Journal of Fiber Bioengineering and Informatics 8:1 (2015) 105–115
doi:10.3993/jfbi03201510

Ensemble Inductive Transfer Learning ⋆

Xiaobo Liu [a,*], Guangjun Wang [a], Zhihua Cai [b], Harry Zhang [c]

[a] *School of Automation, China University of Geosciences, Wuhan 430074, China*

[b] *School of Computer Science, China University of Geosciences, Wuhan 430074, China*

[c] *Faculty of Computer Science, University of New Brunswick, Fredericton, NB, E3B5A3, Canada*

Abstract

Inductive transfer learning is a major research area in transfer learning which aims at achieving a high performance in the target domain by inducing the useful knowledge from the source domain. By combining decisions from individual classifiers, ensemble learning can usually reduce variance and achieve higher accuracy than a single classifier. In this paper, we propose a novel Ensemble Inductive Transfer Learning (EITL) method. EITL builds a set of classifiers by recording the iterative process of knowledge transfer. In each iteration, it uses the classifier of the source domain, the base classifier of the target domain built on the initial labeled data, and the most recent classifier built on the updated labeled data, to classify unlabeled instances, and add some self-labeled instances to the labeled data, and then trains a new classifier. At the end, all the classifiers built in this process are used for classification. We conduct experiments on synthetic data sets and six UCI data sets, which show that EITL is an effective algorithm in terms of classification accuracy.

Keywords: Transfer Learning; Ensemble Learning; Machine Learning

1 Introduction

In the machine learning field, a major challenge is that labeled data is easy to outdate, and data labeling is often expensive and time-consuming. One direct consequence is the lack of training data, which may result in an unsatisfactory performance by using the traditional machine learning methods to construct a model. Moreover, the feature space or distributions may change over time. Transfer learning is a new research area in machine learning, which provides a new approach to tackle this issue. Transfer learning reuses the useful certain parts of auxiliary data sets to train a classifier for the new data, although the auxiliary data sets may have different feature spaces or different distributions [12]. We called the outdated data sets or the auxiliary data sets as the source domain data sets. Usually, there are a variety of approaches to transfer knowledge from source domain to target domain, such as transferring some useful instances [3], feature

⋆Project supported by the Key Project of the Natural Science Foundation of Hubei Province, China (No. 2013CFA004) and the National Natural Science Foundation of China (No. 61403351).

*Corresponding author.
Email address: xbliu@cug.edu.cn (Xiaobo Liu).

representation [13], parameters [2], or relational knowledge [11]. In our method, we utilize the source domain to help the target domain to label the unlabeled data in the target domain.

Ensemble approach is usually more reliable than single approach to make classification [17]. Our motivation is that, we construct a set of classifiers and then classify unlabeled data by taking a vote on their predictions [4]. The initial ensemble includes two distinct classifiers built on the source domain and the target domain with a few initial labeled data, which are used for partly classifying the unlabeled data in target domain. The resulting labeled data is combined with the initial labeled data in the target domain to create a third classifier of the ensemble. The third classifier is updated with newly labeled data at each successive iteration. After many iterations, we have generated a set of classifiers for the target domain. Then these classifiers are used to predict the labels of test data by the majority vote strategy [8].

The main contributions of this paper are: (1) We use the ensemble method to repeatedly label the unlabeled data in the target domain, which can improve the performance of the target domain's task – classification. Meanwhile, the source domain takes part in the decision at each iteration, which can always play a role in the ensemble. (2) Our method is useful to do transfer learning. When only a few labeled data is given in the target domain, our method successfully use the source domain to help the target domain to label the unlabeled data in the target domain. Thus, the smaller the number of labeled data in the target domain is, the more useful our method is.

The rest of this paper is organized as follows: Section 2 presents related work about transfer learning and ensemble machine learning. The detailed introduction of our improved algorithm is described in Section 3. Section 4 displays the experiments on synthetic data sets and six UCI data sets, and analyzes the results. And Section 5 gives the conclusions of our work.

2 Related Work

The source domain data usually has a different distribution from the target domain data. If we reuse the source domain data directly, it will be unfeasible. If we discard the source domain data completely, the few labeled instances in the target domain are insufficient to train a good classifier. How to deal with those situations is the major challenge in transfer learning.

Dai et al. [3] proposed the TrAdaBoost method which iteratively re-weights the source domain data to reduce the effect of the *bad* source data while encouraging the *good* source data to contribute more to the target domain. Shi et al. [14] proposed a framework to actively transfer the knowledge from the source domain to help learn the target domain, and query experts only when necessary. Jiang et al. [6] proposed a general instance weighting framework for domain adaptation, which removed *misleading* source domain instances and added labeled target domain instances with higher weights. Liao et al. [9] proposed *Migratory-Logit* algorithm which is a new active learning approach for selecting the labeled examples in a target domain.

Constructing a good ensemble of classifiers has been an active research area in machine learning [4]. By combining decisions from individual classifiers, ensembles can usually reduce variance and achieve higher accuracy than an individual classifier.

Kamishima et al. [7] proposed a TrBagging method which is an extension of bagging. In order to reuse certain parts of the data in the source domain to benefit the learning in the target domain, the authors combined the source domain data sets and the target domain data sets, used

a bootstrap-sampled approach to train many weak classifiers, then created an ensemble based on their usefulness for the target domain.

Gao et al. [5] presented a locally weighted ensemble framework to combine multiple models for transfer learning, where the weights are dynamically assigned according to a model's predictive power on each test example. By mapping the structures of models onto the structures of the test domain, each model is weighted locally according to its consistency with the neighborhood structure around the test example.

In [16], it is assumed that the training and the test examples are generated from a mixture of different models, and the test distribution has different mixture coefficients than the training distribution. In [10], Marx et al. proposed an algorithm through a simple maximum a posteriori elaboration on the logistic regression approach demonstrating that in the transfer learning an ensemble of background tasks is more helpful than single background task. Bennett et al. [1] proposed a methodology for building a meta-classifier which combines multiple distinct classifiers through the use of reliability indicators.

In this paper, we adopt a different strategy to build an ensemble classifier and label the unlabeled data in the target domain. Our main ideas to the task of the transfer learning problem are briefly described as follows:

(1) Knowledge transfer is an iterative process. In each iteration, typically, only part of the useful knowledge from the source domain is transferred successfully. So it is hard to use a single classifier to represent this iterative process. On the other hand, we can use a classifier to record the transfer in each iteration, and then build an ensemble classifier at the end. It is a natural idea to do so.

(2) Knowledge transfer is also a process with high variance. Because the source and target domains have different distributions or even different feature spaces, any automatic transfer from the source domain to the target domain has a high risk or high variance. The ensemble approach can be used to alleviate the issue of high variance to some extent.

3 Ensemble Inductive Transfer Learning

3.1 Definition

In our setting, a lot of labeled data in the source domain are available, and a few labeled data in the target domain are available, but a large number of unlabeled data remain in the target domain. In order to easily understand our algorithm, we follow the notations of transfer learning problem presented in [12].

The source domain data is represented as D_S and the target domain data as D_T. The condition $D_S \neq D_T$ implies that the source domain and the target domain have different distributions which could be denoted as $P_S(X) \neq P_T(X)$. $\mathcal{X}$ is the space of features; $\mathcal{Y}$ is the space of labels. X is a particular learning instance, x_i is the ith vector in a instance. The source domain data is described as $D_S = \{(x_{S_1}, y_{S_1}), ..., (x_{S_{m_S}}, y_{S_{m_S}})\}$, where $x_{S_i} \in X_S$ is the data instance and $y_{S_i} \in \mathcal{Y}_S$ is the corresponding class label. The target domain includes two parts: one part is a few labeled data $D_{T_L} = \{(x_{T_1}, y_{T_1}), ..., (x_{T_{n_T}}, y_{T_{n_T}})\}$, where $x_{T_i} \in X_T$ is the data instance and $y_{T_i} \in \mathcal{Y}_T$ is the corresponding class label; another part is unlabeled data $D_{T_U} = \{(x_{T_{n_T+1}}, ..., x_{T_{k_T}}\}$. And we have a test data set D_{Test} which has the same distribution with the target domain.

3.2 Ensemble Construction

Because the labeled data in target domain, D_{T_L}, is too scarce to build a high-quality classification model when evaluated on the test data, we have to increase the number of labeled instances.

Although we have a large number of source data in D_S, the two domains have different distributions. So we use source classifier C_S, target classifier C_{T_0}, and a new classifier $C_{T_{k-1}}$ built on the expanded target data, to learn classifiers in the ensemble.

Algorithm 1 Framework of EITL

Input: the source domain data D_S, the labeled target domain data D_{T_L}, the unlabeled target domain data D_{T_U}, a base learning algorithm, and the maximum number of iteration K.
Initialization: build the source classifier C_S based on D_S and the target classifier C_{T_0} based on D_{T_L}.
for $k \in 1, ..., K$ **do**
  repeat
    for $i \in 1, ..., Max(D_{T_U})$ **do**
      repeat
        Assign label $label_s$ to $instance(i)$ by classifier C_S;
        Assign label $label_t$ to $instance(i)$ by classifier C_{T_0}.
        if (k==1) **then**
          if ($label_s$==$label_t$) **then**
            $D_{T_L} = D_{T_L} \bigcup instance(i)$;
          end if
        else
          Assign label $label_{k-1}$ to $instance(i)$ by
          the newly-built classifier $C_{T_{k-1}}$.
          if ($label_s == label_{k-1}$ **and** $label_t == label_{k-1}$) **then**
            $D_{T_L} = D_{T_L} \bigcup instance(i)$;
          end if
        end if
      until the maximum number of instances in D_{T_U}
    end for
    build C_{T_k} on newly obtained D_{T_L}.
  until the maximum number **K** is reached or the target domain has no change.
end for
Output the classifiers C_{T_1}, C_{T_2}, ..., C_{T_K}.

For repeatedly labeling the unlabeled instances in D_{T_U}, an unlabeled instance can be labeled as long as the three classifiers C_S, C_{T_0}, and $C_{T_{k-1}}$ agree on the labeling of this instance, while the confidence of the labeling of the classifiers is not needed to be explicitly measured. For example, if C_S, C_{T_0}, and $C_{T_{k-1}}$ agree on the labeling of an instance x in D_{T_U}, then x can be labeled by $C_{T_{k-1}}$. It is obvious that in such a scheme if the prediction of the three classifiers on x is correct, then D_{T_L} will receive a valid new example for further training; otherwise D_{T_L} will get an example with a noisy label. However, even in the worst case, the increase in the classification noise rate can be compensated if the amount of newly labeled examples is sufficient, which has been demonstrated in [18].

The main flow of the proposed approach EITL to build an ensemble classifier is summarized in Algorithm 1. It could be summarized as the following four parts.

Step 1: Initialization: At the beginning, we build the source classifier C_S and the target classifier C_{T_0} on D_S and D_{T_L} respectively, and utilize C_S and C_{T_0} to label the data in D_{T_U}. We add the newly labeled data into D_{T_L}. Then, we build a new classifier C_{T_1} based on the newly updated labeled data set D_{T_L}.

Step 2: Improvement: Use classifiers C_S, C_{T_0}, and classifier $C_{T_{k-1}}$ that is built on the newly updated labeled data set, to classify unlabeled instances. An unlabeled instance is labeled only when the three classifiers agree on the labeling of this instance. The newly labeled instances are added to the labeled data set, and new classifier C_{T_k} is built on the newly updated labeled data set.

Step 3: Stopping criterion: Two strategies are adopted to stop our algorithm. If the number of iterations reaches the maximum number K, the algorithm stops; or if the labeled data set in the target domain does not change anymore after a iteration, it terminates.

Step 4: Classification: For a given test data instance, we calculate the prediction of every classifier C_{T_k} that is built in each iteration, then we use the majority vote strategy to assign a label to the test data.

4 Experiments

In this section, we demonstrate the effectiveness of the ensemble transfer learning method EITL. The experiments are performed on synthetic data sets from [14] and six UCI Machine Learning Repository data sets[1].

Without loss of generality, we choose two classic classifiers in machine learning, Naive Bayes (NB) and the decision tree learning algorithm J48 [17], as the underlying learners. We set the maximum number of iteration K is five.

In the experiments, we compare EITL with the classifier built on the initial labeled data in target domain (D_{T_L}) which is the "basic" method in the following sections.

4.1 Synthetic Data Sets

The synthetic data sets include five two dimensional data sets which are generated in [14]. The target domain has one data set D_T. The source domain has four data sets with different distributions separately. "Transfer dataset" O_1 is similarly distributed as D_T; "Partly dataset" O_2 has some similarity distribution as D_T; "Different dataset" O_3 is *XOR* distribution; "Reverse dataset" O_4 has a similar *shape* as D_T, but with reversed class labels.

Fig. 1 plots the performance comparison of EITL vs. basic method learned on the four source domain data sets; both are based on NB. It is important to note that, both of the two methods have good convergency. The number of test data is 100, the number of labeled data in D_{T_L} varies from 2 to 60, and the remainder of D_T are unlabeled. From Fig. 1, it could be observed that:

(1) The error rates given by EITL significantly drop when there are only a few labeled data in

[1]http://archive.ics.uci.edu/ml/datasets.html

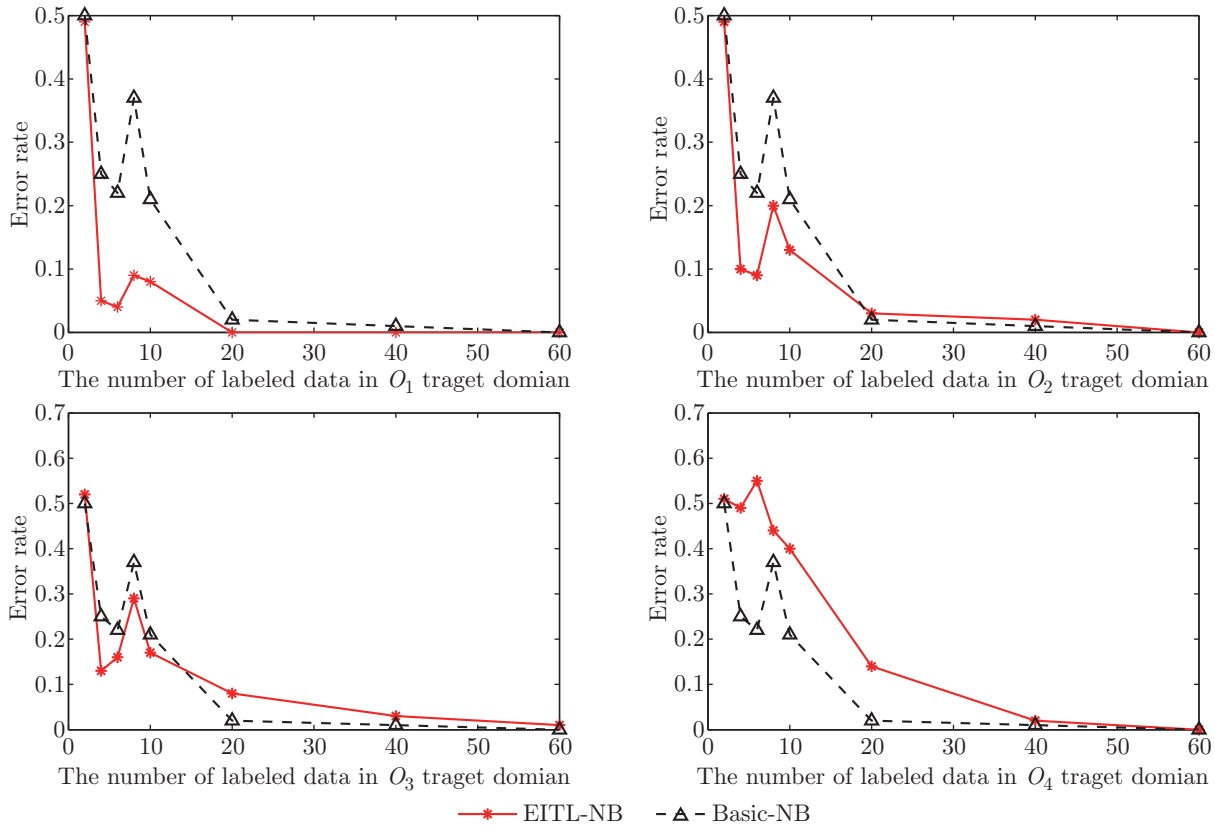

Fig. 1: The classification error rates of NB based EITL vs. NB based basic method on test data, using different number of labeled data in target domain of the synthetic data sets

D_{T_L}. Particularly, when the size of D_{T_L} varies from 2 to 4, the average error rates decreased by 0.31 on four source domain datasets.

(2) EITL has similar performance as the basic method when the size of D_{T_L} is small, however, EITL performs consistently better than the basic method when the size of D_{T_L} is less than 20, which means that EITL can effectively utilizes the source domain knowledge to help label the unlabeled data in the target domain.

(3) O_4 has a similar distribution but totally reversed class labels with the target domain D_T, which causes negative transfer. Although the error rate of EITL is higher than that of the basic method, it can still converge to zero when the number of D_{T_L} reached to 60.

4.2 UCI Data Sets

The data sets *Mushroom*, *Waveform*, *Magic*, and *Splice* are split following the method in [15]. In [15], the class labels are binary. In order to demonstrate that our algorithm can solve multi-class problems, we add *Hypothyroid* and *Segment* data sets to our experiments.

In *Hypothyroid* data set, for each instance, if the value of the second attribute *Sex* is "female", the instance is added to the target domain data set; otherwise, it is added to the source domain data set. In *Segment* data set, for each instance, if the value of the first attribute is larger than 127.5, it is added to target domain data set; otherwise, it is added to source domain data set.

Information on these data sets is tabulated in Table 1. The second column is the number of attributes. The third column is the number of class values. The fourth column is the number of test instances. And the fifth column "source/target" presents the number of source data versus that of target data.

Table 1: The information of the six data sets

Data set	Attri.	Cla.	Test	Sour./targ.
Mushroom	22	2	1000	4608/3516
Waveform	21	2	500	2279/1025
Magic	10	2	1000	16808/2212
Splice	60	2	1000	1571/1619
Hypothyroid	30	4	1000	1142/2629
Segment	20	7	500	1206/1103

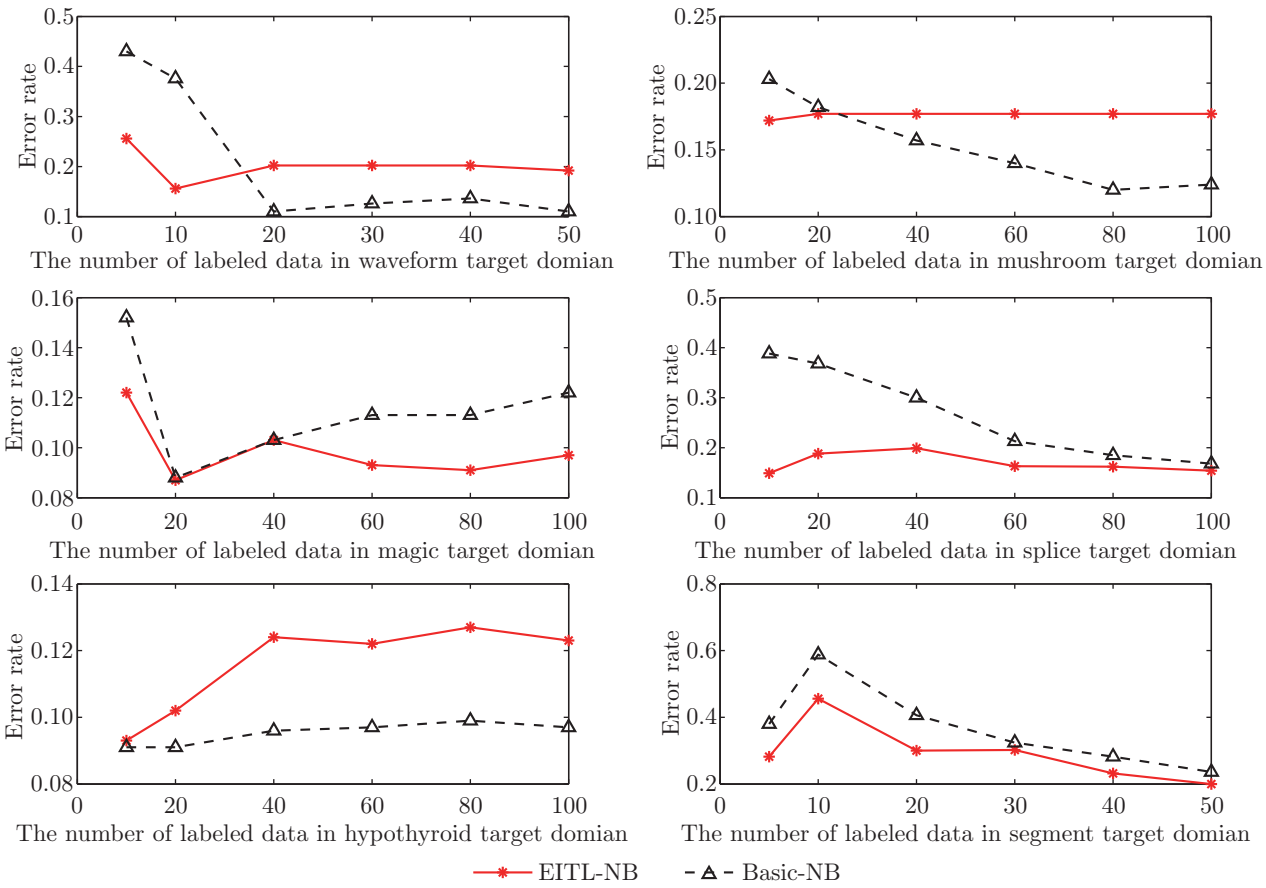

Fig. 2: The classification error rates of NB based EITL vs. NB based basic method on test data, using different numbers of labeled data in target domain of the six UCI data sets

The error rates associated with different sizes of initial labeled data in D_{T_L} are summarized in Table 2. To test the performance of each learning method in different conditions, the size of D_{T_L} is changed from 10 to 500 in *mushroom*, *magic*, *splice*, and *hypothyroid* data sets, and it is increased from 5 to 250 in *waveform* and *segment* data sets. It is observed that, interestingly, the classifier error rates are inversely related to the size of D_{T_L} in almost all the cases.

Table 2: The experimental results of six UCI data sets

(a)

Data set	Learner	Method	The number of labeled data in D_{T_L}							
			10	20	40	60	80	100	250	500
Mushroom	NB	EITL	**0.172**	**0.177**	0.177	0.177	0.177	0.177	0.164	0.164
		Basic	0.203	0.182	0.157	0.14	0.12	0.124	0.119	0.11
	J48	EITL	0.209	0.13	0.047	0.027	**0.027**	**0.027**	0.023	0.001
		Basic	0.209	0.13	0.034	0.027	0.032	0.029	0.023	0.001
Magic	NB	EITL	**0.122**	**0.087**	0.103	**0.093**	**0.091**	**0.097**	**0.115**	0.136
		Basic	0.152	0.088	0.103	0.113	0.113	0.122	0.121	0.13
	J48	EITL	0.3	**0.191**	0.117	**0.141**	0.115	**0.114**	**0.088**	**0.076**
		Basic	0.3	0.194	0.117	0.144	0.115	0.115	0.093	0.079
Splice	NB	EITL	**0.149**	**0.188**	**0.199**	**0.163**	**0.162**	**0.154**	**0.102**	0.085
		Basic	0.388	0.368	0.3	0.213	0.185	0.168	0.108	0.084
	J48	EITL	**0.505**	**0.451**	0.321	0.231	**0.302**	**0.224**	**0.134**	**0.088**
		Basic	0.508	0.455	0.312	0.223	0.314	0.255	0.255	0.097
Hypo	NB	EITL	0.093	0.102	0.124	0.122	0.127	0.123	0.114	0.093
		Basic	0.091	0.091	0.096	0.097	0.099	0.097	0.098	0.085
	J48	EITL	0.14	0.14	0.129	0.095	**0.088**	**0.111**	**0.111**	**0.092**
		Basic	0.14	0.14	0.127	0.091	0.1	0.115	0.112	0.096

(b)

Data set	Learner	Method	The number of labeled data in D_{T_L}							
			5	10	20	30	40	50	125	250
Waveform	NB	EITL	**0.256**	**0.156**	0.202	0.202	0.202	0.192	0.176	0.166
		Basic	0.43	0.376	0.11	0.126	0.136	0.11	0.118	0.126
	J48	EITL	0.508	0.256	0.306	0.202	0.202	0.23	**0.162**	**0.196**
		Basic	0.508	0.256	0.306	0.202	0.202	0.23	0.222	0.204
Segment	NB	EITL	**0.282**	**0.456**	**0.3**	**0.302**	**0.232**	**0.2**	**0.214**	0.162
		Basic	0.38	0.588	0.406	0.324	0.282	0.236	0.218	0.16
	J48	EITL	**0.486**	**0.54**	**0.272**	**0.328**	**0.224**	**0.25**	**0.098**	0.084
		Basic	0.864	0.85	0.85	0.82	0.302	0.252	0.134	0.08

To clearly demonstrate the performance, Fig. 2 and Fig. 3 show the learning curves of EITL compared with the basic method on each data set, when different numbers of labeled data in the target domain are used.

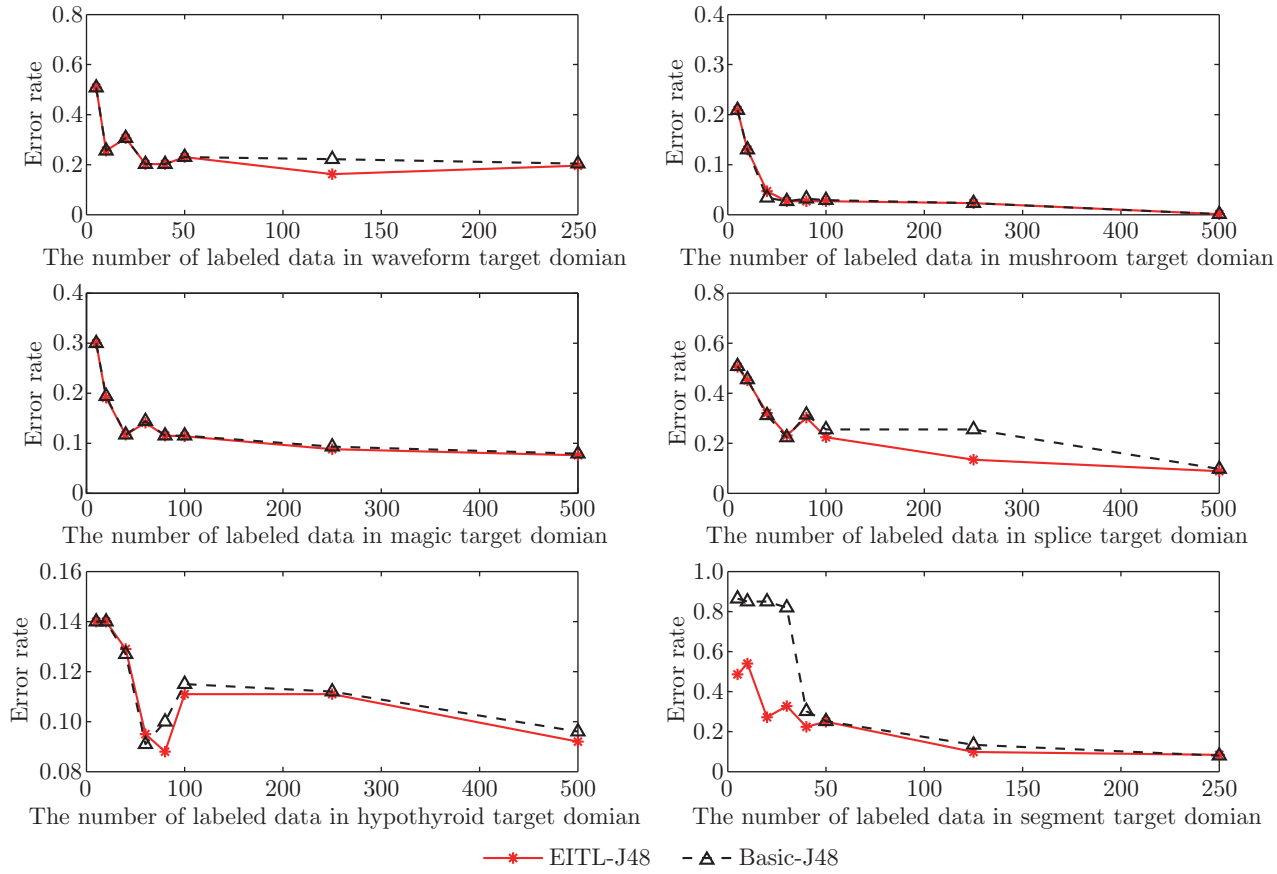

Fig. 3: The classification error rates of J48 based EITL vs. J48 based basic method on test data, using different numbers of labeled data in target domain of the six UCI data sets

Fig. 2 shows the error rates of the methods base on NB. When the number of labeled data in the target domain reaches 250 or 125, the basic method has sufficient data to train a good classifier. Therefore, we just compare EITL with basic method when the number of labeled data increases from 10 to 100 (on *mushroom*, *magic*, *splice*, and *hypothyroid* data sets) or from 5 to 50 (on *waveform* and *segment* data sets). It reveals that:

(1) From the performance on *waveform* and *mushroom* data sets, the EITL method is effective when only a few labeled data exist in the target domain. Compared with the basic method, the error rates of EITL are generally lower when the size of D_{T_L} is smaller than or close to 20. When the size of D_{T_L} is over 20, the basic method has enough data to train a good classifier; if we still use the source domain to take part in classifying, the accuracy will be influenced.

(2) From the performance on *magic*, *splice*, and *segment* data sets, the EITL method always outperforms the basic method. It means that, EITL effectively uses the source domain to help label the unlabeled data in the target domain, no matter how many original labeled data exist in the target domain.

The error rates of the two methods based on J48 are depicted in Fig. 3. It could be obtained that, the EITL method has similar performance as the basic method; however, as the size of D_{T_L}

increases, the error rates of EITL are lower than those of the basic method.

5 Conclusion

In this paper, we propose a novel ensemble method to solve the instance based inductive transfer learning problem. At each iteration, we adopt the ensemble idea that combines the classifier learned from the source domain data, the base classifier built on the initial labeled data in the target domain, and the classifier built on the updated labeled data in the target domain to label unlabeled instances, and then build a new classifier based on the expanded labeled data. The classifier learned in each iteration is added to the ensemble. Then, after many ensemble classifiers are built iteratively, we use the majority vote strategy to predict the labels of test data. Based on the experimental results on synthetic data sets and six UCI data sets, it is concluded that, the new algorithm EITL is effective on inductive transfer learning in terms of accuracy.

Acknowledgement

We thank anonymous reviewers for their valuable comments and suggestions.

References

[1] P. N. Bennett, S. T. Dumais, E. Horvitz, The combination of text classifiers using reliability indicators, Information Retrieval, 8(1), 2005, 67-100

[2] E. Bonilla, K. M. Chai, C. Williams, Multi-task Gaussian process prediction, Proceeding of the 20th Annual Conference on Neural Information Processing Systems, 2008, 153-160

[3] W. Dai, Q. Yang, G. Xue et al., Boosting for transfer learning, Proceedings of The 24th Annual International Conference on Machine Learning (ICML'07), 2007, 193-200

[4] T. G. Dietterich, Ensemble Learning, The Handbook of Brain Theory and Neural Networks, Second Edition, (M. A. Arbib, Ed.), 2002, 405-408

[5] J. Gao, W. Fan, J. Jiang et al., Knowledge transfer via multiple model local structure mapping, Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008), 2008, 283-291

[6] J. Jiang, C. Zhai, Instance weighting for domain adaptation in NLP, ACL'07, 2007, 264-271

[7] T. Kamishima, M. Hamasaki, S. Akaho, TrBagg: A simple transfer learning method and its application to personalization in collaborative filtering, Proceedings of the 2009 Ninth IEEE International Conference on Data Mining, 2009, 219-228

[8] L. I. Kuncheva, Combining Patten Classifiers: Methods and Algorithms, John Wiley & Sons, Inc., Hoboken, New Jersey, 2004

[9] X. Liao, Y. Xue, L. Carin, Logistic regression with an auxiliary data source, Proceedings of the 22nd International Conference on Machine learning, 2005, 505-512

[10] Z. Marx, M. T. Rosenstein, L. P. Kaelbling et al., Transfer learning with an ensemble of background tasks, NIPS Workshop on Transfer Learning, 2005

[11] L. Mihalkova, T. Huynh, R. J. Mooney, Mapping and revising Markov logic networks for transfer learning, Proceeding 22nd Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence, 2007, 608-614

[12] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering, 22(10), 2010, 1345-1359

[13] R. Raina, A. Battle, H. Lee et al., Self-taught learning: Transfer learning from unlabeled data, Proceedings of 24th International Conference on Machine Learning, 2007, 759-766

[14] W. Shi, W. Fan, J. Ren, Actively transfer domain knowledge, Machine Learning and Knowledge Discovery in Databases, 2008, 342-357

[15] Y. Shi, Z. Lan, W. Liu et al., Extending semi-supervised learning methods for inductive transfer learning, Proceedings of IEEE International Conference on Data Mining (ICDM '09), 2009, 483-492

[16] A. Storkey, M. Sugiyama, Mixture regression for covariate shift, Advances in Neural Information Processing Systems, 19, 2007, 1337-1344

[17] I. H. Witten, E. Frank, M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques (Third Edition), Morgan Kaufmann, 2010

[18] Z. Zhou, M. Li, Tri-training: Exploiting unlabeled data using three classifiers, IEEE Transactions on Knowledge and Data Engineering, 17(11), 2005, 1529-1541