SOLVING TWO-CLASS CLASSIFICATION PROBLEM USING ADABOOST

Lih-Heng Chan, Sh-Hussain Salleh

Center for Biomedical Engineering, Universiti Teknologi Malaysia

ABSTRACT This paper presents a learning algorithm based on AdaBoost for solving two-class classification problem. The concept of boosting is to combine several weak learners to form a highly accurate strong classifier. AdaBoost is fast and simple because it focuses on finding weak learning algorithms that only need to be better than random, instead of designing an algorithm that learns deliberately over the entire space. We evaluated algorithms using Breast Cancer Wisconsin dataset which consists of 699 patterns with 9 attributes. It aims at assisting medical practitioners in breast cancer diagnosis. Thus the class output is the diagnosis prediction which is either benign or malignant. For comparison, back propagation neural network (BPNN) is developed and implemented on the same database. Experimental results show that AdaBoost is able to outperform BPNN under same experimental condition.

1. INTRODUCTION

In machine learning, boosting is a general method that can be applied on any learning algorithm to improve its performance. The term "weak leaner" has always been mentioned throughout the evolution of boosting-based algorithms. Literally it refers to weak learning algorithms that perform just slightly better than random guess. Schapire R.E. (1990) showed that these so-called weak learners can be efficiently combined or "boosted" to build a strong accurate classifier. This boosting algorithm applies weak learning algorithms multiple times to instance space with different distribution, and finally construct a strong hypothesis from numerous weak hypotheses

Freund, Y. *et al.*, (1997) first introduced theoretically the adaptive boosting (AdaBoost) method which significantly reduces the error of any learning algorithm that consistently generates classifiers with the condition of that: "better than random guess". In AdaBoost algorithms, distribution over instance space of training set are adjusted adaptively to the errors of weak hypotheses. This helps to move the weak learner towards the "harder" part of classification space more efficiently.

Subsequently, Freund Y. *et al.* (1996) tested AdaBoost on 27 bench mark learning problems of UCI repository. Experimental results showed that, on relatively simple classifiers, improvement made by AdaBoost can be dramatic, which is far better than "bagging". They emphasized the two effects of boosting: 1) reduces the bias of the weak learner by forcing them to concentrate on different parts of instance space; 2) reduce variance of the weak learner by averaging several hypotheses generated from from different subsamples of training set.

Besides, AdaBoost is able to provide effective learning algorithms and strong bounds of generalization performance (Schapire *et al*, 1998). Using the simplest weak learner in the previous experiment, Schapire R.E. *et al.* (1999) proposed confident-rated AdaBoost algorithms for further improvement, including Real AdaBoost.

Since then, boosting-base methods have been widely employed in various applications for the purpose of classification or feature selection. One of the very successful demonstrations by Viola & Jones (2001) had contributed a significant impact on object detection in computer vision. Besides, boosting-based methods had also garnered attention of researchers to seek further enhancement, such as Gentle AdaBoost (Friedman *et al.*, 2000), Modest AdaBoost (Vezhnevets A. & Vezhnevets V., 2005) and FloatBoost (S.Z. Li *et al.*, 2002). FloatBoost incorporates backtrack mechanism of Floating Search into boosting algorithms to disqualify poor weak learners base on their performance in error rate.

In this paper, we investigated the AdaBoost algorithms in terms of the learning and generalization ability on the Wisconsin Breast Cancer Diagnosis (WBDC) data set of the University of Wisconsin (W.H. Wolberg *et al.*, 1990). We used Real AdaBoost and Modest AdaBoost of GML AdaBoost Matlab Toolbox (Vezhenevets A., 2005) which are based on decision tree. We compared the results with back-propagation neural network (BPNN)(Section 2.2).

This paper is organized as follows: Chapter 2

describes the boosting algorithms and BPNN. Chapter 3 reports experimental conditions and results. Conclusion is made on chapter 4.

2. ALGORITHMS

2.1 ADABOOST

The basic concept of boosting is that, multiple weak learning algorithms are called sequentially on different distribution of training sets. These weak learners are later transformed into a strong classifier. Among all the weak learners, weak learners which achieve less error are granted more weights in forming the strong classifier. Weak learners which performs badly are weighted lightly, or even discarded if its error ≥ 50% of accuracy. In AdaBoost, during weak learners' selection, distributions over the instance space are adjusted each time of boosting round, regarding the errors returned by previous weak learners. The re-weighting process aims at emphasizing those data which are incorrectly classified by previous weak learner. More importantly, it combines the weak hypotheses by summing their probabilistic predictions.

Y. Freund *et al.* (1997) proposed the steps to implement the algorithms. First, we obtain a sequenced of N training examples in domain X, $(x_1 \ldots x_N)$ which are labeled as either "1" or "0" to represent positive and negative samples, $(y_1 \ldots y_N)$. Mathematically, we have $x_i \in X$, $y_i \in \{1,0\}$. Next, distribution over the training set is used to initialize the weight vector: $w_i^1 = D(i)$ for $i = 1,\ldots,N$. The following are the steps for training and selections of weak learners; with t is the specifying number of iterations:

For t = 1, 2,..., T

1. Set
$$p^t = \frac{w^t}{\sum_{i=1}^{N} w_i^t} \tag{1}$$
2. Call weak leaner, providing it with the distribution p^t; get back a hypothesis h_t: $X \rightarrow [1,0]$.
3. Calculate the error of h_t:
$$\varepsilon_t = \sum_{i=1}^{N} p_i^t \left| h_t(x_i) - y_i \right| \tag{2}$$
4. Calculate parameter β_t as a function of ε_t:
$$\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t} \tag{3}$$
5. Update the weights:
$$w_i^{t+1} = w_i^t \beta_t^{1-|h_t(x_i)-y_i|} \tag{4}$$

Output the final hypothesis

$$h_f(x) = \begin{cases} 1 & if \sum_{t=1}^{T} (\log \frac{1}{\beta_t}) h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \log \frac{1}{\beta_t} \\ 0 & otherwise \end{cases} \tag{5}$$

The weak learners generated always hold the error within one and zero, $\varepsilon_t \in [0, 1]$. The parameter β_t is used for updating the weight vector. It based on the update rule that can increase the probability of misclassified examples to be focused by next hypothesis.

Schapire R.E. *et al.* (1999) provided a generalized analysis of AdaBoost. With the same inputs in section 2.1, distribution is denoted as $D(i)$. Weight initialization is done by: $D_1(i) = 1/N$.

For t = 1, 2,..., T:

1. Train weak learner using distribution D_t.
2. Get weak hypothesis h_t: $X \rightarrow \Re$.
3. Choose $\alpha_t \in \Re$.
4. Update distribution:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \tag{6}$$

, where Z_t is a normalization factor which makes D_{t+1} a distribution. After T boosting round, the final hypothesis becomes:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right) \tag{7}$$

Proposing Real AdaBoost, R.E. Schapire *et al.* (1999) replace α_t with

$$\alpha_t = \frac{1}{2} \ln\left(\frac{W_{+1}}{W_{-1}}\right) \tag{8}$$

$$W_b = \sum_{i=u_i:b} D(i) \tag{9}$$

, for $b \in \{-1,0,1\}$ and $u_i = y_i h_t(x_i)$. This replacement is to minimizing Z_t as an approach of reducing training error.

Modest AdaBoost (Vezhnevets, V *et al.*, 2005) modify α_t to

$$\alpha_t = W_{+1}(1 - \overline{W_{-1}} - W_{-1}(1 - \overline{W_{-1}}) \tag{10}$$

, where

$$\overline{W_b} = \sum_{i=y_k h_k(x_k)=b} \overline{D(i)} \tag{11}$$

$$\overline{D_k}(i) = (1 - D_k(i)) \overline{Z_k} \tag{12}$$

2.1.1 Weak Classifier

As described in introduction, weak classifier can be any learning algorithm which can make prediction that is slightly better than random guess. Classification and Regression Tree (CART) (Breiman *et al.*, 1984) is one of the most famous classification methods employed in boosting methods. It is a tree graph, as shown in Fig. 1, where the nodes represent the hypotheses made during training and the branches are marked with "yes/no". These branches represent the prediction of the hypotheses in its upper node. The leaves which are represented with square box in the figure are the decision trees final classification output. The output decides which class is

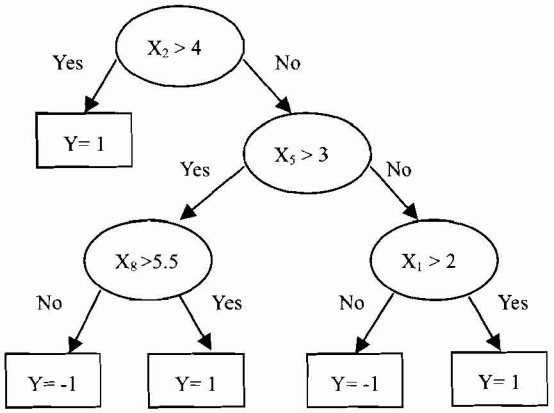


Fig. 1 Example of a CART

most probably the input belongs to, which is labeled as Y=1 or Y=0. To classify a single sample, its corresponding input values will be feed to the root of decision tree (X_2 in Fig 1), then further descend. The hypotheses act like threshold to determine whether to split further or make final decision.

One of the parameters of decision tree is the maximum number of node splitting. When there is only a single node splitting allowed, the decision tree is known as "stump decision" which was employed by Schapire *et al.* (1996) as their simplest weak learner. In our experiment, we used 1-3 node(s) splitting.

2.2 BACK-PROPAGATION NEURAL NETWORK

Neural networks-based methods have been widely used in classification for its good learning and generalization ability. A multi-layer neural network based on back-propagation is used in this work. During training, standard back-propagation algorithms is used to update weights of neural network based on the errors computed at each neuron's output (D.E. Rumelhart *et al.*, 1986). The errors are regarded as the gaps between target and network outputs.

In our BPNN architecture, we set 10 hidden layer neurons and 2 output layer neurons. Features are normalized within -1 and 1 by each vector respectively. Learning rate and momentum rate are set at 0.1 and 0.9. Table 1 shows the patterns we used to classify the features into different classes. The activation function used is hyperbolic tangent sigmoid transfer function which is given by

$$O = f(n) = \frac{2}{1+e^{-2n}} - 1 \qquad (13)$$

3. EXPERIMENT

This dataset contains 699 patterns of which 458 are Tablebenign samples and 241 are malignant samples, Each of these patterns consists of nine measurements taken from fine needle aspirates from a patient's breast (W.H. Wolberg *et al.*, 1990), yielding 9 attributes: 1) clump thickness, 2) uniformity of cell size, 3) uniformity of cell shape, 4) marginal adhesion, 5) single epithelial cell size, 6) bare nuclei, 7) bland chromatin, 8) normal nucleoli, and 9) mitoses. The measurements were graded one to ten (1-10) at the time of sample collection, with one being the closest to benign and ten the most anaplastic. The class output includes 2 classes, benign and malignant. We removed the sixteen instances with missing values from the dataset, turning it into 683 instances.

Table 1 Classification pattern of BPNN.

Class	NN output
Benign	01
Malignant	10

Table 2 Experiments setup for 5-fold cross validation.

Class	Training	Testing	Total
Benign	352	88	440
Malignant	188	47	235

3.1 SMALL TRAINING SAMPLES

In this experiment, training set consists of only 30 benign samples and 30 malignant samples, summing up to 60 training samples. It is small when compared with other experiments which used around 300-400 training samples. On the other hand, 283 samples (200 benign and 83 malignant samples) which do not stack with training samples were selected for testing. For more reliable comparison, we repeated our experiments for 12 times with different ways of partitioning the training and testing set.

Figure 2 shows accuracy rate of classification methods based on AdaBoost and BPNN. R1, R2 and R3 denote Real AdaBoost with 1, 2 and 3 nodes splitting. The same denotation applies to Modest AdaBoost with M1, M2 and M3. Experimental results show that, BPNN is outperformed by all AdaBoost methods. M1 and M2 are found to be the best performers which achieve an average accuracy of 95.76%, followed by R2 (95.64%). BPNN is outperformed by all AdaBoost methods, archiving 88.90%. On the other hand, regarding the results of R3 and M3, it is obvious that increasing number of splitting nodes does not really improve the recognition rate of AdaBoost.

3.2 5-FOLD CROSS VALIDATIONS

For 5-fold cross validations, we used 440 benign samples and 235 testing samples and separated them into five partitions each. At each validation, four partitions were used for training and one partition left for testing, as depicted in Table 2. Classification performances are shown in Figure 3. M3 achieves best performance with an average accuracy of 96.15%, followed by real R2 (96%), M1 and M2 (both 95.85%). Compared with small training samples experiments, BPNN had better performance in

5-fold cross validations. It achieves an accuracy of 93.63% when trained with larger samples.

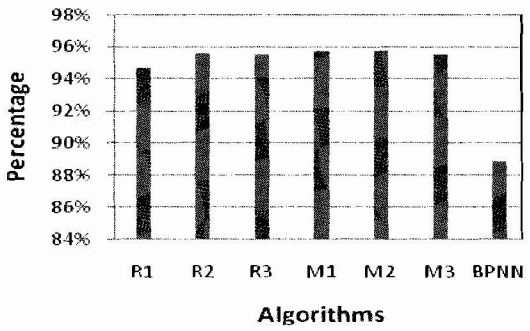


Fig. 2 Classification performance of Real AdaBoost, Modest AdaBoost, and BPNN for 283 testing samples

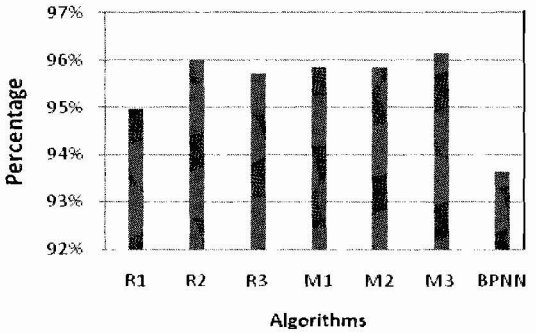


Fig. 3 Classification performance of Real AdaBoost, Modest AdaBoost, and BPNN for 5-fold cross validations

CONCLUSION

In this paper, we investigated and evaluated Real AdaBoost, Modest AdaBoost and BPNN using the original Wisconsin Breast Cancer Diagnosis dataset. Based on the feature provided by dataset, the algorithms are trained and subsequently used to classify testing data. In AdaBoost, decision tree is used as weak classifier, with 1-3 splitting nodes. From the experimental results, BPNN is outperformed by all AdaBoost methods. Among the best, Modest AdaBoost is slightly better than Real AdaBoost.

REFERENCES

D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning Internal Representations by Error Propagation, *Parallel Distributed Processing*, vol.1, MIT Press, MA, pp.318-362, 1986

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

Freund, Y., & Schapire, R.E. (1997), A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences, 55*(1), 119–139

Friedman, J., Hastie, T. and Tibshirani, R., Addictive logistic regression: A statistical view of boosting. Journal of the Annals of Statistics 38: 337-374, 2000.

Schapire R.E., The strength of weak learnability, *Machine Learning* 5, No. 2(1990), 197-227.

Schapire, R.E. and Singer, Y., Improved boosting algorithms using confidence-rated predictions. Journal of Machine Learning 37(3): 297 -336, 1999.

Schapire, R. E., Freund, Y., Bartlett, P. & Lee, W. S. (1998) Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Stat. Vol. 26, No. 5, 1651–1686.

S.Z. Li, Z.Q. Zhang, H.-Y. Shum, and H. Zhang, FloatBoost Learning for Classification, *Proc. Neural Information Processing Systems*, Dec. 2002.

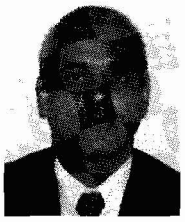
Vezhnevets, A., GML AdaBoost MATLAB Toolbox http://research.graphicon.ru/component/option,com_remository/Itemid,0/func,fileinfo/id,39/

Vezhnevets, A. and Vezhnevets, V., 'Modest AdaBoost' – teaching AdaBoost to generalize better. *GraphiCon* 2005.

Viola, P. and. Jones, M., Robust Real-time Object Detection. *International Conference on Computer Vision*, pp. 747, 2001

William H. Wolberg and O.L. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, *Proceedings of the National Academy of Sciences, U.S.A.*, Volume 87, December 1990, pp 9193-9196.

Lih-Heng Chan received the B.Eng. degree in electrical engineering from Universiti Teknologi Malaysia (UTM), Malaysia. He is now a candidate of M. Eng. in Electrical Engineering, UTM. His research interest includes machine learning, image pattern recognition.



Sh-Hussain Salleh received the B.Eng. degree in electronic engineering from University of Bridgeport, USA in 1988, M. Eng. degree in electrical engineering from Universiti Teknologi Malaysia (UTM), Malaysia in 1993, and Ph.D. degree from University of Edinburgh, U.K. in 1997. From 1995 to 2007, he was a lecturer with Faculty of Electrical Engineering, UTM where he was appointed as Head of Microelectronic and Computer Engineering Department from 1999 to 2004. He is currently Dean of Faculty of Biomedical and Health Science Engineering, and Director of Center for Biomedical Engineering, UTM. His research interests include speech recognition, speaker recognition, bio-medical signal processing and instrumentation. He is a Chairman of National Technical Committee on Biometrics, Malaysia