Research on Applied Technology with Online Boosting Algorithms

Xiaowei Sun[1, a*]

[1]Software College,Shenyang Normal University, Shenyang, 110034, China

[a]email: junyaomail@163.com

Keywords: boosting, ensemble learning,online learning, accuracy

Abstract. Boosting is an effective classifier combination method, which can improve classification performance of an unstable learning algorithm due to its theoretical performance guarantees and strong experimental results. However, the algorithm has been used mainly in batch mode, i.e., it requires the entire training set to be available at once and, in some cases, require random access to the data. Recently, Nikunj C.oza(2001) proved that some preliminary theoretical results and some empirical comparisons of the classification accuracies of online algorithms with their corresponding batch algorithms on many datasets. In this paper, we present online versions of some boosting methods that require only one pass through the training data. Specifically, we discuss how our online algorithms mirror the techniques that boosting use to generate multiple distinct base models. Our online algorithms are demonstrated to be more practical with larger datasets.

Introduction

Traditional supervised learning algorithms generate a single model such as a Naïve Bayes classifier or TAN[1] classifier or BAN[2] classifier and use it to classify examples. Ensemble learning algorithms combine the predictions of multiple base models, each of which is learned using a traditional algorithm. Boosting [3] is a well-known ensemble learning algorithm that has been shown to improve generalization performance compared to the individual base models. Theoretical analysis of Boosting's performance supports these results [4-7].

Nikunj C.oza(2001)[8] developed online versions of bagging and boosting. Online learning algorithms process each training example once "on arrival" without the need for storage and reprocessing, and maintain a current model that reflects all the training examples seen so far. Such algorithms run faster than typical batch algorithms in situations where data arrive continuously. They are also faster with large training sets for which the multiple passes through the training set required by most batch algorithms are prohibitively expensive.

Online Boosting Algorithm

Our online boosting algorithm is designed to be an online version of AdaBoost.M1[9]. AdaBoost generates a sequence of base Models h1,h2,⋯⋯,hM using weighted training sets (weighted by D1,D2,⋯⋯,DM) such that the training examples misclassified by model hm-1 are given half the total weight when generating model hm and the correctly classified examples are given the remaining half of the weight.

```
Online Boosting Algorithm:
Initial conditions : For all $m \in \{1,2,\cdots\cdots,M\}$, $\lambda$ msc=0, $\lambda$ msw=0.
Online Boosting(h, L0, (x, y))
    Set the example's "weighted" $\lambda$ =1.
    For each base model hm, ($m \in \{1,2,\cdots\cdots,M\}$) in h,
        Set k according to Poisson( $\lambda$ ).
        Do k times
            hm = L0(hm, (x, y)).
        If y = hm (x)
        then
            $\lambda$ msc← $\lambda$ msc+ $\lambda$
            ε m← λ msw/( λ msc+ λ msw)
              λ  ← λ (1/(2(1- ε m)))
      else
            λ msw← λ msw+ λ
            ε m← λ msw/( λ msc+ λ msw)
              λ  ← λ (1/(2 ε m))
To classify new examples:
      Return h(x)=arg maxc∈Y∑m:hm(x)=clog((1-εm)/εm)
```

Convergence of Batch and Online Boosting

Lemma 1, Lemma 2, Lemma 3 and Lemma 4 are standard results (Nikunj C.Oza,2001), so we state them without proof.

Lemma 1 As $M \to \infty$ and/or $N \to \infty, q_o^M \xrightarrow{P} q_b^M$.

Lemma 2 If $D_m^o \xrightarrow{P} D_m^b$, then $h_{m,N}^o(x) \xrightarrow{P} h_{m,N}^b(x)$ ·

Lemma 3 If $D_m^o \xrightarrow{P} D_m^b$ and then $h_{m,N}^o(x) \xrightarrow{P} h_{m,N}^b(x)$ then $e_{m,N}^o \xrightarrow{P} e_{m,N}^b$ ·

Lemma 4 If $X_1, X_2, \mathbf{K}$ and X are discrete random variables and $X_n \xrightarrow{P} X$, then $I(X_n = x) \xrightarrow{P} I(X = x)$ for all possible values x.

Theorem 1 Given the same training set, if $h_{m,N}^o(x)$ and $h_{m,N}^b(x)$ for all $m \in \{1,2,\mathbf{L}\, M\}$ are TAN/BAN classifiers,then $h^o(x) \xrightarrow{P} h^b(x)$.

Proof: We first prove the convergence of the base models and their errors by induction on m . For the base case, we show that $D_1^o \xrightarrow{P} D_1^b$. This lets us show that $h_{1,N}^o(x) \xrightarrow{P} h_{1,N}^b(x)$ and $e_{1,N}^o \xrightarrow{P} e_{1,N}^b$ as $N \to \infty$. By lemma 2 and Lemma 3 if $D_m^o \xrightarrow{P} D_m^b$, then $h_{m,N}^o(x) \xrightarrow{P} h_{m,N}^b(x)$ and $e_{m,N}^o \xrightarrow{P} e_{m,N}^b$. From these facts, we get $D_{m+1}^o \xrightarrow{P} D_{m+1}^b$, which let us show that $h_{m+1,N}^o(x) \xrightarrow{P} h_{m+1,N}^b(x)$ and $e_{m+1,N}^o \xrightarrow{P} e_{m+1,N}^b$ as $N \to \infty$. All of these facts are sufficient to show that the classification functions $h^o(x)$ and $h^b(x)$ converge.

We already have $D_1^o \xrightarrow{P} D_1^b$ by Lemma 1 . By Lemma 2, we have $h_{1,N}^o(x) \xrightarrow{P} h_{1,N}^b(x)$.That is, the first online TAN/BAN classifier converges to the first batch TAN/BAN classifier. By Lemma 3, $e_{1,N}^o \xrightarrow{P} e_{1,N}^b$. We have thus proven the base case.

Now we prove the inductive portion. That is, we assume that $D_m^o \xrightarrow{P} D_m^b$, which means that : $h_{m,N}^o(x) \xrightarrow{P} h_{m,N}^b(x)$ and $e_{m,N}^o \xrightarrow{P} e_{m,N}^b$. Given D_m^o and , we can $e_{m,N}^o$ calculate D_{m+1}^o . Given D_m^b and $e_{m,N}^b$, we can calculate D_{m+1}^b .

By the way the algorithms are set up, we have

$$D_{m+1}^o(n) = D_m^o(n)\left[\left(\frac{1}{2e_{m,n}^o}\right)^{|y_n - h_{m,n}^o(x_n)|}\left(\frac{1}{2(1-e_{m,n}^o)}\right)^{1-|y_n - h_{m,n}^o(x_n)|}\right], D_{m+1}^b(n) = D_m^b(n)\left[\left(\frac{1}{2e_{m,N}^b}\right)^{|y_n - h_{m,N}^b(x_n)|}\left(\frac{1}{2(1-e_{m,N}^b)}\right)^{1-|y_n - h_{m,N}^b(x_n)|}\right].$$

Because $e_{m,N}^o \xrightarrow{P} e_{m,N}^b, h_{m+1,N}^o(x) \xrightarrow{P} h_{m+1,N}^b(x)$, and our inductive assumption $D_m^o(n) \xrightarrow{P} D_m^b(n)$, and both $D_{m+1}^o(n)$ and $D_{m+1}^b(n)$ are continuous functions in these quantities that converge, we have that $D_{m+1}^o(n) \xrightarrow{P} D_{m+1}^b(n)$ as long as $e_{m,N}^o$ and $e_{m,N}^b$ are bounded away from 0 and 1. This is a reasonable assumption because, if the error is ever 0 or 1, then the algorithm stops and returns the ensemble constructed so far, so the algorithm would never even calculate the training set distribution for the next base model.

$D^o_{m+1}(n) \xrightarrow{P} D^b_{m+1}(n)$ implies that $h^o_{m+1}(x) \xrightarrow{P} h^b_{m+1}(x)$, which in turn implies that $e^o_{(m+1)N} \xrightarrow{P} e^b_{(m+1)N}$. So by induction, we have that $h^o_{m,N}(x) \xrightarrow{P} h^b_{m,N}(x)$ and $e^o_{m,N} \xrightarrow{P} e^b_{m,N}$ for all $m \in \{1,2,\ldots M\}$. By Lemma 4, we have $I(h^o_{m,N}(x)=c) \xrightarrow{P} I(h^b_{m,N}(x)=c)$ for all $c \in Y$. This means that $\sum_{m\in\{1,2,\ldots M\}} I(h^o_{m,N}(x)=c)\log\frac{1-e^o_{m,N}}{e^o_{m,N}} \xrightarrow{P} \sum_{m\in\{1,2,\ldots M\}} I(h^b_{m,N}(x)=c)\log\frac{1-e^b_{m,N}}{e^b_{m,N}}$. Therefore, the order of the terms $\sum_{m\in\{1,2,\ldots M\}} I(h^o_{m,N}(x)=c)\log\frac{1-e^o_{m,N}}{e^o_{m,N}}$ and $\sum_{m\in\{1,2,\ldots M\}} I(h^b_{m,N}(x)=c)\log\frac{1-e^b_{m,N}}{e^b_{m,N}}$ for each class c is preserved. The final functions $h^o(x)$ and $h^b(x)$ converge.

Comparison and results

In this section, we discuss results on several datasets, whose names and numbers of training examples, test examples, classes, attributes and missing values are given in Table 1. The Census Income dataset comes with fixed training and test sets, which we use in our experiments. For the remaining datasets, we used 5-fold cross-validation. We tested with some small datasets to show that the online algorithms can often achieve performance comparable to batch algorithms even when given a small number of data points. Of course, our results with larger datasets are more important. All but three of the datasets are from the UCI KDD repository [10]. The remaining three are synthetic datasets that were chosen because the performance of a single Naïve Bayes classifier varies significantly across these three datasets. These datasets allow us to compare the performances of the online and batch ensemble algorithms on datasets of varying difficulty.

Table 1. Datasets used in the experiments

No.	Dataset	Training Set	Test Set	Classes	Attributes	Missing values
1	Promoters	84	22	2	57	×
2	Breast-cancer-w	559	140	2	10	√
3	German	800	200	2	20	×
4	Car Evaluation	1382	346	4	6	×
5	Mushroom	6499	1625	2	22	×
6	Synthetic-1	80000	20000	2	20	×
7	Synthetic-2	80000	20000	2	20	×
8	Synthetic-3	80000	20000	2	20	×
9	Census Income	199523	99762	2	40	√
10	Forest Covertype	464809	116203	7	54	√

Table 2 gives the results of running boosting with TANs. Entries in the online TAN and boosting column that are given in boldface/italics indicate that it significantly outperformed/underperformed relative to batch TANs. Entries in the online boosting column given in boldface/italics indicate times when it significantly outperformed/underperformed relative to the online TAN.With TAN classifiers (Table 2), online boosting performed significantly worse than batch boosting on most of the datasets. On Mushroom and Census Income datasets, they performed comparably.

Table 2. Experimental results with Boosting vs.online Boosting, TANs

No.	Dataset	TAN	Online TAN	Boosting	Online Boosting
1	Promoters	88.7	*80.2*	*86.4*	*65.5-*
2	Breast-cancer-w	95.1	*89.2*	95.2	*88.3-*
3	German	75.8	*69.9*	*73.6*	*66.6-*
4	Car Evaluation	94.4	*88.8*	**97.8**	*88.6-*
5	Mushroom	100	99.9	99.9	99.9
6	Synthetic-1	73.3	*68.9*	*72.2*	*67.3-*
7	Synthetic-2	88.3	*87.2*	88.1	*85.7-*
8	Synthetic-3	98.4	*98.1*	98.4	*96.2-*
9	Census Income	94.4	94.1	94.1	94
10	Forest Covertype	76.7	*68.7*	**76.9**	*63.3-*

Table 3 gives the results of running boosting with BANs. Entries in the online BAN and boosting column that are given in boldface/italics indicate that it significantly outperformed/underperformed relative to batch BANs. Entries in the online boosting column given in boldface/italics indicate times when it significantly outperformed/underperformed relative to the online BAN. With BAN classifiers (Table 3), online boosting performed significantly worse than batch boosting on most of the datasets. On Mushroom and Census Income datasets, they performed comparably.

Table 3. Experimental results with Boosting vs.online Boosting, BANs

No.	Dataset	BAN	Online BAN	Boosting	Online Boosting
1	Promoters	89.8	*80.4*	*86.4*	*62.5-*
2	Breast-cancer-w	96.2	*90.3*	**96.8**	*88.5-*
3	German	74.6	*70.6*	**75.1**	*68.2-*
4	Car Evaluation	94.2	*88.1*	**98.3**	*88.1-*
5	Mushroom	100	99.9	100	99.9
6	Synthetic-1	72.2	*65.4*	**72.3**	*63.4-*
7	Synthetic-2	85.6	*83.4*	**85.6**	*81.1-*
8	Synthetic-3	98.3	98.1	98.2	*95.8-*
9	Census Income	95.2	*94.8*	*94.9*	*94.3*
10	Forest Covertype	75.7	*69.7*	**77.8**	*65.3-*

Entries with a '-' after them indicate times when online boosting performed significantly worse than batch boosting. Clearly, the significant loss in using an online TAN/BAN instead of a batch TAN/BAN has rendered the online boosting algorithm significantly worse than batch boosting.

We can see from the tables and from the scatter-plots of batch and online boosting (Fig.1 and Fig.2) that online boosting performs worse than batch boosting. Both batch boosting and online boosting do not improve upon TAN/BAN as much as they do upon Naive Bayes—especially on the larger datasets.

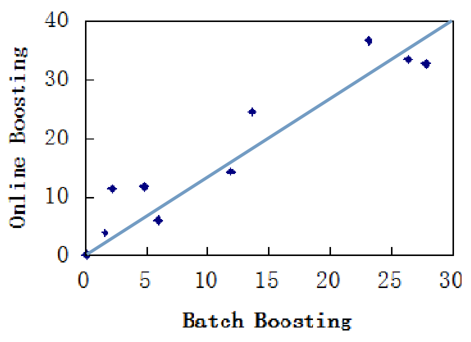

Fig 1. Test Error Rates:Batch Boosting vs. Online Boosting with TAN base

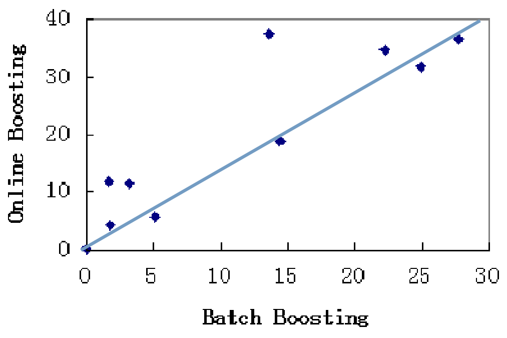

Fig 2. Test Error Rates:Batch Boosting vs. Online Boosting with BAN base

Conclusions

In this paper, we discussed online versions of boosting and gave both theoretical and experimental evidence that they can perform comparably to their batch counterparts in terms of accuracy while running much faster. We proved the convergence of the ensemble generated by the online boosting algorithm to that of batch boosting for TAN/BAN classifiers. The difference between the accuracies of the batch and online ensemble algorithms is largely a function of the differences between the accuracies of the batch and online base model learning algorithms. When lossless online base model learning algorithms are available (such as for Naïve Bayes classifiers), the performances of the ensemble algorithms tend to be comparable. In this paper, we experimented only with batch datasets, i.e., one is not concerned with concept drift. Online algorithms are useful for batch datasets that cannot be loaded into memory in their entirety. This paper provides a stepping stone to using ensemble learning algorithms on large datasets. We hope we can come up with new ideas to make ensemble learning algorithms more practical for modern data mining problems in the future.

References

[1] Cheng Jie, Greiner Russell, “Comparing Bayesian network classifiers”, In: Kathryn Blackmond Laskey, Henri Prade eds. Proc of the 15th Conf on Uncertainty in Artificial Intelligence. San Francisco: Morgan Kaufmann, pp.101-108, 1999.

[2] Friedman Nir, Geiger Dan, Goldszmidt Moises, “Bayesian network classifiers”, Machine Learning, 29 (2/3), pp. 131-163, 1999.

[3] Bauer Eric, Kohavi Ron, “An empirical comparison of voting classification algorithms: Bagging, boosting, and variants”, Machine Learning, 36 (1/2), pp. 105-139, 1999.

[4] Dietterich, Thomas, “Ensemble methods in machine learning”, In Kittler, J., Roli, F., eds.: Multiple Classifier Systems. Lecture Notes Computer Sciences, Vol. 1857, pp. 1-15,2001.

[5] Freund Yoav, Robert Schapire, “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 appeared in Computational Learning Theory: Second European Conf, EuroCOLT, pp.23-37,1995.

[6] Xiaowei Sun, Hongbo Zhou, “An Empirical Comparison of Two Boosting Algorithms on Real Data Sets based on Analysis of Scientific Materials”, Springer, Advances in Intelligent and Soft Computing, vol.105, pp.324-327, 2011.

[7] Hongbo Shi, Houkuan Huang, Zhihai Wang, “Boosting-Based TAN Combination Classifier”, Journal of Computer research and development, 41(2), pp. 340-345,2004.

[8] Nikunj C. Oza, “Online Ensemble Learning,” Ph.D. thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 2001.

[9] Yoav Freund and Robert Schapire, "A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting," Journal of Computer System Sciences, Vol. 55, No. 1, pp. 119-139, 1997.
[10] Stephen D. Bay, "The UCI KDD Archive UCI Machine Learning Repository." http://archive.ics.uci.edu/ml/.