

Transfer learning with reasonable boosting strategy

Lei La · Qiao Guo · Qimin Cao · Yongliang Wang

Received: 13 June 2012 / Accepted: 7 December 2012 / Published online: 21 December 2012
© Springer-Verlag London 2012

Abstract The lack of labeled data is a serious problem which greatly hinders the application of text classification in new domains. In this era of information explosion, dependence of labeled data in traditional classification methods becomes ineffective in emerged new domains. The ideology of transfer learning makes it possible to use labeled identical distribution data of old domains for data mining in new domains. However, previous algorithms and practical application systems did not reach the perfect state. This paper presents a novel complete method for text categorization (TC) in new domains where the labeled data are insufficient. We first present an improved weighting strategy of boosting algorithms family to ensure training data can be used more efficiently. We then introduce boosting ideology with the novel weighting strategy into transfer learning, and a novel text classification algorithm is proposed which has the ability to use labeled data of old domains for new domain classification with a high performance. After the mathematical discussion of the proposed algorithm, we finally deploy a real-world system based on it to evaluate the novel method. Experimental results demonstrate that our method is able to achieve both ideal accuracy and efficiency in TC when dealing with cross-domain problems.

Keywords Text categorization · Transfer learning · Boosting · Lack of labeled data · Weighting strategy

1 Introduction

Text categorization (TC) is a general term that describes the process of deciding which category a text should belong to according to its features or topics, for instance, whether a news report belongs to politics, sports, economics, or others; whether an e-mail is a spam; whether a user comment is complaint, etc. Similar to other data classification tasks, the main purpose of TC is to build a classifier F which could make judgment as function (1) when inputting a text T and category C :

$$F(T, C) = \begin{cases} 1, & T \text{ belongs to category } C \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Classifier design, in another word classification algorithm design, is the key issue in TC. Through the efforts of scholars, many TC algorithms have been proposed and improved in last decade. The most precious algorithms include [1] support vector machine (SVM), k -nearest neighbors (k NN), and boosting [2] methodology-based algorithm such as AdaBoost [3]. The most runtime-efficient algorithms are [4] Naïve Bayes and C 4.5. Improved accuracy and efficiency at the same time is a contradiction which is difficult to be solved. Several years ago, keeping balance between precision and time consumption is the most crucial problem in TC and AdaBoost algorithm family achieve the most ideal results [5]. However, with the explosive growth of Web technology such as microblog and SNS, extremely huge amount of new domain text data needs to be analyzed and processed. This phenomenon has brought a new challenge for text classification: lack of labeled training samples in new domains. Manual labeling is obviously not feasible because a large number of new domains lead to an intolerable labor cost. Unfortunately, the data distribution of old domains and new domains is

L. La (✉) · Q. Guo · Q. Cao · Y. Wang
School of Automation,
Beijing Institute of Technology, Beijing, China
e-mail: lalei1984@yahoo.com.cn

usually different; the basic assumption that the data distribution of training and test sets should be identical makes it impossible to use labeled data of old domains directly.

Transfer learning is a basic human cognitive phenomenon which can give great inspiration to machine learning. Generally speaking, transfer learning is the process of cognitive new things according to the knowledge of known things. Figure 1 is an example of transfer learning.

Figure 1 describes a child who never been to tropical can classify mango into *fruit* for the first time; he see it by his knowledge of temperate fruits such as apples, peaches, and watermelons. This phenomenon reveals the possibility of machine classification in unknown domains based on the knowledge of old domains.

This research paper demonstrates a novel boosting strategy based on transfer learning algorithm. The novel method can use labeled data of known domains to make TC in new domains with a high precision and low cost. The remaining paper is organized as follows. Section 2 reviews related works in this field. A novel weighting strategy for training sets is proposed in Sect. 3. Strategy for weak classifiers is proposed in Sect. 4 to construct a transfer learning-based classification method. After the mathematical discussion about the novel method in Sect. 5, a real-world system is deployed in Sect. 6, and the performance

of the method is evaluated. Finally, Sect. 7 summarizes the paper.

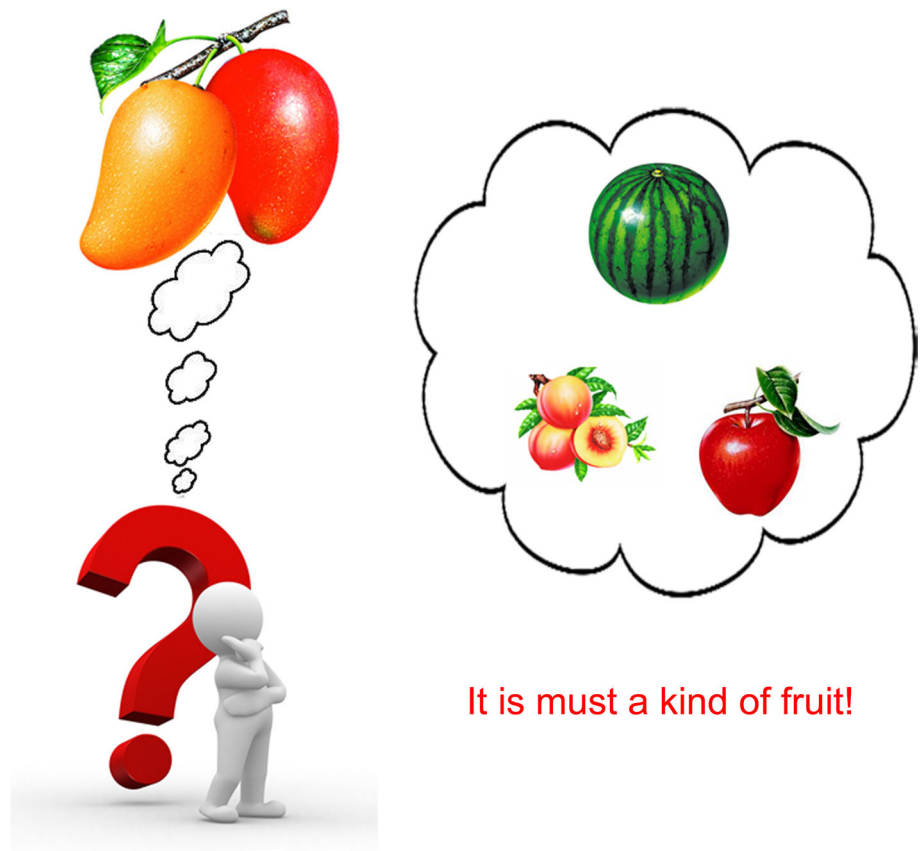
2 Related work

As the number of new domains grows faster and faster in recent years, traditional TC tools lost part of their power. In this case, more and more researchers pay attention to transfer learning-based classification. Therefore, a lot of transfer learning-based methods had been proposed in the previous literatures.

The work in [6] presented a complete framework of transfer learning. In this paper, procedure of transfer learning was divided into eight levels: (1) parameterization, (2) extrapolating, (3) restructuring, (4) extending, (5) restyling, (6) composing, (7) abstracting, and (8) generalizing. Algorithms following the above levels will have a solid theory foundation and expected to achieve high accuracy. However, that framework is sometimes too complex to be implemented in real-world system.

Fuzzy refinement-based transductive transfer learning (FRTTL) [7] is quite a light transfer learning algorithm which has low runtime complexity, but the experimental results reveal the algorithm is parameter sensitive. The

Fig. 1 An example of transfer learning



parameter-sensitive attribution will affect system's robustness when using it for text classification. Multi-step bridged refinement (MSBR) [8] is another widely used transductive transfer learning model. The parameter-sensitive problem in FRTTL does not exist in MSBR. However, the efficiency of MSBR is really not ideal.

In the last 2–3 years, computational intelligence algorithms-based transfer learning attracted some attention from scientists and engineers. Genetic transfer learning [9] probably is the most important algorithm in this field. Reference [9] uses genetic algorithm to find the solutions of transfer learning and greatly improve the efficiency of classification. It is regretful that the main problem of GA-based methods that they are hard to get global optimized solution is not solved in this algorithm so the precision of classification is limited.

Sparse transfer learning (SPA-TL) [10] algorithm can use labeled figures in known cartoons to achieve picture retrieval in new cartoons. The authors implement the algorithm in a real system to evaluate its performance. Experiments demonstrated it has good performance both in accuracy and in efficiency. However, the authors did not give a general framework of the algorithm which can be used not only in cartoon classification but also in other classification tasks.

Dai et al. [11] designed a transfer learning-based algorithm called TrAdaBoost. TrAdaBoost can allocate weights adaptively to labeled training samples of known domains according to the *distance* between labeled old domain samples and new domain data by using weighting strategy similar to AdaBoost [3]. This method is very useful to construct a general algorithm for classification in many fields such as speech recognition, TC, image classification, optical characteristic recognition, and so on. However, several problems exist in TrAdaBoost which will limit precision and efficiency of classification as below:

1. The labeled data from old domain which have different distribution from subject data are allocated initial weights $w \in (0,1]$; when it be classified incorrectly, the weights will be reduced. This strategy will increase time consumption of the algorithm.
2. Only the same distributed training set T_s is used to judge the *distance* between auxiliary set and subject data set. This strategy will lead to a low distinguishing ability.
3. Weight allocation is based on correct times but not the correct proportion. However, correct proportion is obviously the more suitable indicator.
4. Only weighting mechanism of training data is discussed in [11]. However, weighting mechanism of weak learners is also crucial for classification performance and should not be ignored.

To solve the problems mentioned above, we proposed a novel boosting-based weighting strategy and use it to construct a classification algorithm which can achieve transfer learning. The novel algorithm is called TransBoost in this article. It is a general method which can be used in many classification tasks, and we implement it in a text classification system to evaluate its performance.

3 Selecting the most useful additional training data

In the field of TC, new domains and new categories emerged with the development of Web technology such as micro-blog (twitter), SNS (facebook, LinkedIn), and users' comment Web site (yelp). On the one hand, labeled data are lacked in these new domains. On the other, labeled data of traditional domains such as news reports and literatures are widespread in a variety of corpus. Because the data distribution of old domains is usually different from new domains, it is nearly impossible to use labeled data of standard corpus directly for classifying text of new domains. If the labeled data of old domains can be used to categorize texts of new domains in some way, it will bring revolutionary improvement to data mining in new domains. The key issue which makes labeled data of old domains useful for new domains is selecting the labeled data of old domains which are most similar to the new domains and giving them greater weights.

3.1 Weighting mechanism of TraAdaBoost

A small labeled datum of new domain is needed as training set in TraAdaBoost. This set is used as base training document set D_b which contains n documents. The additional training set whose data belonged to old domain is called D_a , and the number of text in D_a is m . $C = \{0, 1\}$ is the set of category labels. Initial weights of training samples are as follows:

$$\omega_i = \begin{cases} 1/n, & 1 \leq i \leq n \\ 1/m, & n + 1 \leq i \leq n + m \end{cases} \quad (2)$$

In TraAdaBoost, base learner will be trained by the document set $D = D_a + D_b$. Then, D_a will be used to check the similarity of data distribution of the base training set and the additional training set. Different from traditional AdaBoost, in TrAdaBoost whenever a text in D_a be misclassified, its weight will be reduced. It is because in TrAdaBoost, data distribution of additional training set and test set is not identical. In this situation the authors regarded the misclassification due to huge distribution difference instead of regarding the misclassified texts as difficult samples. The schematic diagram of additional sample weighting in TrAdaBoost is shown in Fig. 2.

At last D_a and D_b with their weights will be used as the final training set. Using the above weighting strategy TrAdaBoost achieved transfer learning with low cost. However, as the analysis in Sect. 2, several drawbacks exist in its weighting strategy of training documents.

3.2 A more reasonable way for weight allocation

To solve problems of TrAdaBoost, we proposed a novel weighting strategy to make weight allocation for additional training data in a more reasonable way.

First of all, when making distribution similarity evaluation, the strategy of TrAdaBoost in which only D_a is used as the distribution test set leads to a low ability to distinguish because in this way, it is hard to say whether the misclassification caused by *different* or *difficult*. Therefore, we use a comprehensive strategy to measure the distance between same distributed base training set D_b and additional training set D_a , called two-stage distribution evaluation strategy (TSDE).

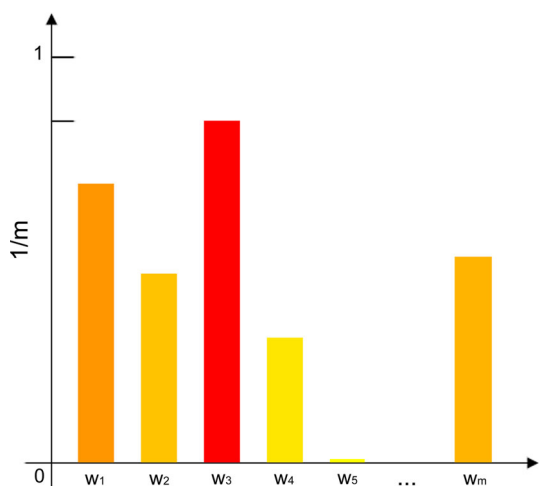


Fig. 2 Additional training data weighting of TrAdaBoost

In TSDE, two stages of distribution test are needed. In the first stage, D_a is used as training set, and $D_a \cup D_b$ is used to evaluate the similarity of data distribution. In the second stage, D_b is used as training set, and $D_a \cup D_b$ is used for evaluation. After the two stages mentioned above, there are three kinds of possible situations: (1) evaluated document be classified correctly in both stages, (2) evaluated document be classified incorrectly in both stages, and (3) the evaluation results of a sample are different in the two stages. The detailed possible situations are shown in Table 1.

When a training document $D_i \in D_a$ is considered as an easy sample or a difficult sample, the system will keep its weight rather than changing it to prevent wrong decisions. When a training document $D_i \in D_b$ is considered as an easy sample or a difficult sample, the weight adjustment strategy which is similar to traditional AdaBoost will be used to improve accuracy. When $D_i \in D_a$ is considered as a significantly different distribution to test set, its weight will be reduced. Distractors will be discarded as noise.

The steps mentioned above will be operated T times until the algorithm is convergence. In this way, the system can distinguish between different-distributed data and difficult data and give training data weights in a more reasonable way.

Secondly, in TrAdaBoost, an initial weight is allocated to data of additional training set, and all data from D_a will be involved in classification. This strategy is suitable for traditional AdaBoost when training set and test set have the same data distribution. In different distribution case, some noisy data should be excluded. To achieve this purpose, a threshold σ of weight is introduced into the system as:

$$\sigma = \frac{\sum_{i=1}^m \omega_i + m \cdot \min_{1 \leq i \leq m} \omega_i}{2m} \tag{3}$$

When weight of an additional training data is lower than σ , it will be moved out of D_a . Moreover, we modified the iterative weighting function of additional training data to

Table 1 Situations of the results of TSDE

First stage			Second stage			Conclusion
Test sample D_i	Training set	Result	Test sample D_i	Training set	Result	
$D_i \in D_a$	D_a	Right	$D_i \in D_a$	D_b	Right	Easy sample
$D_i \in D_b$	D_a	Right	$D_i \in D_b$	D_b	Right	
$D_i \in D_a$	D_a	Wrong	$D_i \in D_a$	D_b	Wrong	Difficult sample
$D_i \in D_b$	D_a	Wrong	$D_i \in D_b$	D_b	Wrong	
$D_i \in D_a$	D_a	Right	$D_i \in D_a$	D_b	Wrong	Different distribution
$D_i \in D_b$	D_a	Wrong	$D_i \in D_b$	D_b	Right	
$D_i \in D_a$	D_a	Wrong	$D_i \in D_a$	D_b	Right	Distractors
$D_i \in D_b$	D_a	Right	$D_i \in D_b$	D_b	Wrong	

reduce system’s time consumption. The definition of training data’s initial weight is function (2). The weight will be updated using the following function:

$$\omega_i^{t+1} = \begin{cases} \omega_i^t - (-1)^{|C'(D_i)-C(D_i)|} \cdot \lambda_1, & 1 \leq i \leq n \\ \omega_i^t + (-1)^{|C'(D_i)-C(D_i)|} \cdot \lambda_2, & n + 1 \leq i \leq n + m \end{cases} \quad (4)$$

where $C'(D_i)$ is the category of D_i given by classifier, $C(D_i)$ is the real category of D_i , and t is the current round of iteration. The weight change λ_1 and λ_2 are defined as:

$$\lambda_1 = 1 / \left(1 + \sqrt{2 \ln n / T} \right) \quad (5)$$

$$\lambda_2 = 1 / \left(1 + \sqrt{2 \ln m / T} \right) \quad (6)$$

where T is the number of iterative rounds. It is easy to known, at the t th round, the classification errors in D_a and D_b are as follows:

$$e_a^t = \sum_{i=1}^m \frac{\omega_i^t |C'(D_i) - C(D_i)|}{\sum_{i=1}^n \omega_i^t} \quad (7)$$

$$e_b^t = \sum_{i=1}^n \frac{\omega_i^t |C'(D_i) - C(D_i)|}{\sum_{i=1}^n \omega_i^t} \quad (8)$$

Obviously, the classification errors are bounded. In this way, the base classifier can output final classification results of test documents in a reasonable weighted way. The detailed workflow of training data weighting is shown by its pseudocode in Fig. 3.

Note that in the aforementioned weighting strategy the weights of additional data are adjusted according to the proportion of the data be considered as different-distributed data in the T rounds. Therefore, the three problems of additional training data weight allocation in TrAdaBoost are theoretically solved.

4 Weighting the base learners

Another important problem of TrAdaBoost is that the weighting mechanism of base learners had not been discussed. However, the way of using weak learners is a crucial point in boosting-based algorithms. Only when the base classifiers get weights according to their classification ability, the system can achieve the best performance.

4.1 Weight allocation strategy for weak classifiers

Benefiting from the distribution similarity evaluation step, no additional stages are needed to evaluate the classification ability of weak classifiers. The results of similarity evaluation can be used directly for assessing base learners’

Algorithm: Weights allocation strategy of training documents

Input: base training set $D_b = \{D_b(1), D_b(2), \dots, D_b(n)\}$ Additional training set $D_a = \{D_a(1), D_a(2), \dots, D_a(n)\}$ iteration times T

Output: Weight ω_i ($1 \leq i \leq n+m$) of training data

```

1 begin
2   for(t = 1, t <= T, t++)
3     train learner using  $D_b$ 
4     similarity evaluation using  $D_a \cup D_b$ 
5     record result  $C_1(D(i))$  where  $1 \leq i \leq n+m$ 
6     training learner using  $D_a$ 
7     similarity evaluation using  $D_a \cup D_b$ 
8     record result  $C_2(D(i))$  where  $1 \leq i \leq n+m$ 
9     for(i = 1, i <= n, i++)
10      if  $C_1(D(i)) = C_2(D(i)) = C(D(i))$ 
11         $\omega_i^{t+1} = \omega_i^t - \Delta_1 \omega$ 
12      else if  $C_1(D(i)) \neq C(D(i)) \&\& C_2(D(i)) \neq C(D(i))$ 
13         $\omega_i^{t+1} = \omega_i^t + \Delta_1 \omega$ 
14      else
15         $\omega_i^{t+1} = \omega_i^t$ 
16      for(j = n+1, j <= n+m, j++)
17        if  $C_1(D(i)) \neq C_2(D(i))$ 
18           $\omega_j^{t+1} = \omega_j^t - \Delta_2 \omega$ 
19        else
20           $\omega_j^{t+1} = \omega_j^t$ 
21      output weights of training document
24 end
    
```

Fig. 3 Pseudocode of training data weighting

performance. Similar to Table 1, the performance of weak classifiers is summarized in Table 2.

As shown in Table 2 once similarity evaluation of each iteration is done, the weak classifiers can be considered as strong classification ability, strong classification ability and insensitive to the distribution of training data, and weak classification ability and sensitive to the distribution of training data.

Obviously, classifier which has strong classification ability and insensitive to the distribution of training data should have greater weight. Therefore, we adjust weight of the i th weak learner as:

$$w_i^{t+1} = w_i^t + \sqrt{2 \ln N / T} \quad (9)$$

where t is the current iterative round and N is the number of weak classifiers. When the base learner is considered as strong classification ability, its weight should not be enhanced rashly because it is unknown whether the classifier is training data distribution insensitive. In this situation, we should keep the classifier’s weight as:

Table 2 Situations of weak classifiers’ performance

Training set	The set which evaluation data belonged to	Result	Conclusion
D_a	Additional training set D_a	Right	Strong classification ability
D_b	Base training set D_b		
D_a	Base training set D_b	Right	Strong classification ability and insensitive
D_b	Additional training set D_a		
D_a	Additional training set D_a	Wrong	Weak classification ability
D_b	Base training set D_b		
D_a	Base training set D_b	Wrong	Training data distribution sensitive
D_b	Additional training set D_a		

$$w_i^{t+1} = w_i^t \tag{10}$$

The situation may lead confusion is the base learner be considered as sensitive to the distribution of training data because its classification ability is unknown. Since the majority of training set is different-distributed old domain samples, whether sensitive to the training data is more important than the classification ability [12]. Therefore, the weight of base classifier in this situation should be reduced as:

$$w_i^{t+1} = \theta \cdot w_i^t \tag{11}$$

where θ is an empirical parameter which satisfied $\theta \in (0.5, 1)$. In this way, the contribution of distribution-sensitive classifiers could be limited. The last situation in which the weak learner has both low classification ability and high distribution sensitive is the worst situation. This kind of classifiers should be restricted by:

$$w_i^{t+1} = w_i^t - \sqrt{2 \ln N/T} \tag{12}$$

When the weight of classifier i is lower than zero, it will be excluded from the *expert committee*.

4.2 Full form of Trans-Boost

Hitherto, the strategy which could achieve weight allocation for base classifiers is proposed completely with no condition missing. Because the weighting procedure for weak classifiers is simultaneous with the distribution similarity evaluation process, the weight allocation for training data and base learners can be achieved at the same time with a cross-iterative method.

The method which combined the analyses mentioned in Sects. 3 and 4 is called Trans-Boost algorithm in this paper. The flowchart of Trans-Boost is shown in Fig. 4.

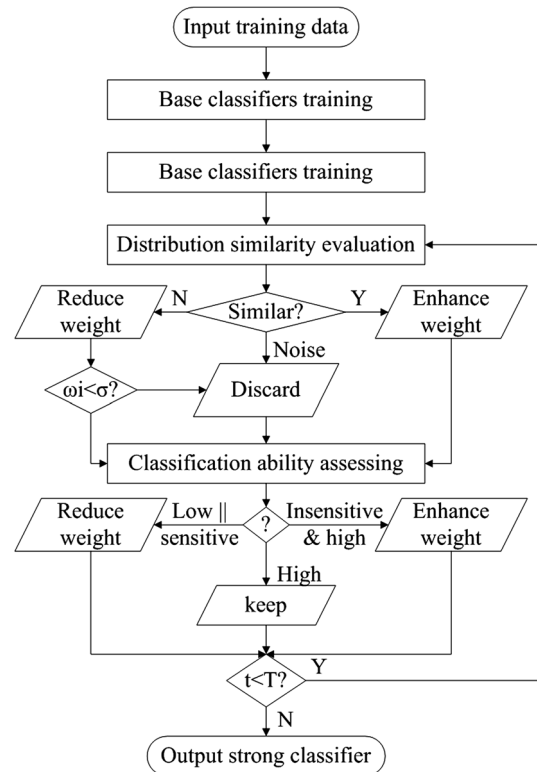


Fig. 4 Flowchart of Trans-Boost

Following the steps shown in Fig. 4, the system can achieve transfer learning by using labeled data of old domains as training documents to make classification in new domains. All problems of TrAdaBoost are solved by the novel boosting-based TC algorithm. Theoretically, the novel algorithm has high accuracy and low time consumption. Furthermore, it is actually a general method which can deal with a wide range of classification problems.

5 Mathematical discussion

The transfer learning framework for classification is fully proposed in previous sections. Some mathematical features of the novel method should be discussed such as runtime complexity and convergence rate.

According to function (4), for each training sample the computation is one, so runtime complexity of the first stage of distribution similarity evaluation is $O(n + m)$. Therefore, computational complexity of TSDE-based training data weighting is $O(2n + 2m)$. As the analysis in Sect. 4, no additional calculation was carried out by the weight allocation process for base classifiers. Taking the number of iterative rounds into account, runtime complexity $H(o)$ of the novel transfer learning classification framework is the following:

$$H(o) = O\{2T(n + m)\} \tag{13}$$

The runtime complexity of Trans-Boost would grow linearly with the increase in training set and iterative rounds. No exponential explosion problem exists in the novel method, and its time consumption is quite low.

Whether the algorithm is convergence or not is a crucial attribution. It will be useless when the algorithm is not convergence. In addition, low convergence rate will injure performance of the algorithm.

The classification error of Trans-Boost is described in functions (7) and (8). Define the training weight of D_i as:

$$W^t = \frac{w^t}{\sum_{i=1}^{n+m} \omega_i^t} \tag{14}$$

Obviously, the following equation is satisfied:

$$\lim_{T \rightarrow \infty} \frac{\sum_{t=\lceil T/2 \rceil}^T \sum_{i=1}^n W_i^t \cdot \varepsilon^t}{T - \lceil T/2 \rceil} = 0 \tag{15}$$

Equation (15) demonstrates that when the times of iterations increase, classification error tends to zero. In other words, the novel algorithm is convergence. Similar to the derivation process in [11], the convergence rate R_C of Trans-Boost is:

$$R_C = \frac{\sqrt{2 \ln n/m}}{T} \tag{16}$$

When the number of texts in additional training set is more than two, convergence rate of the novel algorithm will be faster than the inverse of the number of iterations. It is significantly faster than TrAdaBoost.

In addition, when the system makes iterative weighting, weights of samples in additional training set whose distribution is suitable for test data will gradually increase. The impact of significantly different data will be reduced or even excluded. The extreme situation is that all samples in training set are not similar with test data and be ignored. In this case, Trans-Boost becomes an improved version of traditional AdaBoost whose weighting strategy is more advanced than traditional AdaBoost. Therefore, AdaBoost can be considered as a special case of Trans-Boost, and Trans-Boost is the general form of AdaBoost [13].

6 Simulation, experiment, and analysis

The accuracy and efficiency of Trans-Boost have been proved in theory. However, its performance should be evaluated in simulation and real-world system. Comparing its performance with other classic classification algorithms will lead to an objective conclusion of Trans-Boost.

6.1 Efficiency evaluation

We simulated the time consumption of Trans-Boost in Matlab with different size of training set. In order to control the number of variables, in each size, we keep $T = 50$ to fix the times of iterations. We compared time overhead of the novel algorithm with original AdaBoost [14], TrAdaBoost [15], Naïve Bayes [16], and SVM [17]. The comparative results are shown in Fig. 5.

In the aforementioned figure, 1,000, 3,000, 10,000, 30,000, and 100,000 documents were used. Logarithmic coordinates are used in X axis in order to facilitate the display. Therefore, the functions which appear to be exponential distributed functions are actually linear.

It is clear in Fig. 5 that Trans-Boost has the middle time efficiency among the algorithms. Its time consumption is lower than TrAdaBoost and SVM. Note that although its time overhead is higher than original AdaBoost and Naïve Bayes, Trans-Boost is still an ideal algorithm because firstly Naïve Bayes is a simple and low-accuracy algorithm and secondly original AdaBoost could not achieve transfer learning.

Convergence rate is another important aspect of efficiency. To check convergence rate of the novel algorithm, we make experiment to observe the change of error rate when the number of iterations increased. Standard training materials downloaded from Reuters-21578 [18] are used as training set. In order to evaluate the effect of transfer learning, different-distributed micro-blog corpus which is called *nlpir micro-blog corpus* [19] is used as text data. We compared its convergence rate with TrAdaBoost, the result as shown in Fig. 6.

It is clear that 1 %, 2 %, etc. means the proportion of training data in all data (training data and test data). The iterations are operated from 1 round to 100 rounds. The

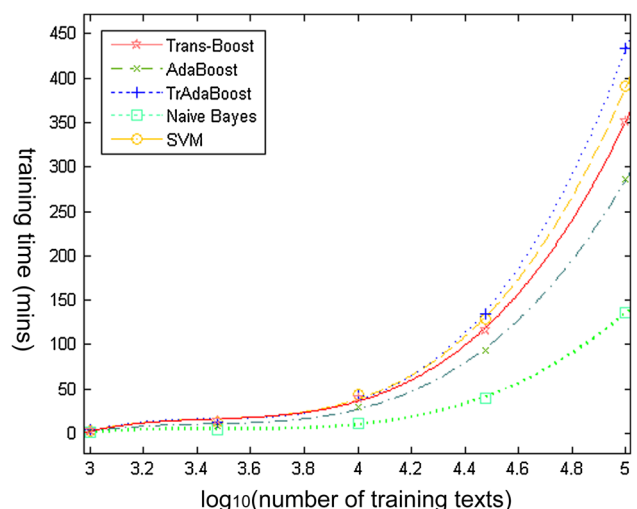


Fig. 5 Time consumption comparison

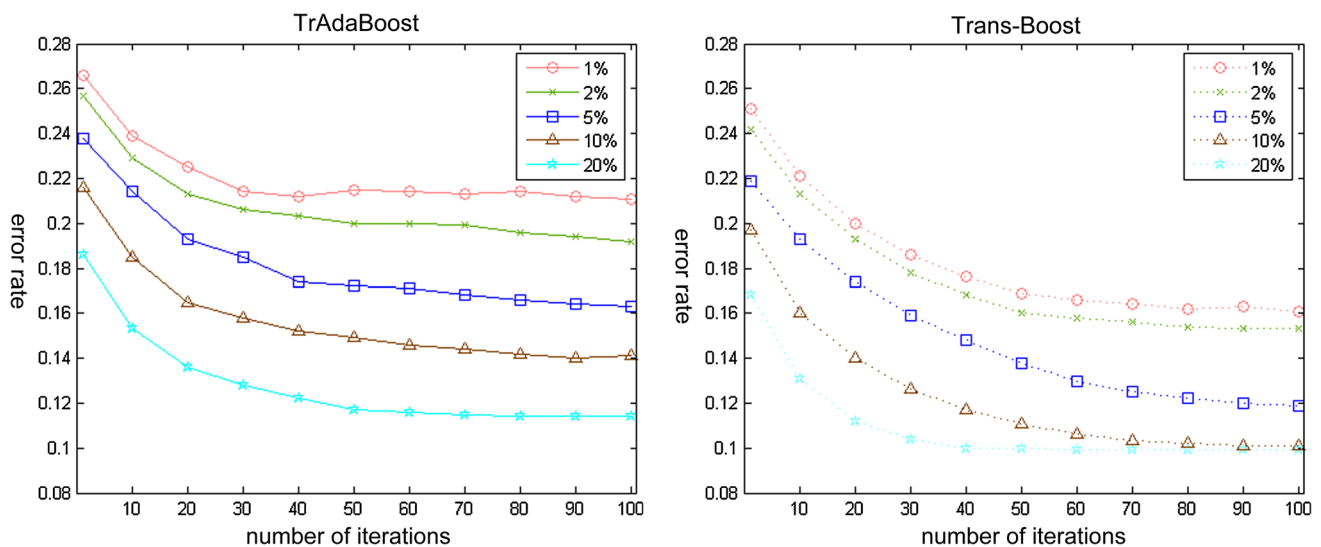


Fig. 6 Convergence rate comparison

Table 3 Precision comparison

Algorithms	Categories					
	Politics	Economics	Sports	Entertainments	Educations	Cultures
TrAdaBoost	0.851	0.843	0.845	0.850	0.844	0.848
MSBR	0.812	0.799	0.793	0.806	0.800	0.810
Naïve Bayes	0.623	0.621	0.623	0.617	0.614	0.612
DTMKL	0.889	0.891	0.888	0.893	0.892	0.885
k NN	0.715	0.705	0.717	0.708	0.716	0.710
Trans-Boost	0.895	0.886	0.889	0.881	0.894	0.882

aforementioned figure reveals Trans-Boost has higher convergence rate than TrAdaBoost. Furthermore, convergence curves of the novel algorithms are smoother than TraAdaBoost. In a word, Trans-Boost is a high efficiency classification algorithm.

6.2 Evaluation and analysis of the precision

Accuracy is the most important criteria to evaluate the performance of a classification algorithm. Scientific texts downloaded from *Penn Treebank* [20] corpus are used as an additional training set. Reuters-21578 is used as base training set and test set. The number of base training text is 500, and the number of additional training documents is 19,500. We test the performance of novel algorithm using test data belonged to politics, economics, sports, entertainments, educations, and cultures. Eighteen thousand documents are used in each category. Works of previous literatures [5, 6, 21–23] are used for comparison. The experimental results are shown in Table 3.

In the classification process of each category, 50 iterative rounds are operated. As shown in Table 3, DTMKL

and the novel algorithm presented in this article have the highest precision. Their accuracy is significantly higher than MSBR, Naïve Bayes, and k NN. In addition, they are more precious than TrAdaBoost. Taking the high computational overhead of DTMKL into account, Trans-Boost is the best algorithm among them. Furthermore, the aforementioned table reveals traditional algorithms without the ability of transfer learning have very low accuracy when facing different-distributed data.

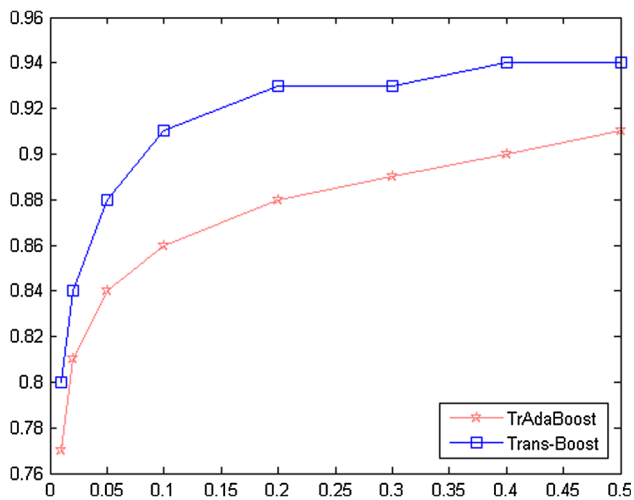
As the analysis in Sect. 5, when all training samples in additional training set are dissimilar to testing samples and be discarded, Trans-boost becomes an improved version of traditional AdaBoost. The accuracy of Trans-Boost in this situation is compared with traditional AdaBoost, which is shown in Table 4.

As shown in Table 4, since more advanced weighting mechanism is used in Trans-Boost, its precision is higher than traditional AdaBoost even no additional training data are used.

Finally, we tested the base training set dependence degree of the novel algorithm and compared its performance with TrAdaBoost. The experimental results are shown in Fig. 7.

Table 4 Precision comparison when no additional data are used

Algorithms	Categories					
	Politics	Economics	Sports	Entertainments	Educations	Cultures
Trans-Boost	0.906	0.929	0.914	0.898	0.903	0.904
Traditional AdaBoost	0.897	0.907	0.899	0.886	0.901	0.892

**Fig. 7** Base training set dependence evaluation

In Fig. 7, the size of training set is 10^5 documents, and the size of testing set is also 10^5 documents. As shown in Fig. 7, the novel algorithm can achieve 0.8 correct rates even when only 1 % identical distribution base training data are available. When the proportion of base training data is more than 10 %, the accuracy curve grows slowly.

7 Conclusion and future work

A boosting-based transfer learning classification algorithm is presented in this paper. It uses novel strategies for weight allocations. These strategies ensure to select training data in old domains which have higher distribution similarity with new domain data. Furthermore, base learners with stronger classification ability and insensitive to data distribution will get greater weights through this algorithm. Simulations and experimental results revealed the novel algorithm proposed in this article has low time consumption and high accuracy. Moreover, it is a robust algorithm.

Whether a transfer learning framework can give up the help of base training data and use totally old domain data for new domain classification with high performance is an interesting problem. This could be undertaken as future work on this topic.

References

- Feldman R, Sanger J (2007) The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge University Press, New York, pp 77–78
- Younes Z, Abdallah F, Deneux T (2011) A dependent multilabel classification method derived from the k -nearest neighbor rule. *EUROSIP J Adv Sig Process* 2011:1–20
- Schapire RE, Freund Y, Bartlett P, Lee WS (1997) Boosting the margin: a new explanation for the effectiveness of voting methods. In: Proceedings of the fourteenth international conference on machine learning, 1997, Nashville, USA, pp 1–30
- Liu L, Qianhui L (2012) A hybrid algorithm for text classification problem. *Przeglad Elektrotechniczny* 88(1B):8–11
- Ahmed R, Alexis J, Nozha B (2012) B-Lasso for object categorization and retrieval: towards interpretable visual models. *Pattern Recognit* 45(6):2377–2389
- Cook DJ, Holder LB, Michael Youngblood G (2007) Graph-based analysis of human transfer learning using a game testbed. *IEEE Trans Knowl Data Eng* 19(11):1465–1478
- Vahid B, Jie L, Guangquan Z (2011) Long term bank failure prediction using fuzzy refinement-based transductive transfer learning. In: IEEE international conference on fuzzy systems, June 27–30, 2011, Taipei, Taiwan, pp 2676–2683
- Qin J-W, Zheng QL, Ma Q-L, Wei J, Lin G-L (2011) Multi-step bridged refinement for transfer learning. *J South China Univ Technol (Nat Sci Ed)* 39(5):108–114
- Koçer B, Arslan A (2010) Genetic transfer learning. *Expert Syst Appl* 37(9):6997–7002
- Jun Yu, Cheng J, Tao D (2012) Interactive cartoon reusing by transfer learning. *Signal Process* 92(9):2147–2158
- Dai W, Yang Q, Xue, Yu Y (2007) Boosting for transfer learning. In: Proceedings of the 24th international conference on machine learning, Corvallis, pp 193–200
- Zhuang F, Luo P, Shen Z, He Q, Xiong Y, Shi Z, Xiong H (2012) Mining distinction and commonality across multiple domains using generative model for text classification. *IEEE Trans Knowl Data Eng* 24(11):2025–2039
- Xue G, Dai W, Yang Q (2008) Topic-bridged PLSA for cross-domain text classification. In: The thirty-first international ACM SIGIR conference on research and development on information retrieval (SIGIR 2008), Singapore, pp 627–634
- Su Yu, Shiguang S, Xilin C, Wen C, Pursuit C-B (2011) Classifiability-based discriminatory projection pursuit. *IEEE Trans Neural Netw* 22(12):2050–2061
- Ling X, Dai W, Xue G (2008) Spectral domain-transfer learning. In: The fourteenth ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2008), Las Vegas, pp 488–496
- Soriaa D, Garibaldia J, Ambrogib F, Biganzolib E, Ellis I (2011) A ‘non-parametric’ version of the naive Bayes classifier. *Knowl Based Syst* 24(6):775–784
- Hui X, Songcan C (2011) Glocalization pursuit support vector machine. *Neural Comput Appl* 20(7):1043–1053

18. Lewis DD Reuters-21578 test collection. <http://www.daviddlewis.com>
19. Huaping Z <http://www.nlpir.org/?action-viewnews-itemid-232>
20. <http://www.cis.upenn.edu/~treebank>
21. Mizianty Marcin J, Kurgan Lukasz A, Ogiela Marek R (2010) Discretization as the enabling technique for the Naive Bayes and semi-Naive Bayes-based classification. *Knowl Eng Rev* 25(4):421–449
22. Duan L, Tsang I, Dong X (2012) Domain transfer multiple kernel learning. *IEEE Trans Pattern Anal Mach Intell* 34(5):465–479
23. Steele Brian M (2009) Exact bootstrap k nearest neighbor learners. *Mach Learn* 74(3):235–255