Chinese Journal of Electronics
Vol.28, No.1, Jan. 2019

An Online Education Data Classification Model Based on Tr_MAdaBoost Algorithm*

YU Lasheng, WU Xu and YANG Yu

(*School of Information Science & Engineering, Central South University, Changsha 410086, China*)

Abstract — With the rapid development of network information technology and the wide application of smart phones, tablet PCs and other mobile terminals, online education plays an increasingly important role in social life. This article focuses on mining useful data from the massive online education data, by using transfer learning, relying on Hadoop, to construct Online education data classification framework (OEDCF), and design an algorithm Tr_MAdaBoost. This algorithm overcomes the traditional classification algorithms in which the required data must be restricted to independent and identically distributed data, since online education using this new algorithm can achieve the correct classification even it has different data distribution. At the same time, with the help of Hadoop's parallel processing architecture, OEDCF can greatly enhance the efficiency of data processing, create favorable conditions for learning analysis, and promote personalized learning and other activities of big data era.

Key words — Transfer learning, Online education, OEDCF, AdaBoost algorithm, Tr_MAdaBoost algorithm.

I. Introduction

In recent years, as the country attaches great importance to education, and relying on cloud computing, big data and other information technologies, online learning platforms such as MOOCs, Baidu kk, Classroom of Netease have sprung up vigorously. In South Korea, the government announced the abolition of paper textbooks before 2015; In Japan, university course graduation requires 124 credits in which 60 credits that can be achieved through "distance learning" since 1998; In America, President Obama had expressed the hope that in the future within four years, 99 percent of American students would complete their education learning over the Internet. According to incomplete statistics in 2014, online education users of China maintain a high growth rate of over 10% since 2010. Compared with the corresponding period previous year, the scale of which reach to 7796.9 million, increased by 16.03%, and the market of which reached to 99.8 billion. It is an important issue to be solved in field of online education that how to do efficient data processing, realize standardized storage and usage with vast amounts of online data. These data may be the same person in different traces of learning platform, if which can be scientific and effective treatment, we can get more valuable information, such as learning interest, learning habits and learning progress and so on. But the traditional machine learning requires a lot of tagged data, which would take so much manpower and resources. Considering despite the different distribution of online and offline education but knowledge related, transfer learning can be a good solution to this problem. This article is trying to propose a transfer learning algorithm to help data mining on online education.

1. Transfer learning

NIPS 2005 gives a representative definition to transfer learning: learning to emphasize that transfer of knowledge between different but similar areas, task and distribution[1]. According to the definition, transfer learning can divided into: based on the example of transfer learning, based on the characteristics of the transfer learning, based on the parameters of transfer learning and based on the knowledge of transfer learning. Dai *et al.* proposed TrAdaBoost algorithm[2], which applied Boost idea into transfer learning and constructed improved classifier by strengthen weak classifier constantly, so as to improve classification performance. Wei *et al.* designed an algorithm called FSFP[3], using the potential semantic analysis to extract the keywords as the seed feature set to construct Laplace feature graph, by the help of which it can realize knowledge transfer from long text to short passage. Kuzborski *et al.* used source hypothesis to select the relevant sources from big data pool to transfer, which can solve the binary transfer learning problem well[4].

2. Classification and clustering of online data

Traditional data classification and clustering assumes

*Manuscript Received Feb. 22, 2017; Accepted Aug. 28, 2017.
 DOI:10.1049/cje.2018.06.006

that the data samples are independent and identically distributed. However, in the Internet, biological networks, social networks and other networks are interconnected between the data samples[5]. The online data classification first needs to solve problems such as big data access and data annotation, and then classifies the relationship between the attributes and nodes of the network. According to the characteristics of online data classification, domestic and foreign scholars have put forward many solutions. The collaborative classification method[6] synthetically using the various information in the network, is a kind of classification accuracy is higher and more widely used methods. Simple neighbor voting method is relying only on the relationship between the nodes, although the algorithm is simpler, but classification effect also good. In addition, the probability method, the method based on graph cut and the method based on information transmission and other methods are under researching. Shang *et al*. proposed GDBNSC-Ncut, GDBNSC-Rcut, DFSC, NSSRD[7−9] methods to do spectral clustering and feature selection clustering, all of which can effectively improve clustering quality. Though online data classif ication and clustering have been widely researched, but the research of online education data classification hasn't been paid more attention.

II. A Proposed Classification Framework

To solve the problems mentioned above, there is an Online education data classification framework called OEDCF proposed, just as shown in Fig.1. Transferring based on data classifier trained from additional school education by using transfer learning algorithm[10−15], and stored the massive online education data classified with the help of HBase, MapReduce, Sqoop which are very useful modules of Hadoop. For it is so easy to store the massive data to HBase by using Sqoop. This article only focuses on how to classify online learning data using transfer learning, not does deep analysis on big data processing.

III. Tr_MAdaBoost Algorithm

Due to the differences in teaching methods, learning environment, teaching resources, the off line data from school and online data from online education are the two areas, distribution of which is different but related, suit to do transfer learning. Since the partial data online education such as academic performance is similar to the school study, these data can be used as part of the training data to learn good classification model, together with a large amount of data available from school, which means based on the instance of transfer learning, can be taken. Using transfer learning to modify the traditional AdaBoost[16] algorithm to Tr_MAdaBoost algorithm, this will be fitter for different but related areas machine learning.

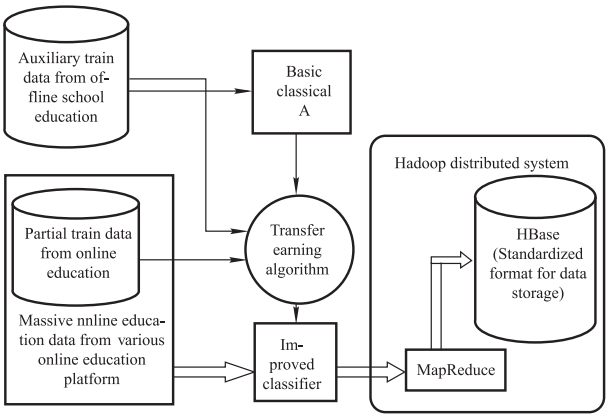

Fig. 1. Online education data classification framework

1. Algorithm description

AdaBoost algorithm is a set of integrated learning algorithm created based on PAC learning theory. The basic idea is simple using multiple weak classifiers to build a very high accuracy of strong classifier[17]. There are four kinds of classical AdaBoost algorithm: DiscreteAdaBoost[18], RealAdaBoost, GentleAdaBoost[18], ModestAdaBoost[19]. In this paper, a new algorithm Tr_MAdaBoost is designed based on ModestAdaBoost, by taking a different sample weights update method with training sample which have different distribution, in order to realize effective transfer learning. The main idea of this algorithm is shown in Algorithm 1.

From Algorithm 1, it's not hard to see that in each iteration if an auxiliary training data are misclassified, let its weight multiplied by $\beta_2{}^{-|h_t(x_i)-c(x_i)|}$, and if a training data from target domain is misclassified, then let its weight multiplied by $\beta_2{}^{-|h_t(x_i)-c(x_i)|}$. Since $\beta_2^{|h_t(x_i)-c(x_i)|} \in \{0,1\}$, $\beta_2^{-|h_t(x_i)-c(x_i)|} > 1$, reducing the weights of auxiliary training datamisclassified means reduce its impact on the classification model in next round of iteration, while increasing the weights of training data from target domain misclassified means draw more attention on the classifier in next round of iteration. Under such circumstances, after several rounds of iterations, the training data sample from auxiliary domain in line with those training data of the target domain will have higher weight, while those who do not meet those training data of the target domain will become increasingly lower weight. At the same time, the training data sample from target domain which is easy to be misclassified will have a higher weight in order to achieve steady improvement in classification accuracy. In extreme cases, it can ignore the impact of all the auxiliary training data, then Tr_MAdaBoost algorithm becomes the traditional AdaBoost algorithm.

Algorithm 1 Algorithm description of Tr_MAdaBoost

Input: Train data set include D_t (some tagged data from online education), D_s (a large number of marked data from offline school education), an unlabeled data sets S for test, the maximum number of iterations M, the weak classifiers using CART decision tree algorithm.

Output: proved strong classifier $B\{h_f : X \to \Upsilon\}$, $h_f(x) = sign(\Sigma_{t=1}^{M} C_t \times h_t(x))$

1. Initialize sample weights vector $W1 = (W_1^1, W_2^1, W_3^1, \cdots, W_{n_1+n_2}^1)$ and $\beta_1, \beta_2, \beta_1 = \beta_2$ as

$$w_i^1 = \begin{cases} 1/n_1, & i = 1, 2, 3, 4, \cdots, n_1 \\ 1/n_2, & i = n_1 + 1, \cdots, n_1 + n_2 \end{cases}$$

$$\beta_1 = 1/(1 + \sqrt{2 \log n_1/m}) = \beta_2$$

2. Do M times iterative training For $t = 1$, M do Step 1 to Step 6

Step 1: calculating sample weight distribution on the training data set D_t of each iteration:

$$p^t = \frac{w_i^t}{\sum_{i=1}^{n_1+n_2} w_i^t}$$

Step 2: Call the traditional classifier A using the combined training data D_t together with the weight distribution of each sample data p^t and data labels to train a new improved weak classifier $h_t : X \to \Upsilon$.

Step 3: Calculate classification error rate of the new weak classifier h_t on training data from the target region,

$$alpha = \sum_{i=n_1+1}^{n_1+n_2} \frac{w_i^t \mid h_t(x_i) - c(x_i) \mid}{\sum_{i=n_1+1}^{n_1+n_2} w_i^t}$$

Step 4: Update the weak classifier sample weight impact factor:

$$\beta_2 = \ln(\frac{1 - alpha}{alpha})$$

Step 5: Update the sample weight

$$w_i^{t+1} = \begin{cases} w_i^t \beta_1^{|h_t(x_i) - c(x_i)|}, & \text{when } i = 1, 2, 3, \cdots, n_1 \\ w_i^t \beta_2^{-|h_t(x_i) - c(x_i)|}, & \text{when } i = n_1 + 1, \cdots, n_1 + n_2 \end{cases}$$

Step 6: Update the weight of the new weak classifier $C_t = betaa \times \beta_2$, $betaa$ is an optimize factor that determine the sample label distribution range of plus or minus of error rate.

2. Algorithm analysis

First, define the sample weights of auxiliary training dataset Ds to ρ_i^t and define the sample weights of target domain dataset D_t to φ_i^t

$$\rho_i^t = \frac{w_i^t}{\sum_{j=1}^{n_1} w_j^t} \varphi_i^t = \frac{w_{i+n_1}^t}{\sum_{j=n_1}^{n_1+n_2} w_j^t}$$

Suppose the loss of n-dimensional sample of training data in a particular iteration is $loss(x_i)^{\mathrm{t}}$

$$loss(x_i)^t = |h_t(x_i) - c(x_i)|$$

where $i = 1, 2, \cdots, n$, and $t = 1, 2, \cdots, M$. Then the loss of sample data in the round of iteration for M is $L(x_i)$

$$L(x_i) = \sum_{t=1}^{M} loss(x_i)^t$$

1) Loss of auxiliary training samples

The total loss of auxiliary training samples from school education in M rounds of iteration can be computed

$$Ls = \sum_{t=1}^{M} \sum_{i=1}^{n_1} \rho_i^t loss(x_i)^t$$

According to the update rule of auxiliary training samples

$$\sum_{i=1}^{n_1} w_i^{t+1} = \sum_{i=1}^{n_1} w_i^t \beta_1^{|h_t(x_i) - c(x_i)|}$$

The average loss is not difficult to launch M iteration of Ls/M

$$\frac{Ls}{M} \le \min_{1 \le i \le n_1} \frac{L(x_i)}{M} + \sqrt{\frac{2 \ln(n_1)}{M}} + \frac{\ln(n_1)}{M}$$

Thus, it can be seen that the convergence speed of Tr_MAdaBoost algorithm is related to auxiliary training sample size n_1 and iteration times M, and the convergence speed is $O(\ln n_1/M)$. When the number of iterations M is sufficiently large, even iterations from $[M/2]$ *to* M, the weighted average loss of auxiliary training data will tend to zero.

$$\lim_{M \to \infty} \frac{\sum_{t=[\frac{M}{2}]}^{M} \sum_{i=1}^{n_1} \rho_i^t loss(x_i)^t}{M - [M/2]} = 0$$

2) Loss of training samples from target domain

Define error rate of the final classification of the training data from target domain is ϵ.

$$\epsilon = \Pr_{x \in Dt}[h_f(x_i) \neq c(x_i)] = \frac{|S|}{n_2}$$

According to the update rule and the error rate of training samples from target domain

$$\sum_{i=n_1+1}^{n_1+n_2} w_i^{t+1} = \sum_{i=n_1+1}^{n_1+n_2} w_i^t \beta_2^{-|h_t(xi) - c(xi)|}$$

$$\epsilon_t = \sum_{i=n_1+1}^{n_1+n_2} \varphi_i^t |h_t(x_i) - c(x_i)|$$

It can be computed that the upper bound of the final error rate of training data from target domain by

the following inequality

$$\epsilon = \prod_{t=\left[\frac{M}{2}\right]}^{M} \frac{1-(1-\epsilon_t)(1-\beta_2)}{\sqrt{\beta_2}} = 2^{\left[\frac{M}{2}\right]} \prod_{t=\left[\frac{M}{2}\right]}^{M} \sqrt{\epsilon_t(1-\epsilon_t)}$$

Thus, when $\epsilon_t < 0.5$,the training error of final classifier on the training dataset D_t from target domain will decreases with the increase of the number of iterations.

3. Algorithm implementation

Reference to the current more mature idea of ModestAdaBoost algorithm, use GML_AdaBoost Matlab_Tool-box to design Tr_MAdaBoost algorithm. The input parameters: *WeakLrn* is weak classifier using CART decision tree algorithm, *Data* is train data set, Labels is the tag of train data, Max_*Iter* is the maximum number of iteration, n_1 is the number of auxiliary train data from offline school education, n_2 is the sample number of train data from online education. The output parameters: Learner is a cell constructed learner, weights is the weight of learner, *final_hyp* is the prediction output of train data. The core code of Tr_MAdaBoost algorithm is shown in Algorithm 2.

Algorithm 2 The core code of Tr_MAdaBoost algorithm

function [Learners, Weights, final_hyp] = $Tr\_MAdaBoost$ ($WeakLrn, Data, Labels, Max\_Iter$, n_1, n_2)
% Initialize related arguments
$distr$ = [ones$(1, n_1)/n_1$, ones$(1, n_2)/n_2$];
$B1 = 1/(1 + \text{sqrt}(2 \times \log(n_1)/Max\_Iter))$;
$B2 = B1$;
%Choose the best weaklearner, formulatedistr and rev_distr to 1 For $It = 1 :$ Max_Iter
$nodes = train(WeakLrn, Data, Labels, distr)$;
for $i = 1$: length ($distr$)
if $i < n_1 + 1$
$distr(i) = distr(i)/\text{sum}(distr(1 : n_1))$;
$rev\_distr(i) = (1./distr(i))/\text{sum}\ ((1./distr(1 : n_1)))$;
elseif $i > n_1$&& $i < n_1 + n_2 + 1$
$distr(i) = distr(i)/\text{sum}(distr((n_1 + 1) : (n_1 + n_2)))$;
$rev\_distr(i) = (1./distr(i))/\text{sum}((1./distr((n_1 + 1) : (n_1 + n_2))))$;
end
end
%Renew the weak learner's weights C_t
for $i = 1$: length($nodes$)
$curr\_tr = nodes\{i\}$;
$step\_out = calc\_output(curr\_tr, Data)$;
$s1 = \text{sum}((Labels == 1) \times (step\_out) \times distr)$;
%The sum weights of positive class
$s2 = \text{sum}((Labels == -1) \times (step\_out) \times distr)$;
%The sum weights of negative class
$s1\_rev = \text{sum}((Labels == -1) \times (step\_out) \times rev\_distr)$;
$s2\_rev = \text{sum}((Labels == 1) \times (step\_out) \times rev\_distr)$;
$betaa = s1 \times (1 - s1\_rev) - s2 \times (1 - s2\_rev)$;
$C_t = -betaa \times B2$;
if(sign($betaa$) $\sim$ = sign$(s1 - s2)||(s1 + s2) == 0)$
continue;
end
$Weights(end + 1) = C_t$;
$Learners\{end + 1\} = curr\_tr$;
$final\_hyp = final\_hyp + step\_out \times C_t$;
%Output of the best weaklearner
end
%Renew the sample weights $distr$
for $j = 1 :$ length($distr$)
if $j < n_1 + 1$
$distr(j) = distr(j) \times B1. \wedge (\text{abs}(final\_hyp(j) - Labels(j)))$; elseif $j > n_1$ & & $i < n_1 + n_2 + 1$
$k1(j) = distr(j) \times \text{abs}(final\_hyp(j) - Labels(j))$ /sum$(distr(n_1 + 1 : n_1 + n_2))$;
$Alpha = \text{sum}(k1(n_1 + 1 : n_1 + n_2))$;
$+\ B2 = \log((1 - Alpha)/Alpha)$;
$distr(j) = distr(j) \times B2. \wedge (-\text{abs}(final\_hyp(j) - Labels(j)))$;
end

IV. Algorithm Validation and Evaluation

To verify the algorithm, using two students ZhangXin and LiMing's school test scores, online learning situations for example. Due to the flexible way of online learning, the learning data may be format diversity. For facilitating observation and experimentation, the school education mainly takes 10 times Chinese, math, English as auxiliary training data, online learning mainly takes two students in the "Chinese online learning", "English online learning", and "online learning mathematics" three platforms' five studies as the source data, two students' online learning to produce all of the data as the target domain data. Experimental environment is Win7+MATLAB2014a+Visual Studio 2010.

1. Data preprocessing

Pointed, online platform due to the various branches study carried on the detailed division, the representation of data is also different. Therefore, before classification processing, first of all to get online data for the pre-treatment of the unified, standardized, select the appropriate data to construct the training set and testing set. Here, training set *TrainData* consists of two parts, *Data1* and *Data5*. *Data1* were 20 groups taken from offline school, contains 10 times' Chinese, math, English test scores of two students; *Data5* were 240 set of data from online learning that contains two students': Chinese (tiankong yuedu xuanze, xiezuo), math (compute, apply, choose and answer), English (write, translate, listen, words) from three subjects of 12 training online learning situation, forming 21 groups after the merger process. Test set *ControlData* from 150 groups, online learning data merging for 10 groups after treatment. The related Matlab

code is shown in Algorithm 3.

Algorithm 3 The code of data preprocessing

% Step1: reading Data from the file
For $i = 1 : 4$ do
$file\_data\_i$ =load('F:\20160510dataconstruct*.mat');
$Data1 = [file\_data1.Chinese,\ file\_data1.Maths,$
$file\_data1.English]$;
$Data2 = (file\_data2.zongfen/4)$;
$Data3 = (file\_data3.score/4)$;
$Data4 = (file\_data4.sum/4)$;
$Labels1 = (file\_data1.xuehao - 20120231)$
$\times(-2)/11 - 1$;
$Labels2 = (file\_data2.Cid - 808)/2 + 1$;
$Labels3 = file\_data3.Mid - 409$;
$Labels4 = file\_data4.Eid - 302$;
% Step2: splitting data to training and control set
$Data5 = [Data2(2 : 8),\ Data3(2 : 8), Data4(2 : 8)]$;
$TrainData = [Data1;\ Data5]'$;
$TrainLabels = [Labels1',\ (Labels2(2 : 8))']$;
$ControlData = [Data2(1 : 10), Data3(1 : 10),$
$Data4(1 : 10)]'$; $ControlLabels = (Labels2(1 : 10))'$;

2. The analysis of algorithm experiment result

1) Comparison experiment with different distribution data

Respectively take RealAdaboost, GentleAdaboost, ModestAdaBoost, Tr_MAdaBoost algorithm on the training set and testing set to do contrast experiments. The classification result of the four algorithms and the test error rate, F-msasure value contrast, were shown from Fig.2 to Fig.7. It is not hard to find that because the offline and online data has different distribution, the Real, Gentle, ModestAdaboost classification accuracy were just 70%, 90%, 60%, significantly less than 100% of Tr_MAdaBoost. As is shown on the Error rate comparison chart, with the increase of the number of iterations, four kinds of algorithm overall classification error rate is on the decline, error rate of Tr_MAdaBoost finally attributable to zero, and eventually Gentle tend to be 20%, Real and Modest tend to be 30%, and after 6 iterations Tr_MAdaBoosttest error rate will tend to zero, and F-measure value is the highest of all algorithms. This is because the algorithm is introduced Tr_MAdaBoost some online data as training data subject, through AdaBoost ideas at the same time to strengthen the emphasis on the training data from the target domain, so can realize correct classification.

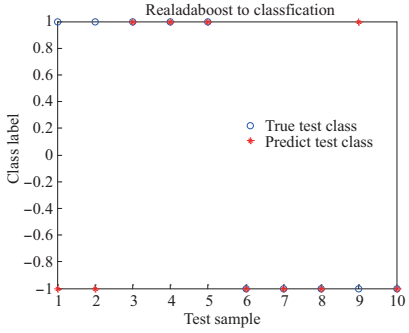

Fig. 2. RealAdaBoost in experiment I

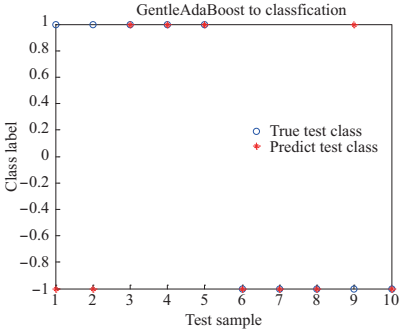

Fig. 3. GentleAdaBoostin experiment I

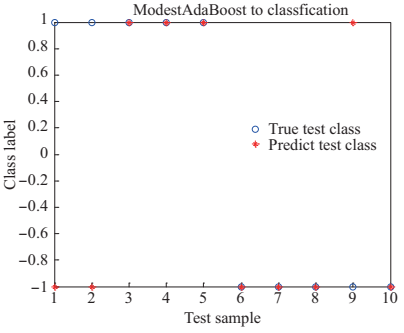

Fig. 4. ModestAdaBoost in experiment I

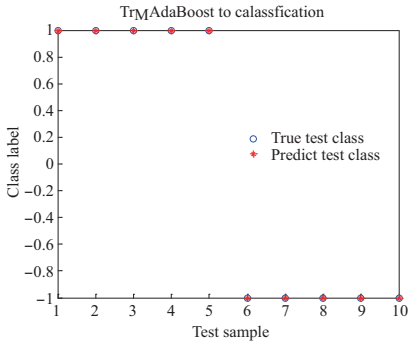

Fig. 5. Tr-MAdaBoostin experiment I

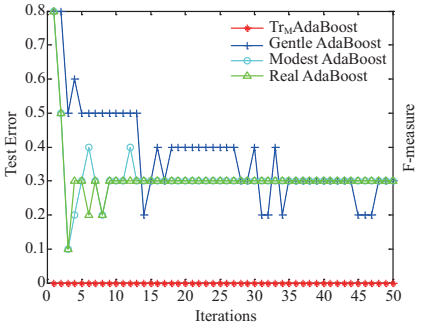

Fig. 6. Test error in experiment I

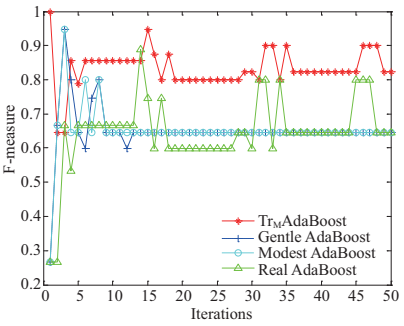

Fig. 7. F-measure in experiment I

As it is difficult to acquire large-scare online education dataset, we can use dataset Isolet to construct train and test data, which can be suited to use transfer learning. Isolet contains 26 categories, 6238 sets of data, and 617 attributes for each set of data. For the data distribution of each category is different but related, it is necessary to construct the train and test datasets to extract the needs of the transfer learning by taking some different attribute data. The train data consists of the data with $label = 1$ in the previous 3120 records and data with $label = 2$ in the last 3118 records. The test set consists of the data with $label = 2$ in the previous 3120 records and data with $label = 1$ in the last 3118 records. Taking the same

method to do contrast experiments, and the results are shown form Fig.8 to Fig.13. According to Fig.8, Fig.9, Fig.10 Fig.11, it is easy to find that Tr_MAdaBoost has the lowest mistake rate, only 1 point predicted wrong while RealAdaboost has 24 points, GentleAdaBoost and ModestAdaBoost both has 5 points. It also can be find From Fig.12 which described the test error rate of the four algorithms that Tr_MAdaBoost has the lowest test error, tending to 0.02, and RealAdaBoost has the highest test error, always keeping around 0.21, and the test error of ModestAdaBoost varied fast with iterations. As it shows in Fig.13, the F-measure of Tr_MAdaBoost is the highest, especially when *iterations* $>$ 20, keeping around 0.96, while ModestAdaBoost has the lowest F-measure, keeping around 0.65. From the analysis above, Tr_MAdaBoost algorithm shows the better classification performance than the other three algorithms, for it added transfer learning idea into traditional AdaBoost algorithm.

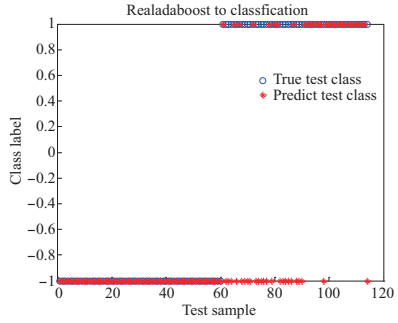

Fig. 8. RealAdaBoost in experiment II

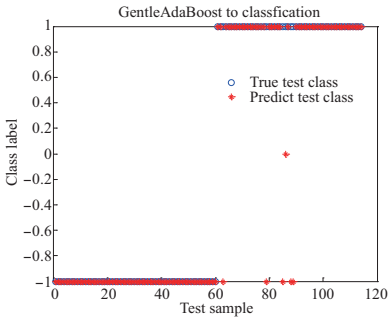

Fig. 9. GentleAdaBoost in experiment II

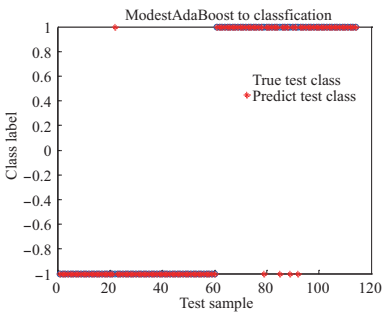

Fig. 10. ModestAdaBoost in experiment II

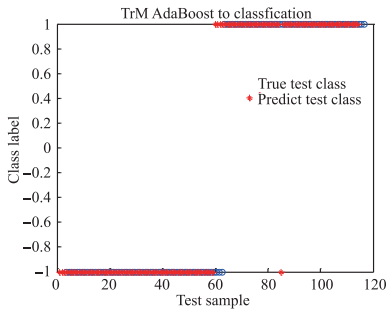

Fig. 11. Tr-MAdaBoost in experiment II

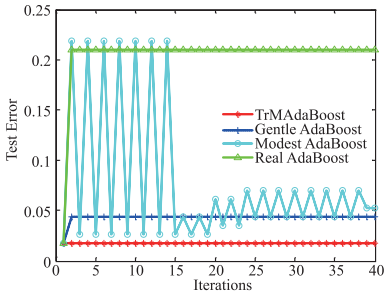

Fig. 12. Test error in experiment II

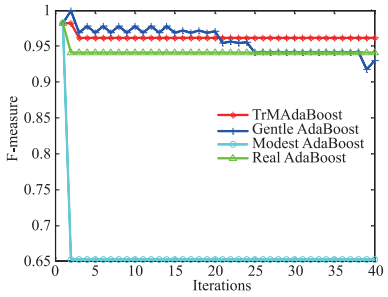

Fig. 13. F-measure in experiment II

2) Comparison experiment with the same distribution data

In order to check Tr_MAdaBoost algorithm's generalization ability, use dataset Ionosphere which consists of 351 groups of data contains 35 characteristics, to do contrast verif-verification. The storage format of Ionosphere dataset is shown as Table 1. Respectively use the four algorithms to do classification on Ionosphere. *TrainData* consists of odd number column, and *ControlData* made up of even number column. Since the *TrainData* of Tr_MAdaBoost consists of two parts, randomly selected $n_1 = 50$, $n_2 = 126$, classification results and the testerror rates, F-measure value along with the change of the number of iterations as shown from Fig.14 to Fig.19. It can be seen from the figures, when iterations equal 40, classification accuracy of Real, Gentle, Modest and Tr_MAdaBoost respectively are 91.5%, 89.8%, 90.4% and 95.5%. Especially when the number of iterations greater than 5, test error rates of Tr_MAdaBoost tend to be lowest around 5%, and with the highest F-measure value tend to be 0.9. It also can be seen from the contrast that Tr_MAdaBoost are correctly classified with the data having the same distribution, but the accuracy of classification influenced by the *TrainData* constituent ratio n_1: n_2 and the number of iterations. When n_1, n_2, select different values, the accuracy of classification is larger fluctuation, this is mainly because the introduction of the purpose of these two parameters is to do transfer learning on the different distribution data. After many experiments, it can be found that when $n_1 < n_2$, and the

Table 1. Data format of ionosphere

1, 0, 0.99539, −0.05889, 0.85243, 0.02306, 0.83398, −0.37708, 1, 0.03760, 0.85243, 0.17755, −0.59755, −0.44945, 0.60536, 0.38223, 0.84356, 0.38542, 0.58212, −0.32192, 0.56971, −0.29674, 0.36946, −0.47357, 0.56811, −0.51171, 0.41078, −0.46168, 0.21266, −0.34090, 0.42267, −0.54487, 0.1864, −0.45300, g
1, 0, 1, −0.18829, 0.93035, −0.36156, −0.10868, −0.93597, 1, −0.04549, 0.50874, −0.67743, 0.34432, −0.69707, −0.51685, −0.97515, 0.05499, −0.62237, 0.33109, −1, 0.13151, −0.45300, −0.18056, 0.35734, 0.20332, −0.26569, 0.20468, −0.18401, −0.19040, −0.11593, 0.16626, −0.06288, −0.13738, −0.02447, b
1, 0, 1, 0.03365, 1, 0.00485, 1, 0.12062, 0.88965, 0.01198, 0.73082, 0.05346, 0.85443, 0.00827, 0.54591, 0.00299, 0.83775, −0.13644, 0.75535, 0.08540, −0.70887, −0.27502, 0.43385, −0.12062, 0.57528, −0.40220, 0.58984, −0.22145, 0.43100, −0.17365, 0.60436, −0.24180, 0.56045, −0.38238, g

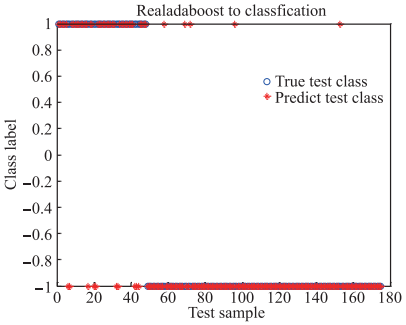

Fig. 14. RealAdaBoostin experiment III

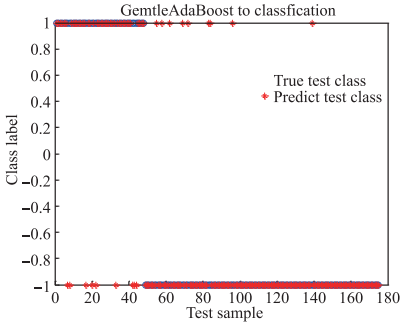

Fig. 15. GentleAdaBoost in experiment III

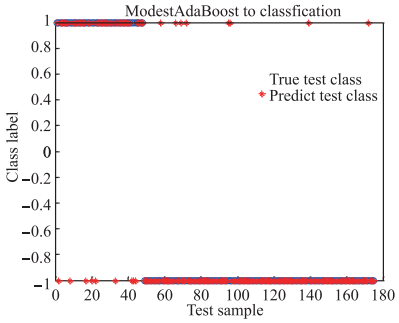

Fig. 16. ModestAdaBoost in experiment III

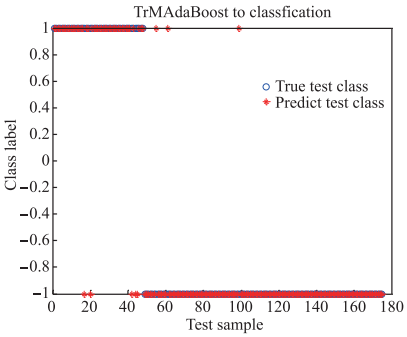

Fig. 17. TrMAdaBoost in experiment III

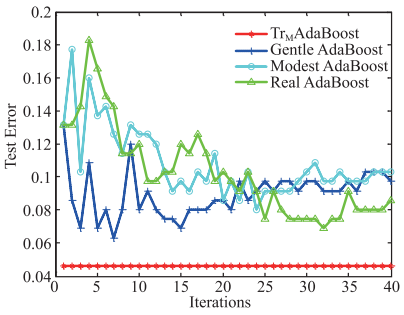

Fig. 18. Test Error in experiment III

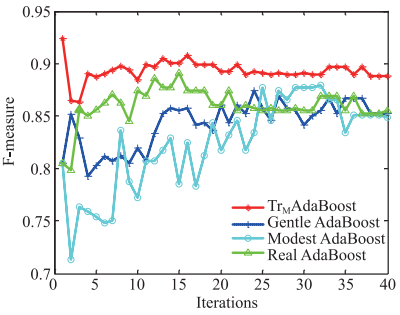

Fig. 19. F-measure in experiment III

number of iterations greater than 5, Tr_MAdaBoost algorithm has lower test error rates (Fig.18) and higher F value (Fig.19) than Real, Gentle and Modest AdaBoost.

V. Conclusions and Prospect

This article in view of the current online education present situation and the characteristics of online data, based on the Hadoop distributed system, build an online education data classification model based on transfer learning, and firstly apply transfer learning to do data mining on online education data. In order to achieve the effective transfer of classification model, by using AdaBoost thought to presents Tr-MAdaBoost algorithm. Experiments show that Tr-MAdaBoost algorithm overcomes the requirements of traditional classification algorithm that the train data must have independent identical distribution, while it can classify online data with different distribution correctly, and also can be adapt to traditional machine learning field. At the same time, the introduction of Hadoop parallel processing architecture, to improve the accuracy of data classification also can greatly enhance the efficiency of data processing, which create favorable conditions for learning analysis, and provide effective personalized learning of online education in the era of big data. However, due to the affected of constraints related to the strength of the field and the experimental sample collection methods, practical application of the algorithm is yet to be verified. In the next step of work, the study will focus on online data preprocessing method and correlation in the field of verification, to improve the scientific, validity and practicability of the algorithm.

References

[1] Dai W, “Instance-based and feature-based transfer learning”, Diss, Shanghai Jiao Tong University, pp.4–5, 2009. (In Chinese)

[2] Dai W, Yang Q, Xue G R, *et al.*, “Boosting for transfer learning”, *Proceedings of the 24th International Conference on Machine learning*, ACM, Vol.238, pp.193-200, 2007.

[3] Wei F, Zhang J, Yan C, *et al.*, “FSFP: Transfer learning from long texts to the short”, *Applied Mathematics & Information Sciences*, Vol.8, No.4, pp.2033–2040, 2014.

[4] Kuzborskij I, Orabona F, Caputo B, *et al.*, “Scalable greedy algorithms for transfer learning”, *Computer Vision & Image Understanding*, Vol.156, pp.174–185, 2016.

[5] Wei X, Zhou S G, Guan J H, *et al.*, “Classification in networked data: A survey”, *Pattern Recognition & Artificial Intelligence*, Vol.24, No.4, pp.527–537, 2011.

[6] Meesookho C, Narayanan S, Raghavendra C S, *et al.*, “Collaborative classification applications in sensor networks”, *Sensor Array and Multichannel Signal Processing Workshop Proceedings*, IEEE, pp.370–374, 2002.

[7] Shang R, Zhang Z, Jiao L, *et al.*, “Global discriminative-based nonnegative spectral clustering”, *Pattern Recognition*, Vol.55, No.C, pp.172–182, 2016.

[8] Shang R, Zhang Z, Jiao L, *et al.*, “Self-representation based dual-graph regularized feature selection clustering”, *Neurocomputing*, Vol.171, pp.1242–1253, 2016.

[9] Shang R, Wang W, Stolkin R, *et al.*, “Non-negative spectral learning and sparse regression-based dual-graph regularized

feature selection", *IEEE Transactions on Cybernetics*, Vol.48, No.2, pp.793–806, 2018.

[10] Wang C, "A geometric framework for transfer learning using manifold alignment", *Dissertations & Theses,* University of Massachuseths Amherst, 2010.

[11] Yao Y and Doretto G, "Boosting for transfer learning with multiple sources", *IEEE Conference on Computer Vision & Pattern Recognition*, pp.1855–1862, 2010.

[12] Kandaswamy C, Silva L M, Alexandre L A, *et al.*, "Improving transfer learning accuracy by reusing Stacked Denoising Autoencoders", *IEEE International Conference on Systems, Man and Cybernetics*, IEEE, pp.1380–1387, 2014.

[13] Chaturvedi I, Ong Y S, Arumugam R V, *et al.*, "Deep transfer learning for classification of time-delayed Gaussian networks", *Signal Processing*, Vol.110, No.C, pp.250–262, 2015.

[14] Fang M, Guo Y, Zhang X, *et al.*, "Multi-source transfer learning based on label shared subspace", *Pattern Recognition Letters*, Vol.51, No.C, pp.101–106, 2015.

[15] Nguyen T T, Silander T, Li Z, *et al.*, "Scalable transfer learning in heterogeneous, dynamic environments", *Artificial Intelligence*, Vol.247, pp.70–94, 2017.

[16] Y Freund and R Schapire, "A decision-theoretic generalization of n-line learning and an application to boosting", *Helvetica ChimicaActa*, Vol.55, No.7, pp.119–139, 2010.

[17] Wu S and Nagahashi H, "Analysis of generalization ability for different adaboost variants based on classification and regression Trees", *Journal of Electrical & Computer Engineering*, Vol.2015, pp.1–17, 2015.

[18] Demirkır C and Sankur B, "Face detection using look-up table based gentle adaboost", *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, pp.339–345, 2005.

[19] Sam K T and Tian X L, "Vehicle logo recognition using modest adaboost and radial tchebichef moments", *International Proceedings of Computer Science & Information Tech*, Vol.25, pp.91–95, 2012.

YU Lasheng is a vice professor in Central South University of China, ACM and CCF member, ACM/ICPC golden medal coach. He received the B.S. degree in computer science, the M.S. and Ph.D. degrees in control theory and control engineering from Central South University. He is the editor of *Journal of Convergence Information Technology* and *Advances in Information Sciences and Service Sciences etc.*, he is also the reviewer for the journals such as *Future Generation Computer Systems*, *Journal of Parallel and Distributed Computing*, *Artificial Intelligence Review, etc.* He has published at least 70 papers on agent technologies or algorithms, and has published 3 books. His main research interests include agent technologies and applications, structure and algorithm, smart computing, *etc.* (Email: ley462@163.com)

WU Xu was born in Suining, Sichuan Proviuce, China. He received the M.S. degree in computer science and technology from Central South University China. (Email: 187353898@qq.com)

YANG Yu was born in Huaihua, Hunan Proviuce, China. He received the M.S. degree in computer science and technology from Master graduate of Central South University China.
(Email: 120486638@qq.com)