Multi-view Transfer Learning with Adaboost

Zhijie Xu
Department of Computer Science and Technology
East China Normal University
Shanghai, P. R. China
zjxu09@gmail.com

Shiliang Sun
Department of Computer Science and Technology
East China Normal University
Shanghai, P. R. China
slsun@cs.ecnu.edu.cn

***Abstract*—Transfer learning, serving as one of the most important research directions in machine learning, has been studied in various fields in recent years. In this paper, we integrate the theory of multi-view learning into transfer learning and propose a new algorithm named Multi-View Transfer Learning with Adaboost (MV-TLAdaboost). Different from many previous works on transfer learning, we not only focus on using the labeled data from one task to help to learn another task, but also consider how to transfer them in different views synchronously. We regard both the source and target task as a collection of several constituent views and each of these two tasks can be learned from every views at the same time. Moreover, this kind of multi-view transfer learning is implemented with adaboost algorithm. Furthermore, we analyze the effectiveness and feasibility of MV-TLAdaboost. Experimental results also validate the effectiveness of our proposed approach.**

***Keywords*-transfer learning; adaboost; multi-view learning; classification;**

I. Introduction

Traditional machine learning usually depends on the availability of a large number of data from a single task to train an effective model. However, researchers often confront the situations that there are not enough data available and they have to resort to the data from other tasks to aid the learning of the target task. Owing to this reason, lots of new strategies have been proposed, including multi-task learning [1] and transfer learning [2]. In these methods, transfer learning is the one beginning to catch more attention in recent years. By this learning methodology, people can not only get more labeled data they want from source tasks, but also combine them into the target task to help train the model. In order to promote the effectiveness of transfer learning, now we plan to incorporate the adaboost algorithm [3] and the principle of multi-view learning into it.

Sometimes, although some data in source tasks are unsuitable for the target task, there may still exist some other data that can be useful for the target task. To find out this kind of data, we can employ adaboost algorithm to help us by voting on every datum in every iteration. In addition, different feature sets of data can exhibit a common underlying structure: they consist of a number of parts of factors, each with a range of possible states [4]. Therefore, through learning the same task from diverse views, we can get various kinds of knowledge which can exert different effects to the model as well.

In this paper, we present a new algorithm, Multi-View Transfer Learning with Adaboost (MV-TLAdaboost), by combining the advantages of multi-view learning and adaboost algorithm. The function of MV-TLAdaboost can be understood from the following two points. Firstly, whether one datum from source tasks can be reused in target task can be judged effectively. Secondly, some latent knowledge can be learned by learning one task from several views. In our algorithm, we make use of the theory of the Embedded Multi-View Adaboost algorithm (EMV-Adaboost) [5]. On the basis of this algorithm, we design some essential steps to construct a new parameter so as to adapt and indicate the characteristic of transfer learning. Next, with the help of this parameter, we update the weighting formula to make a difference for various data which contribute differently to the model. Then, we use the combination of multiple learners [6] to output the hypothesis. Analysis of our proposed algorithm and the experimental resutls can prove the feasibility and efficiency of MV-TLAdaboost.

The remainder of this paper is organized as follows. In Section 2, we introduce the general adaboost algorithm. Following this, in Section 3, we describe our MV-TLAdaboost algorithm in detail. Then, analysis of our algorithm is provided in Section 4. The next section shows the experimental results followed by the conclusion given in Section 6.

II. Adaboost

Adaboost is a general supervised learning technique for incrementally building linear combinations of weak learners to generate a strong predicative model [7]. In this algorithm, we denote the input data set as $X = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ where x_i belongs to a domain D and y_i belongs to the class label set $Y = \{0, 1\}$. Next, we provide a weight set $W = \{w_1, w_2, \cdots, w_n\}$ for samples in this data set, and initialize them by $1/N$, where N is the total number of samples in X. Then, we start the iteration for T times with the distribution P of samples, which can be obtained at the beginning of every iteration with the help of W. Following this, the algorithm outputs one weaker learner h_t for the training step in every iteration. Furthermore, the weight set W needs to be updated by the parameter β_t which is composed by the error rate ε_t of the weaker learner h_t. Finally, the set of weaker learners $H = \{h_1, h_2, \cdots, h_T\}$ are combined by weighted majority voting into the final learner, which performs better than any of the weaker learners. (In our study, $T = 30$.)

The detailed algorithm is given in the table below.

```
Algorithm Adaboost
Input:
Set of $n$ labeled examples:
$X = (x_1, y_1), \cdots, (x_n, y_n)$
Distribution $D$ over the $n$ examples
Integer $T$ specifying number of iterations
----------------------------------------
Initialize the weight vector: $w_1^i = D(i)\,\text{for}\,i = 1, \cdots, n$.
For $t = 1, 2, \cdots, T$
1. Set $P^t = \frac{W^t}{\sum_{i=1}^{n} w_i^n}$
2. Call WeakLearn with distribution $P^t$, and then get
   back the hypothesis below:   $h_t : X \to \{0, 1\}$
3. Calculate the error of $h_t : \varepsilon_t = \sum_{i=1}^{n} p_i^t |h_t(x_i) - y_i|$
4. Set $\beta_t = \frac{\varepsilon_t}{(1-\varepsilon_t)}$
5. Set the new weights vector to be:
   $w_i^{t+1} = w_i^t \beta_t^{1-|h_t(x_i)-y_i|}$
End of For
Output the hypothesis:
$h_f(x) = \begin{cases} 1, & \text{if } \frac{\sum_{t=1}^{T} (\log 1/\beta_t) h_t(x)}{\sum_{t=1}^{T} \log 1/\beta_t} \geq \frac{1}{2} \\ 0, & \text{otherwise} \end{cases}$
```

III. The Proposed Method

A. Overview

For the purpose of making fully use of both transfer learning and multi-view learning in adaboost, we design a new algorithm, MV-TLAdaboost, to integrate them. Now we try to apply the theory of transfer learning into one multi-view adaboost algorithm named EMV-Adaboost [5]. In our algorithm, on the basis of the characteristic of transfer learning, we change the original EMV-Adaboost algorithm by a new parameter and some equations. The detailed algorithm is given in the table named Algorithm MV-TLAdaboost below.

B. MV-TLAdaboost

According to our algorithm, it needs to make a knowledge about EMV-Adaboost firstly, some main steps about it have been described from step 1 to step 6. In step 2, we will come to two weaker learners in every iteration by two views: $V1$ and $V2$, which are formed at the beginning of the whole algorithm. After that, in step 3, this algorithm assumes that one sample will contribute to the error rate as long as it is predicted incorrectly in either of the weaker learners got in step 2.

```
Algorithm MV-TLAdaboost
Input:
1. Set of $n$ labeled target examples:
   $X = (x_1, y_1), \cdots, (x_n, y_n)$
2. Set of $m$ labeled source examples:
   $Z = (x_{n+1}, y_{n+1}), \cdots, (x_{n+m}, y_{n+m})$
3. Weight distribution $D$ over the $n + m$ examples
4. Integer $T$ specifying number of iterations
----------------------------------------
Divide all of the features into two views: $V1$ and $V2$
Initialize the weight vector:
$w_1^i = D(i)\,\text{for}\,i = 1, \cdots, n + m$.
For $t = 1, 2, \cdots, T$
1. Set $P^t = \frac{W^t}{\sum_{i=1}^{n+m} w_i^{n+m}}$
2. Call WeakLearn with distribution $P^t = \{p_1^t, \cdots, p_{n+m}^t\}$,
    and then get back two weaker learners from two views:
   $h_t = \{h_t^{V1}, h_t^{V2}\} : X, Z \to \{0, 1\}$
3. Calculate the error about target data set by $\{h_t^{V1}, h_t^{V2}\}$ :
   $\varepsilon_t = \sum_{i=1}^{n} p_i^t \{max\{|h_t^{V1}(x_i) - y_i|, |h_t^{V2}(x_i) - y_i|\}\}$

   Note that, $\varepsilon_t$ is required to be less than $1/2$
4. Calculate the percentage of the samples predicted
   same by two weaker learners  :
   $agree_t = 1 - \frac{\sum_{i=1}^{n+m} |h_t^{V1}(x_i) - h_t^{V2}(x_i)|}{n+m}$
5. Set $\epsilon_t = \varepsilon_t\, agree_t$
6. Set $\beta_t = \frac{\epsilon_t}{(1-\epsilon_t)}$
7. Set $\beta = 1/(1 + \sqrt{2 \ln m/T})$
8. Calculate the distribution of $Xonly$ :
   $R^t = \frac{w_t^i}{\sum_{j=1}^{n} w_j^n} \; for\ i = 1, \cdots, n$
9. Calculate the accuracy rate of $X$ with $\{h_t^{V1}, h_t^{V2}\}$ under its
   distribution $R^t = \{r_1^t, \cdots, r_n^t\}$ :
   $Acc_t^X = \sigma_t =$
   $\sum_{i=1}^{n} r_i^t \{1 - max\{|h_t^{V1}(x_i) - y_i|, |h_t^{V2}(x_i) - y_i|\}\}$
10. Calculate the overall accuracy rate with $\{h_t^{V1}, h_t^{V2}\}$ :
   $Acc_t^O = 1 - \varepsilon_t$
11. Set $\eta_t = \frac{Acc_t^O}{Acc_t^X}$
12. Set the new weights vector to be:
$w_i^{t+1} =$
$\begin{cases} w_i^t (\beta_t \eta_t)^{-max\{|h_t^{V1}(x_i)-y_i|, |h_t^{V2}(x_i)-y_i|\}}, & 1 \leq i \leq n \\ w_i^t \beta^{max\{|h_t^{V1}(x_i)-y_i|, |h_t^{V2}(x_i)-y_i|\}}, & \text{others} \end{cases}$
End of For
Output the hypothesis:
$h_f(x) = \begin{cases} 1, & \text{if } \sum_{t=1}^{T} \sum_{i=1}^{2} h_t^{Vi}(x) \geq T \\ 0, & \text{otherwise} \end{cases}$
```

Next, from step 4 to step 6, EMV-Adaboost designs a new parameter $agree_t = 1 - \frac{\sum_{i=1}^{n+m} |h_t^{V1}(x_i) - h_t^{V2}(x_i)|}{n+m}$ to indicate the percentage of the samples predicted same by them. At the same time, it defines another new parameter, ϵ_t to combine the ε_t and $agree_t$ together in order to reflect the value about each view and adjust the parameter β_t.

According to the process of transfer learning in our algorithm, we generally have two data sets (X and Z) coming from different tasks and Z always contains much more samples than X. After solving the problem of similarity between them which is not the research emphasis in our paper, we need to make usage of them together to train the weaker learners by two views, h_t^{V1} and h_t^{V2} in every iteration. However, even though the problem of similarity has been solved, X and Z can still possibly possess a small quantity of their own special knowledge respectively. As a result, it is necessary for us to consider one important problem here: these weaker learners got here may tend to indicate the characteristics of Z further more because this data set has much more samples than X.

Due to the above reason, we believe that these kinds of weaker learners can lead to ambiguity to data set X and can not show the characteristic of X thoroughly. Consequently, we hope to reach the following effect: these weaker learners can make a difference between data set X and Z in every iteration. Simply speaking, at the first several iterations in our algorithm, they need to make a less effect on X than Z. Then, with the development of our algorithm, this kind of difference will come to fading resulting from the integration of them. In order to reach this goal, we design the parameter η_t in the equation below:

$$\eta_t = \frac{Acc_t^O}{Acc_t^X}. \tag{1}$$

In (1), Acc_t^X calculate the accuracy rate of X under its own distribution R^t. Acc_t^O calculate the overall accuracy rate under the distribution P^t. The detailed formulas have been written as:

$$Acc_t^X = \sigma_t = \sum_{i=1}^{n} r_i^t \{1 - max\{|h_t^{V1}(x_i) - y_i|, |h_t^{V2}(x_i) - y_i|\}\}. \tag{2}$$

$$Acc_t^O = 1 - \varepsilon_t. \tag{3}$$

Actually, η_t indicates the ratio of overall accuracy rate to the partial accuracy rate (data set X of target task). Because of the ambiguity according to data set X mentioned above, the classification accuracy of it will be less than the one of overall data set. Then we can get the inequality, $Acc_t^X \leq Acc_t^O$. As a result, $\eta_t \geq 1$ can be reached. In addition, with the disappearance of the ambiguity about X in the process of iterations, we think that the experimental core between these two data sets will become balanceable and weaker learners will take the same effect on X and the overall data set finally. This means η_t gradually approaches to one and below equation can be reached at last.

$$\beta_t = \beta_t \eta_t. \tag{4}$$

After explaining the function of η_t, now we use it to update the weight of samples by (5).

$$w_i^{t+1} = \left\{ \begin{array}{ll} w_i^t(\beta_t\eta_t)^{-max\{|h_t^{V1}(x_i)-y_i|,|h_t^{V2}(x_i)-y_i|\}}, & 1\leq i\leq n \\ w_i^t\beta^{max\{|h_t^{V1}(x_i)-y_i|,|h_t^{V2}(x_i)-y_i|\}}, & \text{others}. \end{array} \right. \tag{5}$$

In the process of the above step, we use different formula to update the weights of the samples from X and Z and it is essential for us to consider two factors. Firstly, data set X coming from target task is more representative than Z in our experiment. Thus, we want to make it prominent so that we increase the weight of the wrongly classified samples in it and keep the weight of the correctly ones. What is more, not all of the samples in source data set Z are useful and helpful for the model. Therefore, we hope there exists one screening process when updating the weight of samples in Z to help us weaken the function of the samples which may make a negative impact on the outcome. This goal has been described explicitly in (5).

$$\beta_t \eta_t = \frac{\epsilon_t}{1 - \epsilon_t} \frac{1 - \varepsilon_t}{\sigma_t}. \tag{6}$$

Secondly, for the purpose of alerting the function of η_t which can control the influence to the samples in X, we combine it with parameter β_t to update the weight vector. Moreover, owing to the fact of $\epsilon_t = \varepsilon_t agree_t \leq \varepsilon_t$, $\varepsilon_t \leq \frac{1}{2}$ and $\sigma_t \geq \frac{1}{2}$, we can get the following inequality with the help of (1), (3), (4) and (6).

$$\beta_t \leq \beta_t \eta_t \leq 1. \tag{7}$$

Equation.(7) can help us to ensure that η_t controls the proportion of the update about the weights of wrongly predicted samples in X.

Finally, in the output step, we use the approach of majority vote by all of weaker learners together to construct final learner (In our study, the number of weaker learners is $2{\times}T = 60$). Our hypothesis shows that if at least half of weaker learners predict the sample as label 1, the sum of the outcome made by weaker learners will be larger than T.

IV. Analysis of MV-TLAdaboost

In the above section, we describe our multi-view transfer classification learning framework in detail. In this section, we turn to the theoretical analysis about its error rate on the basis of the theory of adaboost [3].

Theorem 1 *Suppose the weak learning algorithm WeakLearn, when called by MV-TLAdaboost, generates hypotheses with error $\varepsilon_1, \cdots, \varepsilon_T$. Then, let $I = \{i : h_f(x_i) \neq y_i \text{ and } 1 \leq i \leq n\}$. The prediction error on the target data set, $\varepsilon = Pr_{x_i \in X}[h_f(x_i) \neq y_i]$ of the final hypothesis h_f, is bounded by*

$$\varepsilon \leq \prod_{t=1}^{T} (1 - (1 - \varepsilon_t)(1 - \beta_t \eta_t)). \tag{8}$$

The detailed proof is the same as in [5]. In addition, compared to following bound of ε got in [3], it can be seen clearly that our prediction error on target data set reduces the error on the target data set.

$$\varepsilon \leq \prod_{t=1}^{T} \frac{(1-(1-\varepsilon_t)(1-\beta_t))}{\sqrt{\beta_t}}. \tag{9}$$

According to the above Theorem, we can know intuitively that MV-TLAdaboost first performs its learning on the target data set, and then chooses the most useful data from source data set to improve the outcome.

V. Experiments and Results

In this section, we implement three groups of experiments of binary classification problems about three data sets from UCI repository. Table I demonstrates the summary of these data sets.

Table I
Summary of real data sets

Real data sets	*German*	*Landsat*	*Wpbc*
Total number of samples	1000	2866	569
Samples in source task	630	1817	226
Train samples in target task	185	839	257
Test samples in target task	185	210	86
Dimension	24	36	30
Number of classes	2	2	2

For every data set above, we design one special rule to divide it into two sub data sets as target task and source task which contain diverse distributions. For German Credit Data, we split the data set based on the feature *foreign worker*. The target task consists of all the samples with answer no, while the source task consists of all the samples with answer yes.

Similar to German Credit Data, in Landsat Satellite, we split the data set based on the feature *spectral values*. All the samples according with the rule spectral values ≥ 100 belong to target task and others for source task. It also needs to notice in this data set that we extract all the data belonging to classes *grey soil* and *very damp grey soil* to form one binary data set for our experiment here. For Wpbc data set, we split the data set based on the feature *Mean Radius*. We calculate the mean value, M of this feature of all the samples and supply the samples according with the rule Mean Radius$\leq$ M for target task and others for source task.

In the experiments of MV-TLAdaboost, due to the reason that we divide the dimensions about every data set into two views in half randomly, we run the experiments for ten times and get the mean of them as the final scores. Certainly, standard deviation (Std) will be calculated synchronously. Table II gives the classification results.

Table II indicates clearly that our proposed algorithm, MV-TLAdaboost, comes to the best outcome in every data set.

Table II
Classification error rate ($Mean \pm Std$)

	Adaboost	TrAdaboost	MV-TLAdaboost
German	0.2973	0.2541	0.2151±0.0137
Landsat	0.0333	0.0381	0.0209±0.0046
Wpbc	0.0465	0.0233	0.0105±0.0086

VI. Conclusion and Future Work

In this paper, we propose the algorithms, Multi-View Transfer Learning with Adaboost (MV-TLAdaboost). It makes full use of the characteristics of multi-view learning and adaboost algorithm to improve the effectiveness of transfer learning. By experiments on binary classification problems, this algorithm proves to be quite useful and effective. More importantly, appropriate theory analysis helps us believe its reliability and feasibility.

In the future, it can be an interesting challenge to extend the proposed MV-TLAdaboost algorithm to cope with more than two views.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Project 61075005, and the Fundamental Research Funds for the Central Universities.

References

[1] F. Jin and S. Sun, A multitask learning approach to face recognition based on neural networks. In Proceedings of 9th Intelligent Data Engineering and Automated Learning, pp 24-31, 2008.

[2] R. Raina, A. Y. Ng and D. Koller, Constructing informative priors using transfer learning. In Proceedings of the 23rd International Conference on Machine Learning, pp 713-720, 2006.

[3] Y. Freund and R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Science, 55: 119–139, 1997.

[4] D. Ross and R. Zemel, Learning parts-based representations of data. Journal of Machine Learning Research, 7: 2369-2397, 2006.

[5] Z. Xu and S. Sun, An algorithm on multi-view adaboost. In Proceedings of 17th International Conference on Neural Information Processing, pp 355-362, 2010.

[6] Q. Zhang and S. Sun, Multiple-view multiple-learner active learning. Pattern Recognition, 43(9): 3113-3119, 2010.

[7] G. Haffari, Y. Wang, S. Wang, G. Mori and F. Jiao, Boosting with incomplete information. In Proceedings of the 25th International Conference on Machine Learning, pp 368-375, 2008.