Intelligent Information Management, 2010, 2, 386-390
doi:10.4236/iim.2010.26047 Published Online June 2010 (http://www.SciRP.org/journal/iim)

Experiments with Two New Boosting Algorithms

Xiaowei Sun[1], Hongbo Zhou[2]
[1]*Software College, Shenyang Normal University, Shenyang, China*
[2]*Liaoning SG Automotive Group Co., Ltd., Shenyang, China*
E-mail: junyaomail@163.com, hongbo.zhou@hotmail.com
Received April 2, 2010; *revised May* 5, 2010; *accepted June* 8, 2010

Abstract

Boosting is an effective classifier combination method, which can improve classification performance of an unstable learning algorithm. But it dose not make much more improvement of a stable learning algorithm. In this paper, multiple TAN classifiers are combined by a combination method called Boosting-MultiTAN that is compared with the Boosting-BAN classifier which is boosting based on BAN combination. We describe experiments that carried out to assess how well the two algorithms perform on real learning problems. Finally, experimental results show that the Boosting-BAN has higher classification accuracy on most data sets, but Boosting-MultiTAN has good effect on others. These results argue that boosting algorithm deserve more attention in machine learning and data mining communities.

Keywords: Boosting, Combination Method, TAN, BAN, Bayesian Network Classifier

1. Introduction

Classification is a fundamental task in fault diagnosis, pattern recognition and forecasting. In general, a classifier is a function that chooses a class label (from a group of predefined labels) for instance described by a set of features (attributes). Learning accurate classifiers from pre-classified data is a very active research topic in machine learning and data mining. In the past two decades, many classifiers have been developed, such as decision trees based classifiers and neural network based classifiers.

Boosting [1-4] is a general method for improving the performance of any "weak" learning algorithm. In theory, boosting can be used to significantly reduce the error of any "weak" learning algorithm that consistently generates classifiers which need only be a little bit better than random guessing. Despite the potential benefits of boosting promised by the theoretical results, the true practical value of boosting can only be assessed by testing the method on "real" learning problems. In this paper, we present such experimental assessment of two new boosting algorithms.

The first provably effective boosting algorithms were presented by Schapire [5] and Freund [3]. 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. 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 [6] and BAN [7] are augmented Bayesian network classifiers provided by Friedman and Cheng J. In these papers, we treat the classification node as the first node in the ordering. The order of other nodes is arbitrary; we simply use the order they appear in the dataset. Therefore, we 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 [8] to a three-phase BN learning algorithm: *drafting*, which is essentially the Chow-Liu algorithm, *thickening*, which adds edges to the draft, and *thinning*, which verifies the necessity of each edge.

In this paper, multiple TAN classifiers are combined by a combination method called Boosting-MultiTAN that is compared with the Boosting-BAN classifier which is boosting based on BAN combination. Section 2 defines two classes of BNs, *i.e.*, Tree augmented Naive-Bayes (TANs) and BN augmented Naïve-Bayes (BANs), and describes methods for learning each. Section 3 proposes two new boosting algorithms, such as Boosting-

MultiTAN classifier and Boosting-BAN classifier. Section 4 presents and analyzes the experimental results. These results argue that boosting algorithm deserve more attention in machine learning and data mining communities.

2. Learning Bayesian Network Classifiers

Learning Bayesian network classifiers involves two steps: structure learning and parameter (conditional probability tables) learning. We will focus on structure learning methods for different Bayesian network classifiers in the subsections below.

2.1. Tree Augmented Naive-Bayes (TAN)

Letting $X = \{x_1, \ldots, x_n, c\}$ represent the node set (where c is the classification node) of the data. The algorithm for learning TAN classifier first learns a tree structure over V\{c}, using mutual information tests. It then adds a link from the classification node to each feature node in the manner as we construct a Naive-Bayes (*i.e*., the classify-cation node is a parent of all other nodes.) A simple TAN structure is shown in **Figure 1** (Note that features x_1, x_2, x_3, x_4 form a tree).

The learning procedure can be described as follows.

1) Take the training set and $X\backslash\{c\}$ as input.

2) Call the modified Chow-Liu algorithm. (The originnal algorithm is modified by replacing every mutual information test $I(x_i, x_j)$ with a conditional mutual information test $I(x_i, x_j|\{c\})$).

3) Add c as a parent of every x_i where $1 \le i \le n$.

4) Learn the parameters and output the TAN.

This algorithm, which is modified from the Chow-Liu algorithm, requires $O(N^2)$ numbers of conditional mutual information tests. This algorithm is essentially the first phase of the BAN-learning algorithm. TAN classifier is stable that can not be combined with a quite strong learning algorithm by boosting.

2.2. BN Augmented Naive-Bayes (BAN)

BAN classifier has been studied in several papers. The basic idea of this algorithm is just like the TAN learner of Subsection 2.1, but the unrestricted BN-learning algorithm instead of a tree-learning algorithm (see **Figure 2**).

Letting $X = \{x_1, \ldots, x_n, c\}$ represent the feature set (where c is the classification node) of the data, the learning procedure based on mutual information test can be described as follows.

1) Take the training set and $X\backslash\{c\}$ (along with the ordering) as input.

2) Call the modified CBL1 algorithm. (The original algorithm is modified in the following way: replace every mutual information test $I(x_i, x_j)$ with a conditional mutual information test $I(x_i, x_j|\{c\})$; replace every conditional mutual information test $I(x_i, x_j|Z)$ with $I(x_i, x_j|Z + \{c\})$, where $Z \subset X\backslash\{c\}$.

3) Add c as a parent of every x_i where $1 \le i \le n$.

4) Learn the parameters and output the BAN.

Like the TAN-learning algorithm, this algorithm dose not require additional mutual information tests, and so it requires $O(n^2N)$ (where n is the number of node attributes; N is the number of training examples) mutual information tests. The longest time spent in the algorithm is to calculate mutual information. In BAN structure, the second step in the three-phase is used ε to sort mutual information. The ε is a given small positive threshold, it is not fixed, and can be changed in many times. By setting different thresholds ε can construct many BAN classifiers. BAN classifier is unstable that can be combined with a quite strong learning algorithm by boosting.

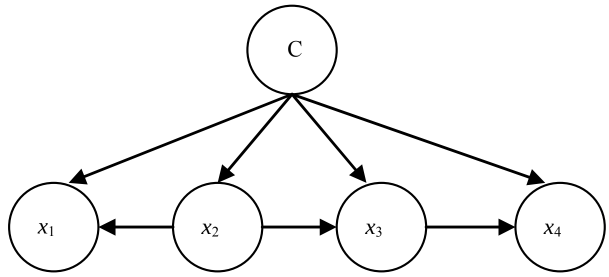

Figure 1. A simple TAN structure.

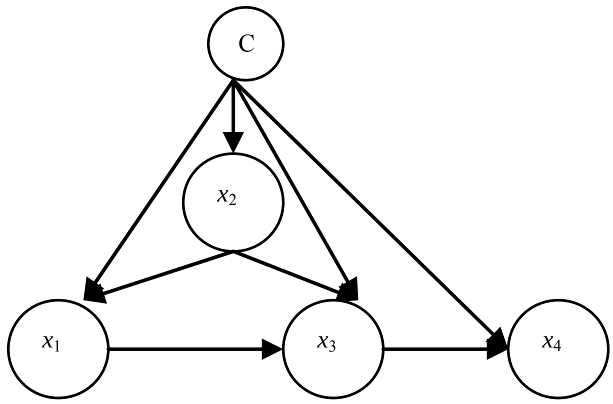

Figure 2. A simple BAN structure.

3. Two New Boosting Algorithms

3.1. Boosting-MultiTAN Algorithm

GTAN [9] is proposed by Hongbo Shi in 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 $w_i^{(1)} = 1/N$ for all *i, TrainData*-1 = *Dataset*

Start-edge = 1; $t = 1$; $l = 1$

While (($t \le T$) *and* ($l \le 2T$))

1) Use *TrainData-t* and *start-edge* call GTAN, providing it with the distribution.

2) Get back a hypothesis $TAN^{(t)} = X \to Y$.

3) Calculate the error of $TAN^{(t)}$:

$$e^{(t)} = \sum_{i=1}^{N} w_i^{(t)} I\left(y_i \ne TAN^{(t)}\left(x_i\right)\right).$$

If $e^{(t)} \ge 0.5$, then set $T = t - 1$ and abort loop.

4) Set $\mu^{(t)} = e^{(t)} \Big/ \left(1 - e^{(t)}\right)$.

5) Updating distribution $w_i^{(t+1)} = w_i^{(t)}\left(\mu^{(t)}\right)^s$, where $s = 1 - I\left(y_i \ne TAN^{(t)}\left(x_i\right)\right)$.

6) Normalize $w_i^{(t+1)}$, to sum to 1.

7) t = t + 1, l = l + 1, *start-edge* = *start-edge* + *n*/2*T*.

8) end While

Output the final hypothesis:

$$H\left(x\right) = \arg\max_{y \in Y}\left(\sum_{t=1}^{T}\left(\log\left(\frac{1}{\mu^{(t)}}\right)\right) * I\left(y = TAN^{(t)}\left(x\right)\right)\right)$$

3.2. 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 $t = 1, 2,\ldots, T$

1) Use *TrainData-t* and threshold ε call BAN, providing it with the distribution.

2) Get back a hypothesis $BAN^{(t)} = X \to Y$.

3) Calculate the error of $BAN^{(t)}$:

$$e^{(t)} = \sum_{i=1}^{N} w_i^{(t)} I\left(y_i \ne BAN^{(t)}\left(x_i\right)\right).$$

If $e^{(t)} \ge 0.5$, then set $T = t - 1$ and abort loop.

4) Set $\mu^{(t)} = e^{(t)} \Big/ \left(1 - e^{(t)}\right)$

5) Updating distribution $w_i^{(t+1)} = w_i^{(t)}\left(\mu^{(t)}\right)^s$, where $s = 1 - I\left(y_i \ne BAN^{(t)}\left(x_i\right)\right)$.

6) Normalize $w_i^{(t+1)}$, to sum to 1.

Output the final hypothesis:

$$H\left(x\right) = \arg\max_{y \in Y}\left(\sum_{t=1}^{T}\left(\log\left(\frac{1}{\mu^{(t)}}\right)\right) * I\left(y = BAN^{(t)}\left(x\right)\right)\right)$$

4. The Experimental Results

We conducted our experiments on a collection of machine learning datasets available from the UCI [10]. A summary of some of the properties of these datasets is given in **Table 1**. Some datasets are provided with a test set. For these, we reran each algorithm 20 times (since some of the algorithms are randomized), and averaged the results. For datasets with no provided test set, we used 10-fold cross validation, and averaged the results over 10 runs (for a total of 100 runs of each algorithm on each dataset).

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

The results of our experiments are shown in **Table 2**. The figures indicate test correct rate averaged over multiple runs of each algorithm. The bold in the table show that the classification is superior than another one obviously. From **Table 2** in the 20 datasets, Boosting-BAN did significantly and uniformly better than Boosting-MultiTAN.

On the data sets “Car”, “Iris” and “LED”, the Boosting-MultiTAN was inferior to the Boosting-BAN. The Boosting-BAN correct rate was better than the Boosting-MultiTAN correct rate in another 17 datasets. The reason is, in these cases the rate of attributes and classes are less than other Datasets. This reveals that the features in the three datasets are most dependent to each other. These weak dependencies can improve the prediction accuracy significantly, as we see from **Table 2**. These experiments also indicate that when the dataset is small and data loss, the boosting error rate is worse.

Table 1. Dataset used in the experiments.

No.	Dataset	Instances	Classes	Attributes	Missing values
1	Anneal	898	6	38	√
2	Audiology	226	24	69	√
3	Breast Cancer	699	2	9	×
4	Bupa	345	2	6	×
5	Car	1728	4	6	×
6	Cleveland	303	2	13	×
7	Crx	653	2	15	√
8	German	1000	2	20	×
9	House-votes-84	435	2	16	√
10	Hypothyroid	3163	2	25	√
11	Iris	150	3	4	×
12	Kr-rs-kp	3169	2	36	×
13	LED	1000	10	7	×
14	Mushroom	8124	2	22	√
15	Promoters	106	2	57	×
16	Segment	2310	7	19	×
17	Soybean Large	683	19	35	√
18	Tic-Tac-Toe	958	2	9	×
19	Wine	178	3	13	×
20	Zoology	101	7	16	×

Table 2. Experimental results.

No.	Dataset	Boosting-MultiTAN	Boosting-BAN
1	Anneal	98.3	**99.2**
2	Audiology	76.2	**78.8**
3	Breast Cancer	95.5	**95.9**
4	Bupa	58.5	**59.8**
5	Car	**87.1**	85.5
6	Cleveland	82.7	**84.6**
7	Crx	85.2	**86.5**
8	German	70.8	**74.6**
9	House-votes-84	94.9	**95.8**
10	Hypothyroid	99.1	**99.8**
11	Iris	**93.8**	90.5
12	Kr-rs-kp	93.3	**99.3**
13	LED	**73.9**	72.3
14	Mushroom	99.9	**100**
15	Promoters	89.3	**91.7**
16	Segment	94.3	**96.4**
17	Soybean Large	92.6	**93.7**
18	Tic-Tac-Toe	74.7	**79.2**
19	Wine	97.3	**98.5**
20	Zoology	96.8	**97.7**

5. Conclusions

GTAN and BAN classifiers are unstable, by setting different parameters, we can form a number of different TAN and BAN classifiers. In this paper, multiple TAN classifiers are combined by a combination method called Boosting-MultiTAN that is compared with the Boosting-BAN classifier which is boosting based on BAN combination. Finally, experimental results show that the Boosting-BAN has higher classification accuracy on most data sets.

When implementing Boosting classifiers, we were able to calculate the value of c directly given our prior knowledge. Of course, in a real situation we would be very unlikely to know the level of class noise in advance. It remains to be seen how difficult it would prove to estimate c in practice.

6. References

[1] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Computational Learning Theory*: *2nd Euro-*

pean Conference (*EuroCOLT*'95), Barcelona, 13-15 March 1995, pp. 23-37.

[2] R. E. Schapire, Y. Freund, Y. Bartlett, *et al*., "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," In: D. H. Fisher, Ed., *Proceedings of the* 14*th International Conference on Machine Learning*, Morgan Kaufmann Publishers, San Francisco, 1997, pp. 322-330.

[3] Y. Freund, "Boosting a Weak Learning Algorithm by Majority," *Information and Computation*, Vol. 121, No. 2, 1995, pp. 256-285.

[4] J. R. Quinlan, "Bagging, Boosting, and C4.5," In: R. Ben-Eliyahu, Ed., *Proceedings of the* 13*th National Conference on Artificial Intelligence*, Portland, 4-8 August 1996, pp. 725-730.

[5] R. E. Schapire, "The Strength of Weak Learnability," *Machine Learning*, Vol. 5, No. 2, 1990, pp. 197-227.

[6] N. Friedman, D. Geiger and M. Goldszmidt, "Bayesian Network Classifiers," *Machine Learning*, Vol. 29, No. 2-3, 1997, pp. 131-163.

[7] J. Cheng and R. Greiner, "Comparing Bayesian Network Classifiers," In: K. B. Laskey and H. Prade, Ed., *Proceedings of the* 15*th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, 15 October 1999, pp. 101-108.

[8] J. Cheng, D. A. Bell and W. Liu, "An Algorithm for Bayesian Belief Network Construction from Data," In: *Proceedings of Conference on Artificial Intelligence and Statistics*, Lauderdale, January 1997, pp. 83-90.

[9] H. B. Shi, H. K. Huang and Z. H. Wang, "Boosting-Based TAN Combination Classifier," *Journal of Computer Research and Development*, Vol. 41, No. 2, 2004, pp. 340-345.

[10] UCI Machine Learning Repository. http://www.ics.uci.edu/~mlearn/ML.Repository.html

[11] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," In: L. Saitta, Ed., P*roceedings of the* 13*th International Conference on Machine Learning*, Bari, 3-6 July 1996, pp. 148-156.

[12] X. W. Sun, "Augmented BAN Classifier," *Proceedings of International Conference on Computational Intelligence and Software Engineering*, Wuhan, 11-13 December 2009.