Research Journal of Applied Sciences, Engineering and Technology 5(24): 5550-5555, 2013
DOI:10.19026/rjaset.5.4234
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: September 24, 2012 Accepted: November 12, 2012 Published: May 30, 2013

Research Article

An Empirical Study of Combining Boosting-BAN and Boosting-MultiTAN

[1]Xiaowei Sun and [2]Hongbo Zhou
[1]Software College, Shenyang Normal University,
[2]BMW Brilliance Automotive Ltd. Co, Shenyang 110034 China

Abstract: An ensemble consists of a set of independently trained classifiers whose predictions are combined when classifying novel instances. Previous research has shown that an ensemble as a whole is often more accurate than any of the single classifiers in the ensemble. Boosting-BAN classifier is considered stronger than Boosting-Multi TAN on noise-free data. However, there are strong empirical indications that Boosting-MultiTAN is much more robust than Boosting-BAN in noisy settings. For this reason, in this study we built an ensemble using a voting methodology of Boosting-BAN and Boosting-MultiTAN ensembles with 10 sub-classifiers in each one. We performed a comparison with Boosting-BAN and Boosting-MultiTAN ensembles with 25 sub-classifiers on standard benchmark datasets and the proposed technique was the most accurate.

Keywords: Bayesian network classifier, combination method, data mining, boosting

INTRODUCTION

The goal of ensemble learning methods is to construct a collection (an ensemble) of individual classifiers that are diverse and yet accurate. If this can be achieved, then highly accurate classification decisions can be obtained by voting the decisions of the individual classifiers in the ensemble. Many authors, just like Breiman (1996), Kohavi and Kunz (1997) and Bauer and Kohavi (1999), have demonstrated significant performance improvements through ensemble methods.

An accessible and informal reasoning, from statistical, computational and representational viewpoints, of why ensembles can improve results is provided by Dietterich (2001). The key for success of ensembles is whether the classifiers in a system are diverse enough from each other, or in other words, that the individual classifiers have a minimum of failures in common. If one classifier makes a mistake then the others should not be likely to make the same mistake.

Boosting, the machine-learning method that is the subject of this study, is based on the observation that finding many rough rules of thumb can be a lot easier than finding a single, highly accurate prediction rule. To apply the boosting approach, we start with a method or algorithm for finding the rough rules of thumb. The boosting algorithm calls this "weak" or "base" learning algorithm repeatedly, each time feeding it a different subset of the training examples (or, to be more precise, a different distribution or weighting over the training examples). Each time it is called, the base learning algorithm generates a new weak prediction rule and after many rounds, the boosting algorithm must combine these weak rules into a single prediction rule that, hopefully, will be much more accurate than any one of the weak rules.

The first provably effective boosting algorithms were presented by Freund and Schapire (1995). Boosting works by repeatedly running a given weak learning algorithm on various distributions over the training data and then combining the classifiers produced by the weak learner into a single composite classifier. The first provably effective boosting algorithms were presented by Schapire (1990). More recently, we described and analyzed AdaBoost and we argued that this new boosting algorithm has certain properties which make it more practical and easier to implement than its predecessors.

TAN and BAN are augmented Bayesian network classifiers provided by Friedman *et al*. (1999) and Cheng and Greiner (1999). They treat the classification node as the first node in the ordering. The order of other nodes is arbitrary; they simply use the order they appear in the dataset. Therefore, they only need to use the CLB1 algorithm, which has the time complexity of O (N^2) on the mutual information test (N is the number of attributes in the dataset) and linear on the number of cases. The efficiency is achieved by directly extending the Chow-Liu tree construction algorithm to a three-phase BN learning algorithm (Cheng *et al*., 1997): drafting, which is essentially the Chow-Liu algorithm, thickening, which adds edges to the draft and thinning, which verifies the necessity of each edge.

Corresponding Author: Xiaowei Sun, Software College, Shenyang Normal University, Shenyang 110034, China, Tel.: +86 024 86578320

Boosting-BAN classifier is considered stronger than Boosting-MultiTAN classifier on noise-free data; however, Boosting-MultiTAN is much more robust than Boosting-BAN in noisy settings (Xiaowei and Hongbo, 2011). For this reason, in this study, we built an ensemble combing Boosting-BAN and Boosting-MultiTAN version of the same learning algorithm using the sum voting methodology. We performed a comparison with Boosting-BAN and Boosting-MultiTAN ensembles on standard benchmark datasets and the proposed technique had the best accuracy in most cases.

ENSEMBLES OF CLASSIFIERS

Boosting-BAN algorithm: Boosting-BAN works by fitting a base learner to the training data using a vector or matrix of weights. These are then updated by increasing the relative weight assigned to examples that are misclassified at the current round. This forces the learner to focus on the examples that it finds harder to classify. After T iterations the output hypotheses are combined using a series of probabilistic estimates based on their training accuracy.

The Boosting-BAN algorithm may be characterized by the way in which the hypothesis weights w_i are selected and by the example weight update step.

Boosting-BAN (Dataset, T): Input: sequence of N example *Dataset* = $\{(x_1, y_1),\ldots, (x_N, y_N)\}$ with labels $y_i \in Y = \{1,\ldots,k\}$, integer T specifying number of iterations.

Initialize $w_i^{(1)} = 1/N$ for all i, TrainData-1 = Dataset
Do for t = 1, 2,…, T:

- Use TrainData-t and threshold ε call BAN, providing it with the distribution.
- Get back a hypothesis $BAN^{(t)}$: X→Y.
- Calculate the error of $BAN^{(t)}$: $e^{(t)} = \sum_{i=1}^{N} w_i^{(t)} I(y_i \neq BAN^{(t)}(x_i))$.
- If $e^{(t)} \geq 0.5$, then set T=t-1 and abort loop.
- Set $\mu^{(t)} = e^{(t)}/(1-e^{(t)})$.
- Updating distribution $w^{(t+1)}{}_i = w^{(t)}{}_i (\mu^{(t)})^s$, where $s=1-I(y_i \neq BAN^{(t)}(x_i))$.
- Normalize $w^{(t+1)}{}_i$ to sum to 1.

Output the final hypothesis:

$$H(x) = argmax_{y \in Y}(\sum_{t=1}^{T}(log(1/\mu^{(t)}))I(y=BAN^{(t)}(x)))$$

Boosting-Multi TAN algorithm: GTAN is proposed by Hongbo *et al*. (2004). GTAN used conditional mutual information as CI tests to measure the average information between two nodes when the statuses of some values are changed by the condition-set C. When $I(x_i, x_j|\{c\})$ is larger than a certain threshold value ε, we choose the edge to the BN structure to form TAN. Start-edge and ε are two important parameters In GTAN. Different start-edges can construct different TANs. GTAN classifier is unstable that can be combined with a quite strong learning algorithm by boosting.

The Boosting-MultiTAN algorithm may be characterized by the way in which the hypothesis weights w_i are selected and by the example weight update step.

Boosting-MultiTAN (Dataset, T): Input: sequence of N example *Dataset* = $\{(x_1, y_1),\ldots, (x_N, y_N)\}$ with labels $y_i \in Y = \{1,\ldots,k\}$, integer T specifying number of iterations.

Initialize $w^{(1)}{}_i = 1/N$ for all i, TrainData-1=Dataset
Start-edge = 1; t = 1; l = 1
While (($t \leq T$) and ($l \leq 2T$)):

- Use TrainData-t and start-edge call GTAN, providing it with the distribution
- Get back a hypothesis $TAN^{(t)}: X \rightarrow Y$
- Calculate the error of $TAN^{(t)}$: $e^{(t)} = \sum_{i=1}^{N} w_i^{(t)} I(y_i \neq TAN^{(t)}(x_i))$
- If $e^{(t)} \geq 0.5$, then set T=t-1 and abort loop
- Set $\mu^{(t)} = e^{(t)}/(1-e^{(t)})$
- Updating distribution $w^{(t+1)}{}_i = w^{(t)}{}_i (\mu^{(t)})^s$, where $s=1-I(y_i \neq TAN^{(t)}(x_i))$
- Normalize $w^{(t+1)}{}_i$ to sum to 1
- t = t+1, l = l+1, start-edge = start-edge+n/2T.
- End while

Output the final hypothesis:
$H(x) = argmax_{y \in Y}(\sum_{t=1}^{T}(log(1/\mu^{(t)}))I(y=TAN^{(t)}(x)))$.

PROPOSED METHODLOGY

Recently, several authors have proposed theories for the effectiveness of boosting based on bias plus variance decomposition of classification error. In this decomposition we can view the expected error of a learning algorithm on a particular target function and training set size as having three components:

- A bias term measuring how close the average classifier produced by the learning algorithm will be to the target function
- A variance term measuring how much each of the learning algorithm's guesses will vary with respect to each other (how often they disagree)

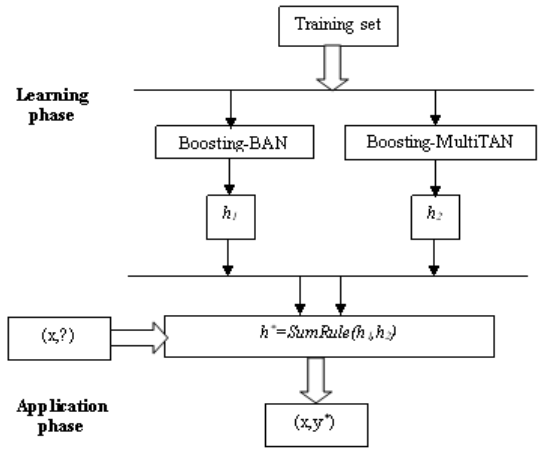

Fig. 1: The proposed ensemble

- A term measuring the minimum classification error associated with the Bayes optimal classifier for the target function (this term is sometimes referred to as the intrinsic target noise)

Boosting appears to reduce both bias and variance. After a base model is trained, misclassified training examples have their weights increased and correctly classified examples have their weights decreased for the purpose of training the next base model. Clearly, boosting attempts to correct the bias of the most recently constructed base model by focusing more attention on the examples that it misclassified. This ability to reduce bias enables boosting to work especially well with high-bias, low-variance base models.

For additional improvement of the prediction of a classifier, we suggest combing Boosting-BAN and Boosting-MultiTAN methodology with sum rule voting (Vote B&B). When the sum rule is used each sub-ensemble has to give a confidence value for each candidate. In our algorithm, voters express the degree of their preference using as confidence score the probabilities of sub-ensemble prediction. Next all confidence values are added for each candidate and the candidate with the highest sum wins the election. The proposed ensemble is schematically presented in Fig. 1, where h_i is the produced hypothesis of each sub-ensemble, x the instance for classification and y^* the final prediction of the proposed ensemble.

It has been observed that for Boosting-BAN and Boosting-MultiTAN, an increase in committee size (sub-classifiers) usually leads to a decrease in prediction error, but the relative impact of each successive addition to a committee is ever diminishing. Most of the effect of each technique is obtained by the first few committee members (Freund and Schapire, 1996). We used 10 sub-classifiers for each sub-ensemble for the proposed algorithm.

The proposed ensemble is effective owing to representational reason. The hypothesis space h may not contain the true function f (mapping each instance to its real class), but several good approximations. Then, by taking weighted combinations of these approximations, classifiers that lie outside of h may be represented.

It must be also mentioned that the proposed ensemble can be easily parallelized (one machine for each sub-ensemble). This parallel execution of the presented ensemble can reduce the training time in half.

COMPARISONS AND RESULTS

For the comparisons of our study, we used 20 well-known datasets mainly from many domains from the UCI repository (UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/ML Repository html.). These datasets were hand selected so as to come from

Table 1: Datasets used in the experiments

No	Dataset	Instances	Classes	Attributes	Missing values
1	Labor	57	2	16	√
2	Zoo	101	7	16	×
3	Promoters	106	2	57	×
4	Iris	150	3	4	×
5	Hepatitis	155	2	19	√
6	Sonar	208	2	60	×
7	Glass	214	7	9	×
8	Cleve	303	2	13	√
9	Ionosphere	351	2	34	×
10	House-votes-84	435	2	16	√
11	Votes1	435	2	15	√
12	Crx	690	2	15	√
13	Breast-cancer-w	699	2	9	√
14	Pima-indians-di	768	2	8	×
15	Anneal	898	6	6	√
16	German	1000	2	20	×
17	Hypothyroid	3163	2	25	√
18	Splice	3190	3	60	×
19	Kr-rs-kp	3196	2	36	×
20	Mushroom	8124	2	22	×

Table 2: Experimental results

No	Dataset	TAN	BAN	Boosting-multiTAN	Boosting-BAN	Vote B&B
1	Labor	95.8	95.1	95	96.8	96.4
2	Zoo	95.1	94.7	96.8	97.2	97.7
3	Promoters	95.2	95.5	95.3	95.3	95.3
4	Iris	96	95.7	96.7	97.1	96.9
5	Hepatitis	80.4	81.8	81.9	81.1	81.2
6	Sonar	83.3	83.5	87.5	87.1	87.5
7	Glass	64.5	66.3	66.8	68.5	68.9
8	Cleve	77.6	79.7	80.3	81.2	81.5
9	Ionosphere	92	92.4	92.0	93.0	92.4
10	House-votes-84	95.1	95.7	94.9	95.3	95.3
11	Votes1	94.2	95.6	94.7	95.1	95.8
12	Crx	85.5	85.5	85.6	86.5	85.8
13	Breast-cancer-w	96.6	96.7	96.1	96.8	96.6
14	Pima-Indians-di	73.6	73.9	74.7	75.5	75.8
15	Anneal	89.5	89.9	95.2	94.1	92.6
16	German	70.4	69.6	74.6	75.1	75.6
17	Hypothyroid	93.8	93.8	93.8	93.6	93.8
18	Splice	96.0	96.0	95.7	96.0	95.8
19	Kr-rs-kp	99.1	99.1	99.6	99.4	99.4
20	Mushroom	99.8	100	99.9	100	100

real-world problems and to vary in characteristics. Thus, we have used datasets from the domains of: pattern recognition (anneal, iris, mushroom, zoo), image recognition (ionosphere, sonar), computer games (kr-vs-kp).

Table 1 is a brief description of these datasets presenting the number of output classes, the type of the features and the number of examples. In order to calculate the classifiers' accuracy, the whole training set was divided into ten mutually exclusive and equal-sized subsets and for each subset the classifier was trained on the union of all of the other subsets. Then, cross validation was run 10 times for each algorithm and the median value of the 10-cross validations was calculated.

The time complexity of the proposed ensemble is less than both Boosting-BAN and Boosting-MultiTAN with 25 sub-classifiers. This happens because we use 10 sub-classifiers for each sub-ensemble (totally 20). The proposed ensemble also uses less time for training than both Multiboost and Décorare combining methods.

In our experiments, we set the number of rounds of boosting to be T = 100.

We compare the presented methodology with TAN, BAN, Boosting-BAN and Boosting-MultiTAN method. In the last raw of the Table 2 one can see the aggregated results.

The results of our experiments are shown in Table 2. The figures indicate test correct rate averaged over multiple runs of each algorithm.

The presented ensemble is significantly more accurate than single others in 8 out of the 20 datasets from Table 2, while it has significantly higher error rate in none dataset. BAN can only slightly increase the average accuracy of TAN without achieving significantly more accurate results. In addition, Boosting- BAN and Boosting-MultiTAN are

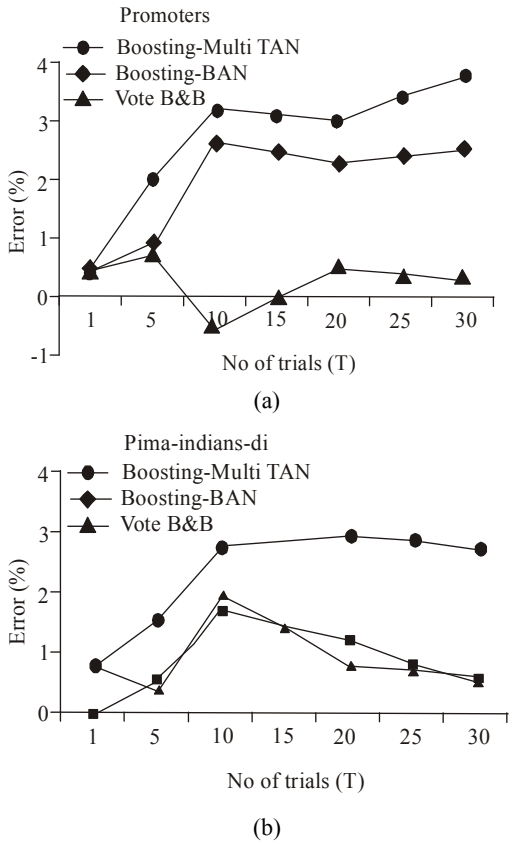

Fig. 2: Comparison of three classifiers on two datasets

significantly more accurate than single one in 6 and 3 out of the 20 datasets respectively, while they have significantly higher error rate in none dataset.

To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles. The proposed ensemble can achieve a reduction in error rate about 9% compared to simple TAN and BAN.

The differences are highlighted in Fig. 2, which compaires Boosting-BAN and Boosting-MultiTAN on two datasets, Pima-Indians-di and Promoters, as a function of the number of trials T. For T=1, Boosting-BAN is identical to Boosting-MultiTAN and both are almost always inferior to Vote B&B. As T increases, the performance of Boosting-BAN and Boosting-MultiTAN usually lead to a rapid degradation and then improve.

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples. One of the most active areas of research in supervised learning has been to study methods for constructing good ensembles of classifiers. The main discovery is that ensembles are often much more accurate than the individual classifiers that make them up. The main reason is that many learning algorithms apply local optimization techniques, which may get stuck in local optima. For instance, decision trees employ a greedy local optimization approach and neural networks apply gradient descent techniques to minimize an error function over the training data. As a consequence even if the learning algorithm can in principle find the best hypothesis, we actually may not be able to find it. Building an ensemble may achieve a better approximation, even if no assurance of this is given.

CONCLUSION

Boosting-BAN classifier is considered stronger than Boosting-MultiTAN on noise-free data, however, there are strong empirical indications that Boosting-MultiTAN is much more robust than Boosting-BAN in noisy settings. In this study we built an ensemble using a voting methodology of Boosting-BAN and Boosting-MultiTAN ensembles. It was proved after a number of comparisons with other ensembles, that the proposed methodology gives better accuracy in most cases. The proposed ensemble has been demonstrated to (in general) achieve lower error than either Boosting-BAN or Boosting-MultiTAN when applied to a base learning algorithm and learning tasks for which there is sufficient scope for both bias and variance reduction. The proposed ensemble can achieve an increase in classification accuracy of the order of 9% to 16% compared to the tested base classifiers.

Our approach answers to some extent such questions as generating uncorrelated classifiers and control the number of classifiers needed to improve accuracy in the ensemble of classifiers. While ensembles provide very accurate classifiers, too many classifiers in an ensemble may limit their practical application. To be feasible and competitive, it is important that the learning algorithms run in reasonable time. In our method, we limit the number of sub-classifiers to 10 in each sub-ensemble.

Finally, there are some open problems in ensemble of classifiers, such as how to understand and interpret the decision made by an ensemble of classifiers because an ensemble provides little insight into how it makes its decision. For learning tasks such as data mining applications where comprehensibility is crucial, voting methods normally result in incomprehensible classifier that can not be easily understood by end-users. These are the research topics we are currently working on and hope to report our findings in the near future.

ACKNOWLEDGMENT

Fund Support: The 6th Education Teaching Reform Project of Shenyang Normal University (JG2012-YB086).

REFERENCES

Bauer, E. and R. Kohavi, 1999. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. Mach. Learn., 36(1-2): 105-139.

Breiman, L., 1996. Bias, variance and arcing classifiers. Technical Report, 460, Department of Statistics, University of California, Berkeley, CA.

Cheng, J. and R. Greiner, 1999. Comparing Bayesian Network Classifiers. In: Kathryn Blackmond Laskey, Henri Prade (Eds.), Proceeding of the 15th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Francisco, pp: 101-108.

Cheng, J., D.A. Bell and W. Liu, 1997. An algorithm for Bayesian belief network construction from data. Proceeding of AI and STAT. Lauderdale, Florida, pp: 83-90.

Dietterich, T.G., 2001. Ensemble methods in machine learning. Kittler, J. and F. Roli (Eds.): Multiple classifier systems. Lect. Note. Comput. Sci., 1857: 1-15.

Freund, Y. and R.E. Schapire, 1995. A decision-theoretic generalization of on-line learning and an application to boosting. Unpublished manuscript available electronically (on our web pages, or by email request). An extended abstract. Second European Conference on Computational Learning Theory (EuroCOLT), pp: 23-37.

Freund, Y. and R.E. Schapire, 1996. Experiments with a new boosting algorithm. Proceedings of International Conference on Machine Learning, pp: 148-156.

Friedman, N., D. Geiger and M. Goldszmidt, 1999. Bayesian network classifiers. Mach. Learn., 29 (2-3): 131-163.

Hongbo, S., H. Houkuan and W. Zhihai, 2004. Boosting-based TAN combination classifier. J. Comput. Res. Dev., 41(2): 340-345.

Kohavi, R. and C. Kunz, 1997. Option decision trees with majority votes. Proceeding of 14th International Conference on Machine Learning, pp: 161-169.

Schapire, R.E., 1990. The strength of weak learns ability. Mach. Learn., 5(2): 197-227.

Xiaowei, S. and Z. Hongbo, 2011. An empirical comparison of two boosting algorithms on real data sets based on analysis of scientific materials. Adv. Intell. Soft Comput., 105: 324-327.