domain is costly to collect for retraining. Existed studies are incapable to handle the issue of information granularity, in this paper, we propose a new fuzzy domain adaptation method based on self-constructing fuzzy neural network. This approach models the transferred knowledge supporting the development of the current models granularly in the form of fuzzy sets and adapts the knowledge using fuzzy similarity measure to reduce prediction error in the target domain.

Keywords: Domain adaptation; Fuzzy Neural Network; Fuzzy Similarity.

1. Introduction

Traditional machine learning techniques have been proved to be effective when assuming the training and test data are sampled from the same distribution. However, this assumption does not hold in reality due to the quick change of external environment, which means that we can not make a direct use of out-dated model for a new task. Moreover, to gather enough manually labeled data for retraining a classifier is also a tough task in many practical applications, like computer vision,¹ object recognition,² natural language processing.³ Domain adaptation seeks to solve the problem of generalizing a model trained in a previous domain (source domain) to perform predictions in a different but related domain (target domain) where we have little or no labeled data. Existed domain adaptation methods seek to solve this

problem primarily in three ways: re-weighting source instances iteratively,³ finding the common features shared between source and target domain,⁴ or learning a new representation which is intrinsic in both domains.^{1,5,6}

Although many domain adaptation methods have been proposed, the performance is not yet acceptable. One key problem is lack of considering information granularity.^{7,8} It is intuitively legitimate to anticipate that, while the models developed on the basis of the previously available experimental data produce numeric outcomes, the transferred knowledge supporting the development of the current models becomes granular (e.g., in the form of fuzzy sets) to capture the resemblance of the previously encountered environment with the current situations and account for differences. In this paper, in order to handle this key problem and take the information granularity into domain adaptation, we model the transferred knowledge granularly in the form of fuzzy sets and propose a new fuzzy domain adaptation method based on self-constructing fuzzy neural network.

The rest of this paper is organized as follows. Section 2 gives out necessary notations about domain adaptation and a brief introduction of self-constructing fuzzy neural network. In Section 3, we describe our proposed method. We summarize the paper and point out future study in Section 4.

2. Problem settings and Notations

In this section, we introduce some notations for domain adaptation problem, then we briefly describe the components and functions of self-constructing fuzzy neural network.

2.1. Notations

Formally, let $D_s^l = \{(x_1^s, y_1^s), (x_2^s, y_2^s), \dots, (x_{n_s}^s, y_{n_s}^s)\}$ denotes labeled source domain instances which are drawn from some distribution $P_s(x, y)$, where $x_i^s \in \chi^s \subset R^n$ represents a n dimensional feature vector developed from feature space χ^s and y_i^s is the corresponding label value from label space $Y = \{1, 2, \dots, m\}$. Similarly, let $D_t^u = \{x_1^t, x_2^t, \dots, x_{n_t}^t\}$ denotes unlabeled target domain dataset which is sampled from distributin $P_t(x, y)$, where $x_i^t \in \chi^t \subset R^n$. In the setting of domain adaptation, as the marginal distribution $P_s(\chi^s) \neq P_t(\chi^t)$, so that $P_s(x, y) \neq P_t(x, y)$. Regarding fuzzy domain adaptation, let $x_i^s \in \Lambda^s$ and $x_i^t \in \Lambda^t$, where $\Lambda^s = \{A_{ij}^s, i = 1, 2, \dots, n, j = 1, 2, \dots, r\}$ and $\Lambda^t = \{A_{ij}^t, i = 1, 2, \dots, n, j = 1, 2, \dots, r\}$ are the set of membership functions of fuzzy sets (clusters). n denotes the dimension of the data and r is the number of fuzzy sets generated in each dimension. So that fuzzy domain adaptation problem can be described as

$\exists A_{ij}^s \in \Lambda^s$ and $A_{ij}^t \in \Lambda^t$, $A_{ij}^s \neq A_{ij}^t$. Our objective is to learn a classifier $f(x): \chi^t \rightarrow Y$ for unlabeled target domain data D_t^u by exploiting the knowledge of labeled source domain data D_s^l .

2.2. Self-constructing fuzzy neural network

Our method is based on self-constructing fuzzy neural network,⁹ which contains five layers with n input nodes and one output node. The learned rules have the form

$$\begin{aligned} & \text{If } x_1 \text{ is } A_{1k} \text{ and } x_2 \text{ is } A_{2k} \text{ and } \dots \text{ and } x_n \text{ is } A_{nk} \\ & \text{then } y \text{ is } y_m \end{aligned} \quad (1)$$

where A_{ik} is the k th fuzzy set generated by the i th dimensional variable of an input instance \vec{x} .

Layer1: Input layer. Each node represents one dimension of an input data.

Layer2: Fuzzification layer. In this layer, each node acts as a fuzzy set for one of the input variables in Layer1 based on the idea of clustering. Different fuzzy sets depict different rules, so the number of fuzzy sets formed by an input variable in Layer1 is equal to the number of fuzzy rules. Gaussian function is adopted to calculate the membership of an input variable belonging to a fuzzy set as follows:

$$A_{ik}(x_i) = \exp \left\{ - \left(\frac{(x_i - \mu_{ik})^2}{\sigma_{ik}^2} \right) \right\} \quad (2)$$

where μ_{ik} and σ_{ik}^2 denotes the k th center and width of the fuzzy set respectively corresponding to the i th input variable x_i .

Layer3: Rule layer. Each node represents a fuzzy rule. The antecedent matching of a rule is expressed as:

$$E_k(\vec{x}) = \exp \left\{ \prod_{i=1}^n A_{ik}(x_i) \right\}, k = 1, 2, \dots, r \quad (3)$$

where $E_k(\vec{x})$ denotes the firing strength of rule R_k when instance \vec{x} is entered, and r is the total number of rule nodes.

Layer4: Defuzzification layer. There is only one node which sums up outputs from all the rule nodes and performs defuzzification as follows:

$$y_m(\vec{x}) = \frac{\sum_{k=1}^r \omega_k E_k(\vec{x})}{\sum_{k=1}^r E_k(\vec{x})} \quad (4)$$

where ω_k denotes the k th connecting parameter and $y_m(\vec{x})$ is the output value based on the input instance \vec{x} . Generally, all the connecting parameters are selected randomly in the beginning.

Layer5: Output layer. There is only one label node in this layer which converts the output value in Layer4 to the label information. In binary classification problem, if $y_m(\vec{x}) \geq T_s$, where T_s controls the conversion degree, then the label node outputs “1” for the input data \vec{x} , otherwise outputs “0” instead.

3. Fuzzy domain adaptation based on self-constructing fuzzy neural network

The partition of input space determines the number of fuzzy rules generated. Each partition represents a specific rule and has a simple interpretation: it projects a Gaussian function on each dimension and generates a cluster based on the center and variance of the Gaussian function. As the source and target domain differ in distribution, the Gaussian functions will be different. If the difference is essential, an additional cluster may need to be added to in the target domain and the output will be computed with existing similar clusters, because it is intuitive to believe that the more close of two instances measured in a Gaussian function the more similar they are in the knowledge representation. Our method can be accomplished in three steps as described below.

Step 1. Learning initial structure and connecting parameters

In this stage, we adopt the same structure and parameter learning method proposed in Ref. 9. Based on labeled source domain instances D_s^l , initial network structure can be formed in an unsupervised way and supervised gradient decent is applied with the loss function

$$F = \frac{1}{2} \sum_{j=1}^{n_s} (y_l^d(\vec{x}_j) - y_l(\vec{x}_j))^2 \quad (5)$$

to learn optimal connecting parameters for each initial rule node and to upgrade both center and width for each fuzzy set membership function. $y_l^d(\vec{x}_j)$ and $y_l(\vec{x}_j)$ denotes the desired and actual label of input instance \vec{x}_j , respectively.

Step 2. Adjusting the initial structure with target domain instances

Given a new input \vec{x}_t in D_t^u , we can calculate the firing strength of this input to all existed rules in equation (3). If there is any a rule R_k that satisfies $E_k(\vec{x}_t) \geq T_D$, where T_D is a user defined threshold value, then the instance \vec{x}_t can get the output variable through those fired rules associated with those corresponding connecting parameters in equation(4) and finally obtain the data label. If not, a new rule node R_{new} and n new fuzzy sets

are added into the network, and the center and width of each new fuzzy set membership function is defined in Ref. 9 as following:

$$\mu_{(r+1)i} = \bar{x}_t^i, i = 1, \dots, n \quad (6)$$

$$\sigma_{(r+1)i} = \frac{-1}{\beta} \left(\frac{1}{\ln \arg \max_{1 \leq k \leq r} (E_k(\bar{x}_t))} \right), i = 1, \dots, n \quad (7)$$

where \bar{x}_t^i is the i th dimensional variable of input data \bar{x}_t and β controls the overlap degree between two fuzzy sets.

Step 3. Learning new connecting parameters

To learn the new connecting parameter for each newly generated rule node, we need to compare each new fuzzy set with all the existed ones. The fuzzy similarity measure in Ref. 9 is adopted:

$$S(A, B) = \frac{M | A \cap B |}{M | A \cup B |} = \frac{M | A \cap B |}{\sigma_1 \sqrt{\pi} + \sigma_2 \sqrt{\pi} - M | A \cap B |} \quad (8)$$

$$\begin{aligned} M | A \cap B | &= \sum_{x \in U} (\min[u_A(x), u_B(x)]) = \\ &= \frac{1}{2} \frac{h^2 [c_2 - c_1 + \sqrt{\pi}(\sigma_1 + \sigma_2)]}{\sqrt{\pi}(\sigma_1 + \sigma_2)} + \frac{1}{2} \frac{h^2 [c_2 - c_1 + \sqrt{\pi}(\sigma_1 - \sigma_2)]}{\sqrt{\pi}(\sigma_2 - \sigma_1)} \\ &+ \frac{1}{2} \frac{h^2 [c_2 - c_1 - \sqrt{\pi}(\sigma_1 + \sigma_2)]}{\sqrt{\pi}(\sigma_1 - \sigma_2)} \end{aligned} \quad (9)$$

where $S(A, B)$ represents the fuzzy similarity between two fuzzy sets A and B and $h(\cdot) = \max \{0, \cdot\}$. Membership functions of two fuzzy sets A and B are $\mu_A(x) = \exp \{ -(x - c_1)^2 / \sigma_1^2 \}$ and $\mu_B(x) = \exp \{ -(x - c_2)^2 / \sigma_2^2 \}$, respectively, and we assume $c_1 \geq c_2$.

Ranking on the fuzzy similarity value, the most similar existed fuzzy set associated with the most similar existed fuzzy rule can be picked out. We then calculate average weight as the new connecting parameter in the formula below:

$$\omega_{new} = \frac{\sum_{i=1}^n (S(A_{ik}, A_{i-new}) \times \omega_k)}{\sum \omega_k} \quad (10)$$

where A_{i-new} and A_{ik} denotes, respectively, the new fuzzy set and the most similar existed fuzzy set generated by the new input variable \bar{x}_t^i , and ω_k is the corresponding rule node's connecting parameter.

After the new connecting parameter has been fixed, the output value for the new input instance \bar{x}_t can be calculated in quation (4) based on the new network and the data label can be also obtained sequentially.

4. Conclusion and future work

Although there are tremendous studies in the field of domain adaptation, the role of information granularity is not yet investigated well. In this paper, a new fuzzy domain adaptation method based on self-constructing fuzzy neural network is proposed. It models the transferred knowledge from source domain to target domain in the form of fuzzy sets, and utilizes the fuzzy similarity measure to adapt knowledge for predicting target domain instances. Applying the proposed approach on the benchmark data sets, analysing experimental results and comparing with other domain adaptation methods will be the next step in this study.

Acknowledgments

The work presented in this paper was supported by the Australian Research Council (DP 140101366) and China Scholarship Council (No. 2013066020023).

References

1. B. Gong, Y. Shi, F. Sha and K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in *Computer Vision and Pattern Recognition, IEEE Conference on*, 2012.
2. R. Gopalan, R. Li and R. Chellappa, Domain adaptation for object recognition: An unsupervised approach, in *Computer Vision, IEEE International Conference on*, 2011.
3. J. Jiang and C. Zhai, Instance weighting for domain adaptation in nlp, in *Association for Computational Linguistics*, 2007.
4. M. Chen, K. Q. Weinberger and J. Blitzer, Co-training for domain adaptation, in *Advances in Neural Information Processing Systems 24*, eds. J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira and K. Weinberger 2011 pp. 2456–2464.
5. S. J. Pan, I. W. Tsang, J. T. Kwok and Q. Yang, *Neural Networks, IEEE Transactions on* **22**, 199 (2011).
6. M. Long, J. Wang, G. Ding, S. Pan and P. Yu, *Knowledge and Data Engineering, IEEE Transactions on* **PP**, 1 (2013).
7. V. Behbood, J. Lu and G. Zhang, *Industrial Informatics, IEEE Transactions on* **PP**, 1 (2013).
8. X. Yang, G. Zhang, J. Lu and J. Ma, *Fuzzy Systems, IEEE Transactions on* **19**, 105 (2011).
9. C.-T. Lin, C.-M. Yeh, S.-F. Liang, J.-F. Chung and N. Kumar, *Fuzzy Systems, IEEE Transactions on* **14**, 31 (2006).