A Weighted-resampling based Transfer Learning Algorithm

Xiaobo Liu, Zhentao Liu, Guangjun Wang
School of Automation
China University of Geosciences
Wuhan, China 430074
Hubei key Laboratory of Advanced Control
and Intelligent Automation for Complex Systems
Wuhan, China 430074
Email: {xbliu, liuzhentao, gjwang}@cug.edu.cn

Zhihua Cai
School of Computer Science
China University of Geosciences
Wuhan, China 430074
Email: zhcai@@cug.edu.cn

Harry Zhang
Faculty of Computer Science
University of New Brunswick
NB, Canada E3B 5A3
Email: hzhang@unb.ca

***Abstract*—Transfer learning has attracted more and more attention, and many scholars proposed some useful strategies. Boosting is the main strategy for transfer learning. In boosting, resampling is preferred over reweighting, and it can be applied to any base learner. In this paper, we propose a weighted-resampling method for transfer learning, called TrResampling. Firstly, resampling is applied to the data with heaven weight in the source domain, and the resampled data is used with the target data as the training data to build a classifier. Then the TrAdaBoost algorithm is used to adjust the weights of source data and target data. We discuss Decision Tree, Naive Bayes, and SVM as the base learner in TrResampling, and choose the suitable for TrResampling. In order to illustrate the performance of the proposed algorithm, we compare TrResampling with the state-of-the-art algorithm TrAdaBoost and the base learner Decision Tree, experimental results on UCI data sets indicate that TrResampling is superior to TrAdaBoost and Decision Tree on many data sets.**

I. Introduction

Transfer learning is a branch of machine learning, which can transfers useful knowledge from the source domain to the target domain for building a good mode for classification [1]. As transfer learning's population, many scholars combine transfer learning with other algorithms to solve many real world problems. Such as, Gong et al. [2] used transfer learning and differential evolution to build the fuel cell models. Feng et al. [3] presented a memetic computational paradigm based on evolutionary optimization and transfer learning. Xiao et al. [4] combine ensemble learning and transfer learning to build customer credit scoring classifier ensemble model. Acharya et al. [5] proposed an optimization framework for combining ensembles of classifiers and clusterers with applications to nontransductive semisupervised learning and transfer learning. Kandaswamy et al. [6] proposed an ensemble of deep transfer learning methodology for character recognition, biomedical image recognition, and so on. Bhatt et al. [7] combine the co-transfer learning and ensemble learning to solve the cross-resolution face matching problem. Mei [8] proposed a SVM ensemble based transfer learning model for membrane proteins discrimination.

Boosting is one of the most widely used ensemble methods, which can significantly improve the performance of any learning algorithm [9]. The main process of boosting is: many classifiers are learned on training data by a given weak learner, then combining the classifiers produced by the weak learner into a single composite classifier. In every iteration, the training data's distribution are different, because the next iteration focus on the misclassified examples of previous iteration. In boosting, the classifiers learned in every iteration only need to be a little bit better than random guessing, and the true practical value of boosting can only be assessed by testing the method on real machine learning problems [10].

AdaBoost, short for "adaptive boosting", is the most famous boosting algorithm, which self-adapts to the training data [11]. In AdaBoost, if an example is correctly classified, its weight will decrease, otherwise, its weight will increase. That's to say, the higher example's weight is, the harder to classify. So in the next iteration, the learner pay more attention to the misclassified examples. AdaBoost by repeatedly learning the same training set and combining an arbitrary number of weak classifiers, it not only does not suffer from overfitting even after a larger number of iterations, but also sometimes it is even able to reduce the generalization error after the training error has already reached zero [12]. As AdaBoost is simple and robust, it has been successfully applied in many fields, such as text classification[13], face recognition [14], melanoma diagnosis [15], and so on.

Many researches have applied AdaBoost in transfer learning, such as TrAdaBoost [16], TransferBoost [17], TrBagging [18]. In fact, boosting by resampling is prefered over boosting by reweighting [10]. Boosting by reweighting pass each instance's numerical weight to the base learner, which requires the base learner can handle the instance weight information. Boosting by resampling can be applied to any base learner, which creates a new training data set with the same size as the original training data set, and then selects the higher weight instances to it.

In this paper, we propose a weighted resampling based transfer learning algorithm, through resampling the weighted

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

source data, add the higher weights data in the source domain and the labeled data in the target domain in the training data set. Then, we discuss different base learners to learn a classification model on the training data set. Moreover, the TrAdaBoost strategy is utilized to adjust the weights of training data, the weights of misclassified data in source domain are decreased and the misclassified data in target domain are increased. At the end, a new classification model is learned on the reweighted labeled data.

Furthermore, we discuss three classic machine learning algorithms as the baser classifiers in transfer learning [19]. Firstly, C4.5 is a decision tree based algorithm, derives from the simple divide and conquer algorithm for growing the decision tree, which very flexible, easy to understand, and easy to debug. Naive Bayes probabilistic classifiers are commonly studied in machine learning. It is very easy to construct, without needing any complicated iterative parameter estimation schemes, and non-sensitive to the miss data. Support Vector Machine (SVM) is an algorithm that uses a nonlinear mapping to transform the original training data into a higher dimension. SVM has the ability to build model on the complex nonlinear decision boundaries, so it is more robust and accurate, and less prone to over fitting than other methods.

II. Related Work

Like humans cognition, transfer learning can recognize new things according to the knowledge that is learned from other things. In recently years, based on the idea of transfer learning that can effectively utilizes the old domains knowledge to identify the targets of unknown domain, a lot of improved transfer learning methods had been proposed.

TrAdaBoost [16] is an improved transfer learning algorithm based on AdaBoost, which can adaptively change the weights of the source data and target data. In the source data set, if an example is misclassified, meaning the example's distribution is very different from the target data set, so its weight will be reduced. In the target data set, TrAdaBoost uses the standard AdaBoost to increase the weight of misclassified examples. From the results, TrAdaBoost is a high-performance and simple transfer learning algorithm, has strong transfer ability and well convergence. However, several problems exist in TrAdaBoost, such as follows.

Eaton et al. [17] pointed out the fact that TrAdaBoost discards the first half ensemble will degrade the ensemble's performance in practice. So they proposed novel set-based boosting for instance-based transfer TransferBoost, assigning higher weight to source tasks that have positive transferability to target tasks and using AdaBoost to adjust the weight of instances in source tasks. When given a mix of relevant and irrelevant source data, TransferBoost outperforms TrAdaBoost.

La et al. [20] pointed out that when TrAdaBoost makes distribution similarity evaluation, it just uses the source data set as the distribution test set, so that it is hard to say whether the misclassified examples in the source data set was caused by difference or difficulty. So they proposed a two-stage distribution evaluation strategy (TSDE) to measure the distance between same distributed base source data set and target data set. From the simulations and experimental results, the method proposed by La et al. is robust, and it has low time consumption and high accuracy.

Kamishima et al. [18] proposed a TrBagging method that is an extension of bagging [21]. In the learning phase, training data sets are bootstrap-sampled from target and source data, and weak classifiers are learned from these training sets. In the filtering phase, some good weak classifiers are selected based on their accuracy in the target data, while other weak classifiers are abandoned if their accuracy in target data is low. TrBagging works easily without the need to severely tune learning parameters. If the amount of source data is significantly larger than that of target data, however, the imbalance adversely influence its performance.

Xu et al. [22] proposed Multi-View Transfer Learning with AdaBoost. Firstly, they constructed a multi-view transfer learning, which through combined source and target task to constituent views, and these two tasks can be learned from every views at the same time. Then they used the AdaBoost method to update the weights of data according to their contribute model. This algorithm proved useful and effective on binary classification problems.

In the above related work, most of AdaBoost methods are done by reweighting. However, some learning algorithms require an unweighted set of examples [11]. By comparing the results of 10 boosting algorithms, with 4 learners on 15 data sets, Seiffert et al. [10] found that boosting by resampling is preferred over by reweighting generally.

III. Weighted-resampling based Transfer Learning Algorithm

In TrResampling, a new training data set is created and has the same size as the original training data sets, then the data in original training data sets with higher weights are selected under higher probability to the new training data. That means, the data in the resampled training data set, the more frequently it appears, the more likely to be correctly classified. The detail shown in Algorithm 1.

Algorithm 1 weighted-resampling Framework

```
Input:
    $D$, a sequence of $N$ examples in data set $D$
randomly initial the weights of data set $D$
create a new data set $D'$ with the same size of data set $D$
give each example of data set $D$ a resampling probabilities
for $i$=1 to N do
    if $\sum_{j=1}^{j=i} probabilities[i] <= \sum_{j=1}^{j=i} weights[i]$ then
        add examples[i] to $D'$
    end if
end for
Output: $D'$
```

The main idea of TrResampling algorithm as follow. In each iteration round, a new source training data set is created by weighted-resampling from the original source data sets, then the labeled data in the target data set are added to the new source training data set as the training data set. As a result, the data assigned higher weights in the source data set will receive more emphasis. Then, the TrAdaBoost strategy is utilized to adjust the source data's influence on building model. Thus, the misclassified data in the source data set will have lower weights in the next round. At the end, the final hypothesis is combined by all model that are built on each iteration. The framework of TrResampling is presented in Algorithm 2.

According to the definition of transfer learning [1], data in source domain D_S and data in target domain D_T have the same feature space $\mathcal{X}$ but with different distributions. $D_S = \{(x_1^S, y_1^S), ..., (x_m^S, y_m^S)\}$, where $x_i^S \in \mathcal{X}_S$ is an instance and $y_i^S \in \mathcal{Y}_S$ is the corresponding class label. $D_T = \{(x_1^T, y_1^T), ..., (x_n^T, y_n^T)\}$, where $x_i^T \in \mathcal{X}_T$ is an instance in the target domain and $y_i^T \in \mathcal{Y}_T$ is the corresponding class label. $\mathcal{Y}$ is the label space. Training set *D* includes D_S and D_T. The test data set D_{Test} has the same distribution as the target domain. In our setup, a significant number of labeled instances in the source domain are available, and only a few labeled instances are available in the target domain, $0 < n \ll m$.

Algorithm 2 TrResampling Framework

```
Input:
    D, a sequence of m labeled examples in source data set $D_S$ and a sequence of n labeled examples in target data set $D_T$ with labels $y_i \in \mathcal{Y}$.
    BaseLearner is the base learning algorithm.
1:  $D'$=D with instance weights assigned to be 1.
2:  for t=1 to T do
3:      $D'$= weighted-resampling($D_S$)+ $D_T$
4:      $h_t$=BaseLearner($D'$).
5:      Calculate the error of $h_t$ on $D_T$ as
        $\varepsilon_t = \sum_{i=n+1}^{n+m} \frac{w_i^t |h_t(x_i) - y_i|}{\sum_{i=n+1}^{n+m} w_i^t}$
6:      t++
7:      if $\varepsilon_t > 0.5$ or t=T then
8:          go to Step 5.
9:      else
10:         Set $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$, $\beta = \frac{1}{1+\sqrt{2ln\frac{n}{m+n}}}$.
11:     end if
12:     Update the new weight vector:
        $w_i^{t+1} = \begin{cases} w_i^t \beta^{|h_t(x_i) - c(x_i)|}, 1 \le i \le n \\ w_i^t \beta_t^{-|h_t(x_i) - c(x_i)|}, n+1 \le i \le m \end{cases}$
13: end for
Output: $H^*(x) = argmax_{y \in Y} \sum_{t:h_t(x)=y_i} log\frac{1}{\beta}_t$.
```

IV. Experimental Evaluation

A. Data Sets

To demonstrate our algorithm's effectiveness, we conducted experiments on 36 representative data sets from the UCI repository[1]. The 36 UCI data sets,, which has been well used for algorithm validation [23] [24] [25] [26], represent a wide range of domains and data characteristics, but data set *labor* (57 instances) and *zoo* (101 instances) are too small, and *audiology* is unsuitable for transfer learning, thus we used the remaining 33 data sets in experiments. All data set were preprocessed the same way as our paper [27].

To use a data set for transfer learning, we must create two sets, namely, the source and target domains. One way to do this is to select a binary attribute and use its values to split the data set into two sets, each corresponding to one value of the selected attribute. Intuitively, the two resulting data sets will have different distributions. In this paper, we choose the attribute in every data set for splitting both the binary attribute and the multi-values attribute. The detail of which attribute is selected for splitting and the corresponding data set distribution can be seen in [27].

B. The Base Learner of TrResampling

In this paper, Decision Tree(C4.5), Naive Bayes(NB), and SVM are used as the base learner in our proposed TrResampling algorithm, such as TrRes-C4.5, TrRes-NB, and TrRes-SVM. To seek an accurate error estimate, the algorithm with different base learner repeat run 10 times tenfold cross-validation, and the results are averaged.

TABLE I: Prediction Accuracy of Different Base Learner in TrResampling

Data Set	TrRes-C4.5	TrRes-NB	TrRes-SVM
anneal	**99.67**	99.45	99.59
anneal.ORIG	90.53	97.99	**91.48**
autos	**79.5**	72.72	79.18
balance-scale	70.76	81.43	**86.26**
breast-cancer	64.04	66.52	**67.91**
breast-w	**93.06**	88.54	89.53
colic	**79.4**	74.44	74.91
colic.ORIG	**78.73**	72.06	73.54
credit-a	**90.16**	85.64	89.4
credit-g	72.06	**72.69**	71.11
diabetes	68.05	**71.75**	70.54
glass	**58.07**	45.48	50.81
heart-c	**90.79**	77.60	86.48
heart-h	**82.94**	74.14	76.72
heart-statlog	**84.36**	77.53	79.93
hepatitis	83.84	84.89	87.41
hypothyroid	90.92	91.91	**92.29**
ionosphere	**87.08**	91.62	81.43
iris	**92.29**	88.14	86.39
kr-vs-kp	**99.44**	94.76	98.94
letter	85.93	70.96	**87.47**
lymph	**88.71**	84.44	84.72
mushroom	100	100	100
primary-tumor	38.58	**46.18**	39.65
segment	95.59	92.69	**95.74**
sick	**97.62**	96.08	97.53
sonar	68.88	70.52	**74.32**
soybean	**94.25**	90.54	93.64
splice	**94.12**	89.96	93.2
vehicle	**80.11**	68.04	75.62
vote	96.28	95.58	**96.33**
vowel	87.63	77.78	**89.98**
waveform	86.48	88.80	**90.91**
W/T/L	19/1/13	3/1/29	11/1/21
Rank	**2.30**	1.45	2.24

From Table I we can get the conclusion that C4.5 has better performance than the Naive Bayes and SVM as the base

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

learner of TrResampling. The results demonstrated the Kearns and Mansour's argument that C4.5 can be viewed as a kind of boosting algorithm [28]. On the 33 data sets, TrResampling with C4.5 defeats Naive Bayes on 23 data sets, and defeat SVM on 19 data sets. In order to clearly demonstrate the advantage of C4.5 based TrResampling, we recall Friedman test [29] on TrRes-C4.5, TrRes-NB, and TrRes-SVM. From the average ranks of three algorithms, we consider C4.5 as the base learner designed to handle the unweighted training examples benefits the resampling.

C. Results and Analysis

To demonstrate our algorithm's effectiveness on transfer learning, we conducted experiments on 33 data sets. The base learner of TrResampling and TrAdaBoost are C4.5. Our experiments compare TrResampling using two algorithms, i.e., TrAdaBoost and C4.5, and the number of boosting iterations is set to 100, each algorithm runs 10 time, and results are averaged. For each target data set, the percentage of labeled data varies from 1% to 90%. We conducted a two-tailed T-test with a 95% confidence level to compare TrAdaBoost and C4.5 to our new algorithm.

In Fig.1, we focus on the *heart-statlog*, *segment*, *vowel*, and *waveform* four data sets, the whole experimental results showed in Table III. The x axis is the percent of the labeled data in the target domain, which are 1%, 3%, 5%, 10%, 30%, 60%, 90% respectively. The y axis is the Accuracy. As seen in Fig.1, when the ratio of labeled data in target domain is lower than 10%, the TrResampling's transferability is better than TrAdaBoost. Comparing with C4.5, TrResampling shows strong transferability, when the ratio of labeled data in target domain arrived at 90%, however, the classification accuracy are similar, which means that the less training data the more useful of transfer learning.

In Table II, TrResampling and TrAdaBoost show their transferability on the 33 UCI data sets. From the whole table, TrResampling and TrAdaBoost are useful on 24 data sets. But when the ratio of labeled data in target domain is lower than 30%, TrResampling and TrAdaBoost jeopardize the classification on the "auto", "colic", "heart-h", "hypothyroid", and "lymph", which means that negative transfer is still a problem need to solve.

On the 33 UCI data sets, TrResampling and TrAdaBoost has similar performance. When the ratio of labeled data in target domain is lower than 30%, the prediction accuracy of TrResampling is near to TrAdaBoost, their divergence are almost 1%. When the ratio of labeled data in target domain is higher than 30%, the training data sets has enough data to train a classifier, TrResampling is better than TrAdaBoost to help the base learner to learn an accuracy classifier.

Furthermore, we recall nonparametric statistical test to clarify the performances of TrResampling, TrAdaBoost, and C4.5. Friedman test is employed to detect the overall performances of all testes algorithms regarding to Accuracy. Friedman testing results are presented in Table II, when the ratio of labeled data in target domain are 30% and 90%, TrResampling are higher ranking than TrAdaBoost. Compare with C4.5, TrResampling has higher ranking than it regardless the ratio of labeled data in target domain. From the p-value of Friedman test, TrResampling has significant differences comparing with TrAdaBoost and C4.5, which implies TrResampling significantly outperforms TrAdaBoost and C4.5 on many data sets.

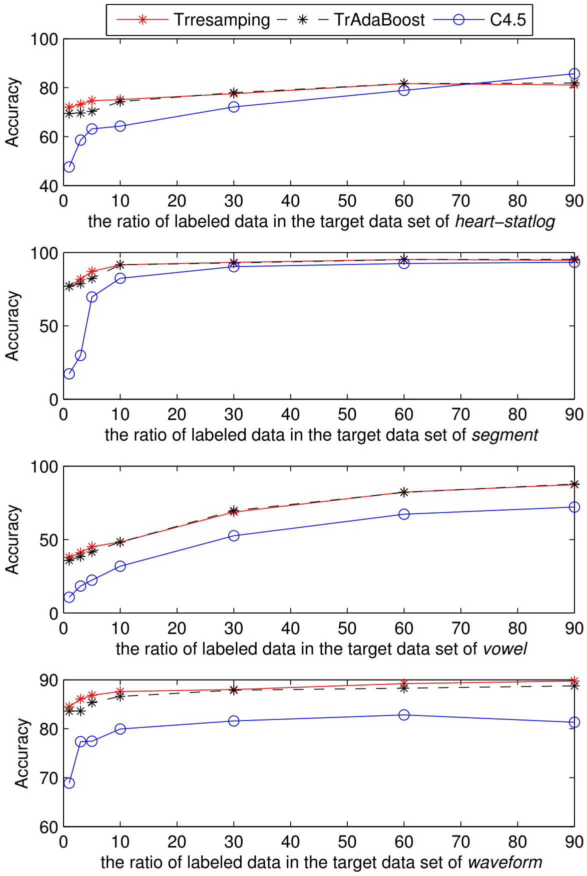

Fig. 1: The classification Accuracy on Four UCI data sets

V. Conclusion

A novel transfer learning algorithm TrResampling is proposed in our paper, which not only resamples the data with heaven weight in source domain data , but also reweights the data in source domain and target domain. Experimental results show that our algorithm has outstanding performance comparing with TrAdaBoost on many UCI datasets, and obviously improve the classification accuracy based on the base learners. At the same time, we found C4.5 more suitable as the base learner of TrResamping than Naive Bayes and SVM.

Although TrResampling has good performance on 33 UCI data set, the negative transfer problem still exist. The main reason that induces the negative transfer is the the target domain has very few labeled training data, causing target domain and the source domain to be too dissimilar. In the further research, we will try to divide the data in the source domain into many subsets, so that it enables balance to the target domain, and can effectively avoid negative transfer problem.

Acknowledgment

We thank the reviewers for their valuable comments and suggestions. This work was partially supported by the National Natural Science Foundation of China under grant No. 61603355, No. 61403422, and the Hubei Provincial Natural Science Foundation of China under Grant 2015CFA010.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345 – 1359, Oct. 2010.

[2] W. Gong, X. Yan, X.Liu, and Z. Cai, "Parameter extraction of different fuel cell models with transferred adaptive differential evolution," *Energy*, vol. 86, pp. 139–151, Jun. 2015.

[3] L. Feng, Y. Ong, A. Tan, and I. W. Tsang, "Memes as building blocks: a case study on evolutionary optimization + transfer learning for routing problems," *Memetic Computing*, vol. 7, no. 3, pp. 159 – 180, Aug. 2015.

[4] J. Xiao, R. Wang, G. Teng, and Y. Hu, "A transfer learning based classifier emsemble model for customer credit scoring," in *7th International Joint Conference on Computational Sciences and Optimization (CSO'14)*, New York, United states, Jul. 2014, pp. 64–68.

[5] A. Acharya, E. R. Hruschka, J. Ghosh, and S. Acharyya, "An optimization framework for combining ensembles of classifiers and clusterers with applications to nontransductive semisupervised learning and transfer learning," *Acm Transactions on Knowledge Discovery from Data*, vol. 9, no. 1, pp. 1–35, Sep. 2014.

[6] C. Kandaswamy, L. M. Silva, L. A. Alexandre, and J. M. Santos, "Deep transfer learning ensemble for classification," in *13th International Work-Conference on Artificial Neural Networks*, Palma de Mallorca, Spain, Jun. 2015, pp. 335 – 348.

[7] H. S. Bhatt, R. Singh, M. Vatsa, and N. K. Ratha, "Improving cross-resolution face matching using ensemble-based co-transfer learning," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5654 – 5669, Dec. 2014.

[8] S. Mei, "Svm ensemble based transfer learning for large-scale membrane proteins discrimination," *Journal of Theoretical Biology*, vol. 340, pp. 105–110, Jan. 2014.

[9] Z. H. Zhou, Ed., *Ensemble methods: foundations and algorithms*. Florida, United States: Chapman & Hall/CRC, 2012.

[10] C. Seiffert, T. M. Khoshgoftaar, J. Hulse, and A. Napolitano, "Resampling or reweighting: A comparison of boosting implementations," in *Proceedings International Conference on Tools with Artificial Intelligence(ICTAI)*, Dayton, OH, United States, Nov. 2008, pp. 445 – 451.

[11] Y. Frenud and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of 13th International Conference on Machine Learning(ICML)*, Bari, Italy, Jan. 1996, pp. 148–156.

[12] Z. H. Zhou, *Ensemble learning*. Boston, MA: Springer US, 2009, pp. 270–273.

[13] A. Onan, S. Korukolu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Systems with Applications*, vol. 57, pp. 232 – 247, Sep. 2016.

[14] Z. Mahmood, T. Ali, and S. Khan, "Effects of pose and image resolution on automatic face recognition," *IET Biometrics*, vol. 5, no. 2, pp. 111 – 119, Jun. 2016.

[15] G. Schaefer, B. Krawczyk, M. E. Celebi, and H. Iyatomi, "An ensemble classification approach for melanoma diagnosis," *Memetic Computing*, vol. 6, no. 4, pp. 233 – 240, Oct. 2014.

[16] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th Annual International Conference on Machine Learning*, Corvallis, Oregon, United states, Jun. 2007, pp. 193 – 200.

[17] E. Eaton and M. DesJardins, "Set-based boosting for instance-level transfer," in *ICDM Workshops 2009 - IEEE International Conference on Data Mining*, Miami, FL, United States, Dec. 2009, pp. 422 – 428.

[18] T. Kamishima, M. Hamasaki, and S. Akaho, "Trbagg: A simple transfer learning method and its application to personalization in collaborative tagging," in *Proc. IEEE International Conference on Data Mining(ICDM)*, Miami, FL, United states, Dec. 2009, pp. 219 – 228.

[19] X. Wu, V. Kumar, J. R. Quinlan, and et al., "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, Jun. 2008.

[20] L. La, Q. Guo, Q. Cao, and Y. Wang, "Transfer learning with reasonable boosting strategy," *Neural Computing and Applications*, vol. 24, no. 3-4, pp. 807 – 816, Mar. 2014.

[21] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123 – 140, Aug. 1996.

[22] Z. Xu and S. Sun, "Multi-view transfer learning with adaboost," in *Proceedings International Conference on Tools with Artificial Intelligence(ICTAI)*, Boca Raton, FL, United states, Nov. 2011, pp. 399 – 402.

[23] J. Wu, Z. Cai, and S. Ao, "Hybrid dynamic k-nearest-neighbour and distance and attribute weighted method for classification," *International Journal of Computer Applications in Technology*, vol. 43, no. 4, pp. 378–384, Jun. 2012.

[24] J. Wu, Z. Cai, and X. Zhu, "Self-adaptive probability estimation for naive bayes classification," in *Proceedings of the International Joint Conference on Neural Networks(IJCNN)*, Dallas, TX, United states, Aug. 2013, pp. 1–8.

[25] J. Wu, Z. Cai, S. Zeng, and X. Zhu, "Artificial immune system for attribute weighted naive bayes classification," in *Proceedings of the International Joint Conference on Neural Networks(IJCNN)*, Dallas, TX, United states, Aug. 2013, pp. 1–8.

[26] J. Wu and Z. Cai, "A naive bayes probability estimation model based on self-adaptive differential evolution," *Journal of Intelligent Information Systems*, vol. 42, no. 3, pp. 671–694, Jun. 2014.

[27] X. Liu, G. Wang, Z. Cai, and H. Zhang, "A multiboosting based transfer learning algorithm," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 19, no. 3, pp. 381–387, May 2015.

[28] M. Kearns and Y. Mansour, "On the boosting ability of top-down decision tree learning algorithms," *Journal of Computer and System Sciences*, vol. 58, no. 1, pp. 109 – 128, Feb. 1999.

[29] D. Sheskin, Ed., *Hand book of Parametric and Nonparametric Statistical Procedures*. London,U. K.: Chapman & Hall/CRC, 2006.

TABLE II: Prediction Accuracy of TrResampling (TrRes) compared to TrAdaBoost (TrAdaB) and C4.5

Data Set		3%			5%			10%	
	TrRes	TrAdaB	C4.5	TrRes	TrAdaB	C4.5	TrRe	TrAdaB	C4.5
anneal	94.54	**96.35**	74.11	96.1	**96.47**	83.73	**97.57**	97	88.99
anneal.ORIG	**75.3**	71.61	69.42	**79.37**	75.21	71.35	82.26	**82.52**	74.16
autos	19.28	21.35	**36.91**	21.8	22.62	**29.21**	26.5	27	**41.63**
balance-scale	**56.76**	**56.76**	56.08	56.82	56.82	**58.92**	56.7	57.68	**59.36**
breast-cancer	68.62	70.08	**73.08**	64.33	**70.31**	69.12	67.4	**69.65**	69.64
breast-w	57.89	60.94	**75.81**	72.75	72.58	**77.94**	78.67	**80.41**	78.78
colic	53.58	54.4	**68.85**	65.59	72.52	**73.01**	76.44	76.44	**78.51**
colic.ORIG	65.7	**70.56**	65.51	68.92	**70.97**	64.64	71.53	**72.61**	64.39
credit-a	82.32	86.5	**93.02**	82.36	84.87	**93.03**	87.26	87.74	**92.99**
credit-g	**66.32**	64.01	66.19	**65.75**	64.35	63.97	67.43	**67.79**	65.78
diabetes	**70.09**	69.87	59.49	67.18	**68.64**	60.29	**66.46**	65.86	59.84
glass	**56.27**	56.12	35.66	54.68	**54.98**	39.64	53.71	**54.05**	40.79
heart-c	79.67	**81.93**	74.87	77.31	**81.08**	74.8	**82.2**	80.71	78.74
heart-h	80.62	83.93	**85.34**	80.88	83.74	**85.18**	80.36	83.09	**85.29**
heart-statlog	**73.21**	69.7	58.58	**74.69**	70.34	63.15	**75.11**	74.3	64.25
hepatitis	**87.05**	85.45	85.98	81.39	**86.12**	85.99	**86.91**	85.92	83.92
hypothyroid	88.38	88.13	**91.4**	89.03	89.11	**91.5**	89.48	89.41	**91.65**
ionosphere	74.69	**81.89**	56.25	73.92	**78.69**	56.48	79.77	**80.31**	56.42
iris	**67.97**	**67.97**	44.57	**70.8**	**70.8**	49.57	**81.1**	79.09	69.03
kr-vs-kp	95.74	**96.41**	79.24	95.12	**96.23**	89.36	95.72	**96.66**	91.9
letter	62.28	**62.37**	48.83	**67.22**	67	54.53	70.75	**71.23**	60.77
lymph	49.9	49	**75.57**	54.56	59.47	**74.44**	71.21	71.4	**73.77**
mushroom	**97.76**	**97.76**	96.09	**97.66**	**97.66**	97.04	98.88	**98.99**	98.82
primary-tumor	11.86	**20.89**	11.46	12.76	**20.34**	13.06	**26.72**	26.15	24.08
segment	**81.69**	78.76	29.78	**87.11**	82.36	69.68	**91.58**	**91.58**	82.42
sick	95.71	**96.04**	93.07	**97.27**	97.1	93.7	**96.37**	95.65	95.71
sonar	61.5	**61.94**	61.44	58.83	62.99	**63.66**	60.38	60.87	**69.32**
soybean	63.94	**71.89**	35.40	**76.4**	74.67	45.16	**84.94**	81.06	61.07
splice	91.34	**92.49**	77.8	**92.28**	92.02	79.34	92.68	**92.7**	82.88
vehicle	61.35	**68.75**	39.75	67.22	**70.2**	45.09	66.04	**68.31**	54.28
vote	97.79	**97.91**	83.46	97.69	**97.86**	88.93	**97.62**	97.5	93.88
vowel	**41.09**	38.37	18.33	**45.14**	41.53	22.35	48.27	**48.32**	31.81
waveform	**86.09**	83.61	77.36	**86.85**	85.34	77.45	**87.61**	86.62	79.95
Rank	2.14	**2.32**	1.53	2.02	**2.35**	1.64	2.09	**2.33**	1.58
p-value		0.004			0.014			0.007	
Data Set		30%			60%			90%	
	TrRes	TrAdaB	C4.5	TrRes	TrAdaB	C4.5	TrRes	TrAdaB	C4.5
anneal	98.53	**98.83**	97.03	99.54	**100**	98.3	99.17	**100**	98.75
anneal.ORIG	**88.1**	87.6	84.11	**90.44**	**90.44**	89.06	88.63	89.66	**91.92**
autos	57.36	**60.07**	49.54	**70.02**	69.46	58.46	**78.31**	77.44	69.22
balance-scale	63.2	**68.12**	65.49	69.46	**72.24**	65.48	70.83	**73.21**	61.64
breast-cancer	63.56	63.66	**72.53**	68.12	67.37	**71.46**	68.99	70.41	**73.32**
breast-w	**90.44**	**90.44**	86.3	**91.82**	91.66	88.63	**93.98**	**93.98**	89.54
colic	76.38	77.9	**79.61**	78.48	76.98	**78.66**	74.21	74.17	**76.71**
colic.ORIG	73.91	**74.64**	70.13	79.45	**80.14**	75.97	**80.74**	79.72	75.56
credit-a	89.28	89.63	**93.03**	90	89.77	**93.05**	91.37	91.06	**93.23**
credit-g	68.96	**69.48**	67.1	71.21	**72.65**	70.44	70.36	73.98	**74.69**
diabetes	67.04	**67.09**	64.96	66.64	65.75	**67.79**	**71.5**	**71.5**	68.66
glass	**53.57**	51.44	47.32	**56.74**	56.03	54.88	65.01	**68.01**	59.43
heart-c	**86.71**	86.53	84.88	90.9	**91.96**	85.73	89.36	**94.09**	86.34
heart-h	82.55	80.43	**84.48**	80.5	**83.61**	83.3	**87.74**	83.81	84.13
heart-statlog	77.57	**77.98**	72.16	**81.64**	**81.64**	78.91	81.15	81.87	**85.71**
hepatitis	86.82	83.72	**88.11**	86.57	**86.9**	86.89	**83.75**	82.5	**83.75**
hypothyroid	90.15	90.03	**92.01**	90.8	90.84	**92.17**	91.07	90.91	**92.09**
ionosphere	83.96	**85.2**	76.28	85.23	**86.23**	80.88	83.19	**85.11**	79.86
Rank	**2.23**	2.14	1.64	2.05	**2.36**	1.59	**2.21**	2.12	1.67
iris	**88.59**	88.21	84.66	93.01	**93.03**	87.29	**94.78**	**94.78**	86.27
kr-vs-kp	**97.64**	97.57	95.13	99.08	**99.1**	98.3	99.6	**99.7**	99.6
letter	**79.09**	79.06	70.80	**83.32**	83.3	75.62	**85.82**	85.58	79.16
lymph	**81.64**	80.06	77.25	86.52	**86.56**	82.22	**90.1**	86.17	83.67
mushroom	99.97	99.97	99.97	100	100	100	100	100	100
primary-tumor	**30.73**	30.34	34.28	33.37	**33.49**	38.22	33.37	33.49	**38.22**
segment	**93.28**	92.88	90.32	**95.22**	95.17	92.55	94.75	**95.38**	93.4
sick	**97.31**	97.28	96.9	97.59	**97.68**	97.66	98.33	98.33	**98.6**
sonar	**67.84**	62.64	67.22	67.48	**68.84**	66.64	**71.58**	71.47	60.94
soybean	**90.57**	90.53	84.41	**94.01**	**94.01**	91.62	**96.6**	96.21	91.27
splice	83.83	89.55	**89.61**	92.42	92.73	**93.67**	**93.7**	93.64	93.08
vehicle	73.59	**74.13**	66.65	78.59	**78.97**	72.14	**78.91**	78.15	73.33
vote	96.95	96.79	**97.48**	96.26	**96.79**	**96.79**	**97.37**	96.84	96.81
vowel	68.72	69.71	52.61	82.31	82.15	67.30	87.44	87.84	72.12
waveform	**88.03**	87.87	81.60	**89.21**	88.29	82.82	**89.74**	88.76	81.32
Rank	**2.23**	2.14	1.64	2.05	**2.36**	1.59	**2.21**	2.12	1.67
p-value		0.031			0.005			0.047	