Dynamical Ensemble Learning with Model-Friendly Classifiers for Domain Adaptation

Wenting Tu and Shiliang Sun
Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, China
w.tingtu@gmail.com, slsun@cs.ecnu.edu.cn

Abstract

In the domain adaptation research, which recently becomes one of the most important research directions in machine learning, source and target domains are with different underlying distributions. In this paper, we propose an ensemble learning framework for domain adaptation. Owing to the distribution differences between source and target domains, the weights in the final model are sensitive to target examples. As a result, our method aims to dynamically assign weights to different test examples by making use of additional classifiers called model-friendly classifiers. The model-friendly classifiers can judge which base models predict well on a specific test example. Finally, the model can give the most favorable weights to different examples. In the experiments, we firstly testify the need of dynamical weights in the ensemble learning based domain adaptation, then compare our method with other classical methods on real datasets. The experimental results show that our method can learn a final model performing well in the target domain.

1. Introduction

Recent years, domain adaptation becomes a hot topic [1, 2]. It arises when the data distributions in the training and test domain are different to each other. The need for domain adaptation research is prevalent in many real-world application problems. For example, training data collected from different user groups can have different but related patterns.

Moreover, the prospect of ensemble learning [3, 4] in the domain adaptation research should be paid attentions to. In domain adaptation, source domains often have some relations to the target domain but they are of different distributions. Therefore, the different base models constructed by source domains have good but not sufficient performance on data from the target domain. Ensemble learning is promising to be used to combine these base models to expect that the final model can perform well on the target domain since diversity among the members of a team of base models is deemed to be advantageous in ensemble learning. However, owing to the distributions differences between source and target domains, the target examples are sensitive to the weights to the source models.

In this paper, we present a novel ensemble-based approach for domain adaptation based on the dynamical weighting idea. Concretely speaking, firstly, a model group is constructed with datasets from source domains. The models in the group will be different to each other owing to the distribution differences among source domains. Then, for each base model, a model-friendly classifier will be trained for predicting whether a test example should be classified by this model. This is achieved by construct a model friendly training set whose positive examples are those that the model can predict correctly and negative examples are those that the model predicts wrongly. With the model-friendly training set, a classifier can be trained to point out which example the model can predict rightly. Finally, for each test example, the ensemble learner can obtain dynamical weights based on the output of the model-friendly classifiers. The main advantage of this method is to give more flexible weights to the test set, which aims to account for the distribution differences between training and test sets.

The remainder of this paper is organized as follows. In Section 2, we describe our method in detail. The next section shows the experimental results followed by the conclusion given in Section 4.

2. Our Method

Our method is motivated by the need of dynamical weighting in the ensemble learning based domain adaptation research. In domain adaptation research, training and test datasets are underlying different distributions, so that the test examples are sensitive to the weights in ensemble learner. Here, a ensemble learning with a dynamical weighting strategy is proposed to increase the generalization in test examples to enable the final model against to the dangerous of over-fitting. Our method, Dynamical Ensemble Learning with Model-Friendly Classifiers (DELMFC) includes three steps. We will discuss these steps in details.

2.1. Base-model Group Construction

First, with datasets from source domains, base models should be learned. In this step, there are many ways to construct enough base models. If the number of source domains is large enough, the training set from one source domain can contribute to constructing a base model. Otherwise, if the number of source domains is small, there are several ways to construct large base-model group.

$\mathbf{example-level}$ example-based ensemble learning means weak hypotheses are learned within the different example subspaces constructed by repeated random example selection or different example selection strategies.

$\mathbf{Feature-level}$ Similar as the example-based ensemble learning strategy, the feature-level means the differences of base models can be resulted from different feature subspaces constructed by different feature selection or extraction methods.

$\mathbf{Model-level}$ The most common way to learn different base models is to use different model-learning methods. For example, for classification tasks, the base models can be constructed by many different statistical models such as Support Vector Machine(SVM)[5], k-Nearest Neighbor (kNN)[6], Linear discriminative analysis (LDA)[7] and so on.

2.2. Model-friendly Classifiers Construction

In this step, another classifier group called model-friendly classifier group is constructed. Denote the base models constructed in Sec.2.1 as $M_{b_1}, \cdots, M_{b_n}$. In this section, model-friendly classifiers denoted as $C_{f_1}, \cdots, C_{f_n}$ are constructed. C_{f_i} can indicate which examples are suitable to be classified by M_{b_i}. C_{f_i} is constructed by two steps: Firstly, a model-friendly training set T_{f_i} is construct to train C_{f_i}. The positive examples in T_{f_i} are examples that the model M_{b_i} can rightly classify, and its negative examples are those M_{b_i} wrongly classify. That is, for a test example, if M_{b_i} can give it right prediction, the label of this example in model-friendly training set of C_{f_i} is positive. By this way, model-friendly training set T_{f_i} records the examples that M_{b_i} can predict rightly. Then, with T_{f_i}, C_{f_i} can be trained to predict which examples M_{b_i} can predict rightly.

2.3. Combination with Dynamical Weights

After performing steps in Sec.2.1 and Sec.2.2, we have base models $M_{b_1}, \cdots, M_{b_n}$ and their corresponding model-friendly classifiers $C_{f_1}, \cdots, C_{f_n}$. Now, for a test example x_{te}, suppose the output of M_{b_i} is V_{b_i} and the the output of C_{f_i} is V_{f_i} $(i = 1, 2, \cdots, n)$. Note that in our algorithm, V_{f_i} can be a probability or confidence value $\in [0, 1]$ or just a classification value $\in \{0, 1\}$. $V_{b_i} \in \{-1, 1\}$ is the prediction of base model C_{f_i} to the label of x_{te}. V_{f_i} indicates the probability of the fact that M_{b_i} is right. In other words, V_{f_i} can be seen as the confidence of V_{b_i}. The final prediction can be formulated as their multiplication.

To sum up, the algorithm is summarized in Algorithm 1.

3. Experiment

In this section, we take use of a dataset from BCI research to testify the effectiveness of our method. The reason of choosing this kind of data is that the prospect of using the domain adaptation theory to BCI research is always ignored by many researchers (As far as we know, only [8] and one public competition [9] paid enough attention to this issue) though inter-section and inter-subject nonstationaries of BCI data have been found already [10].

The EEG data used in this study were made available by Dr. Allen Osman of University of Pennsylvania during the NIPS 2001 BCI workshop [11]. There were a total of nine subjects denoted $S_1, S_2, \cdots, S_9$, respectively. For each subject, the task was to imagine moving his or her left or right index finger in response to a highly predictable visual cue. Here, in the base-model construction step, one dataset from one source domain contributes to three base model with SVM, kNN and LDA methods (kNN with $k = 7$, SVM with $C = 1$ and polynomial kernel). Therefore, for one target user, training sets from other eight source subject can construct 24 base models. Then, for each base model, a SVM-based model-friendly classifier is learned.

Algorithm 1 DELMFC

Training Session:
Input: source training sets $X_1, X_2, \cdots X_{n_s}$
Output: base model set $\mathbf{M_b}$
model-friendly classifier set $\mathbf{C_f}$
1. suppose we used n_1 example-level methods, n_2 feature-level methods, n_3 model-level methods. construct $n_s \times n_1 \times n_2 \times n_3$ base models: $M_{b_1}, \cdots, M_{b_n}$ $(n = n_s \times n_1 \times n_2 \times n_3)$.
2. **For** $i = 1, 2, \cdots, n$
construct model-friendly training set T_{f_i}:
For $x \in \{X_1, X_2, \cdots X_{n_s}\}$
IF M_{b_i} can predict x rightly
put x and its positive label into T_{f_i}
ELSE put x and its negative label into T_{f_i}
End of For
train model-friendly C_{f_i} with T_{f_i}
End of For

Test Session:
Input: the test example x_{te}
Output: the prediction of the label of x_{te}
For $i = 1, 2, \cdots, n$
obtain the output of M_{b_i} of x_{te}: V_{b_i}
obtain the output of C_{f_i} of x_{te}: V_{f_i}
End of For
$prediction = Sgn(\sum_i^n (V_{b_i} \cdot V_{f_i}))$

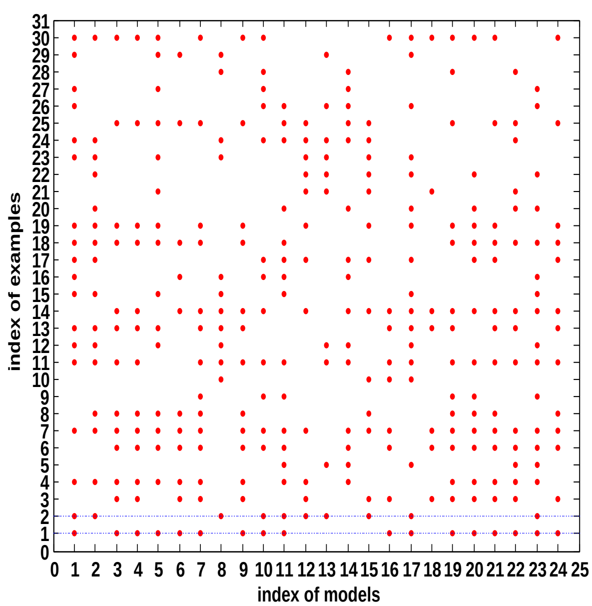

Figure 1. Preferences of base models on different target examples

In the experimental section, firstly, we show the performances of each base model to each test example to testify the need of dynamical weighting strategy in Fig.1. The x-axis presents the index of base models from source subject $S_2, S_3, \cdots, S_9$. The y-axis records index of random 30 examples of the target subject S_1. The red dots mean the corresponding base models can give the right prediction to the corresponding test examples. As Fig.1 shows, the base models performance differently on different test examples. For example, the base model sets that can rightly predict the labels of the first and second examples (as the blue broken lines indicate) are very different to each other. Therefore, facing different test examples, it is necessary to assign different weights to base models. Then, an experiment to compare our method with other methods is presented. Here, we compare our method with two classic methods in ensemble learning research and domain adaptation research, respectively. They are Adaptive Mixtures of Local Experts (AMLE) and Tradaboost (for details, see [12] and [13]). Table1 presents the classification accuracies of three methods, which indicate DELMFC can obtain good performance.

4. Conclusion and Future Work

This paper proposes a dynamical ensemble learning framework with model-friendly classifiers for domain adaptation. Our method includes three steps that are base-model group construction, model-friendly classifier construction and combination with dynamical weights. The experimental results show that it is necessary to assign dynamical weights in ensemble learning based domain adaptation research and our method can enhance the generalization ability on target domain. Note that, in this paper, we employ it for domain adaptation research. However, it is natural that employing our method in other scenarios such as semi-supervised learning[14] or unsupervised learning[15].

For the future research, the theoretical analysis of our method is a big challenge. Moreover, one of our method's advantages is that it can be used in many other real-world domains such as web-document classification[16], natural language processing[17] and image classification[18]. Showing our method's prospects in these tasks in detail will be given in our next work.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Project

Table 1. The classification accuracies (%) of DELMFC, AMLE, and Tradaboost methods.

Method	Target Subject				
	S_1	S_2	S_3	S_4	S_5
DELMFC	$\mathbf{70.22 \pm 2.4}$	$\mathbf{70.05 \pm 2.3}$	$\mathbf{70.01 \pm 2.1}$	$\mathbf{69.54 \pm 2.7}$	$\mathbf{69.06 \pm 2.2}$
AMLE	68.25 ± 2.2	67.17 ± 2.3	67.86 ± 2.9	67.43 ± 3.1	65.37 ± 2.1
Tradaboost	67.89 ± 2.3	66.69 ± 2.1	66.21 ± 2.8	64.83 ± 2.9	65.09 ± 2.2
Method	Target Subject				$Average$
	S_6	S_7	S_8	S_9	
DELMFC	$\mathbf{71.44 \pm 2.3}$	$\mathbf{71.95 \pm 2.3}$	$\mathbf{71.02 \pm 2.4}$	$\mathbf{70.94 \pm 2.2}$	$\mathbf{70.47 \pm 2.3}$
AMLE	69.50 ± 2.8	68.91 ± 2.3	67.56 ± 2.8	66.68 ± 2.4	67.64 ± 2.5
Tradaboost	68.91 ± 2.5	66.13 ± 2.1	65.97 ± 2.9	65.18 ± 2.4	66.32 ± 2.4

61075005, and the Fundamental Research Funds for the Central Universities.

References

[1] Daumé III, H. and Marcu, D.: Domain adaptation for statistical classifiers. Journal of Artificial Intelligence Research, 26(1), 101–126 (2006)

[2] Mansour, Y. and Mohri, M. and Rostamizadeh, A.: Domain adaptation with multiple sources. Advances in Neural Information Processing Systems, 21, 1041–1048 (2009)

[3] Opitz, D. and Maclin, R.: Popular ensemble methods: An empirical study. Arxiv preprint arXiv:1106.0257, (2011)

[4] Polikar, R.: Ensemble based systems in decision making. IEEE Circuits and Systems Magazine, 6(3), 21–45 (2006)

[5] Cortes, C. and Vapnik, V.: Support-Vector Networks. Machine Learning, 20(3), 273–297 (1995)

[6] Shakhna-rovich, G. and Darrell, T. and Indyk, P.: Nearest-Neighbor Methods in Learning and Vision. IEEE Transactions on Neural Networks, 19(2), 377 (2008)

[7] Abdi, H.: Discriminant correspondence analysis. Encyclopedia of measurement and statistics, 270–275 (2007)

[8] W. Tu and S. Sun: A subject transfer framework for EEG classification. Neurocomputing, 82, 109-116 (2012)

[9] Klami, A. and et al: ICANN/PASCAL2 Challenge: MEG Mind-ReadingłOverview and Results. Proceedings of the 20th International Conference on Artificial Neural Networks, 3 (2011)

[10] Blankertz, B. and et al: Invariant common spatial patterns: Alleviating nonstationarities in brain-computer interfacing. Advances in Neural Information Processing Systems, 20, 113–120 (2008)

[11] W. Dai, Q. Yang, G. R. Xue and Y. Yu. Boosting for transfer learning. Proceedings of the 24th International Conference on Machine Learning, 193-200 (2007)

[12] Jacobs, R.A. and Jordan, M.I. and Nowlan, S.J. and Hinton, G.E.: Adaptive mixtures of local experts. Neural computation, 3(1), 79–87 (1991)

[13] W. Dai, Q. Yang, G. R. Xue and Y. Yu.: Boosting for transfer learning. Proceedings of the 24th International Conference on Machine Learning, 193-200 (2007)

[14] Zhu, X.: Semi-supervised learning literature survey. Tech.Report 1530, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, (2006)

[15] Richard O. Duda, Peter E. Hart, David G. Stork: Unsupervised Learning and Clustering. Lecture notes on Learning Theory for Information and Signal Processing, Vienna University of Technology, 29, 30 (2006)

[16] Kosala, R. and Blockeel, H.: Web mining research: A survey. ACM Sigkdd Explorations Newsletter, 2(1), 1–15 (2000)

[17] Bates, M.: Models of natural language understanding. Proceedings of the National Academy of Sciences of the United States of America, 92(22), 9977–9982 (1995)

[18] Lu, D. and Weng, Q.: A survey of image classification methods and techniques for improving classification performance. International Journal of Remote Sensing, 28(5), 823–870 (2007)