A Boosting Algorithm for Regression*

A. Bertoni, P. Campadelli, M. Parodi

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
via Comelico 39/40, I-20135 Milano (Italy)

Abstract. A new boosting algorithm ADABOOST-RΔ for regression problems is presented and upper bound on the error is obtained. Experimental results to compare ADABOOST-RΔ and other learning algorithms are given.

1 Introduction

Boosting refers to the general problem of producing a very accurate prediction algorithm by appropriately combining rough and moderately inaccurate ones. It works by calling repeatedly a given "weak" learning algorithm on various distributions on the training set, and combining the hypotheses obtained with a linearly separable boolean function.

The boosting algorithm ADABOOST proposed by Freund and Schapire [1] and presented in Section 2 has been successfully applied to improve the performance of different learning algorithms used for classification problems both binary and multiclass [2]. The first extension for regression problems $f : X \to [0,1]$ is the algorithm ADABOOST-R [1]. In this paper we present ADABOOST-RΔ, a different and more general extension for problems $f : X \to [0,1]^m$. The main theoretical result is an upper bound on the error.

To analyse the performance of ADABOOST-RΔ we have done experiments using backpropagation as "weak" learning algorithm. Preliminary results show good convergence properties of ADABOOST-RΔ; notably it is able to lower both the mean and the maximum error on either the training and the test set. For regression problems $f : X \to [0,1]$ it works better than ADABOOST-R.

2 Preliminary definitions and results

Given a set X and $Y = \{0,1\}$, let $\mathcal{P}$ be a probability distribution on $X \times Y$. An $N-$sample is a sequence $\langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$ with $(x_k, y_k) \in X \times Y$; we call $Samp_N$ the set of all the $N-$samples and $Samp = \cup_N Samp_N$.

Given a class of functions $H \subseteq \{f \mid f : X \to \{0,1\}\}$, a learning algorithm $\mathcal{A}$ on H is a function $\mathcal{A} : Samp \to H$.

* Supported by grants CT 9305230.ST 74, CT 90.021.56.74

Let $S = \langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$ be a $N-$sample and let $h = \mathcal{A}_S$ be the hypothesis output by $\mathcal{A}$ on input S, the empirical error $\widehat{\epsilon}$ of $\mathcal{A}$ on S is

$$\widehat{\epsilon} = \frac{\#\{(x_k, y_k) \mid y_k \neq h(x_k)\}}{N}$$

while the generalization error is

$$\epsilon_g = \mathcal{P}\{y \neq h(x)\}.$$

Under weak conditions on H (that is the Vapnik-Chervonenkis dimension [3] of H is finite), choosing elements of $X \times Y$ randomly and independently according to $\mathcal{P}$, for sufficiently large samples, the empirical error is close to the generalization error with high probability [4]. For this reason a good learning algorithm should minimize the empirical error. Often this is a difficult task because of the large amount of computational resources required and computationally efficient algorithms are usually moderately accurate.

Boosting is a general method for improving the accuracy of a learning algorithm. In particular we refer to the algorithm ADABOOST presented in [1] and described below. It has in input a learning algorithm $\mathcal{A}$, a $N-$sample $S = \langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$, a distribution D on the elements of S, an integer T and gives in output a final hypothesis $h_f = HS(\sum_{k=1}^{T} w_k h_k - \lambda)$ with $h_k \in H \quad (k = 1, T)$; HS denotes the function $HS(x) =$ if $x \geq 0$ then 1 else 0. Even whether $\mathcal{A}$ is moderately inaccurate, for a sufficiently large T the error made by the final hypothesis h_f on the sample S can be made close to 0. Besides, the generalization ability of h_f is good since the Vapnik-Chervonenkis dimension of the family of the final hypothesis does not grow too much [1].

Algorithm ADABOOST

Input: a $N-$sample $S = \langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$, a distribution D on S, a learning algorithm $\mathcal{A}$, an integer T.
Initialize the weight vector $w_i^1 = D(i)$ for $i = 1, \ldots, N$
Do for $t = 1, 2, \ldots, T$
1. Set $\mathbf{p}^{(t)} = \frac{\mathbf{w}^{(t)}}{\sum_{i=1}^{N} w_i^{(t)}}$
2. Choose randomly with distribution $\mathbf{p}^{(t)}$ the sample $S^{(t)}$ from S; call the learning algorithm $\mathcal{A}$ and get the hypothesis $h_t = \mathcal{A}_{S^{(t)}}$
3. Calculate the error $\epsilon_t = \sum_{i=1}^{N} p_i^{(t)} \mid h_t(x_i) - y_i \mid$
4. Calculate $\beta_t = \frac{\epsilon_t}{(1-\epsilon_t)}$
5. Set the new weights vector to be: $w_i^{(t+1)} = w_i^{(t)} \beta_t^{1-|h^{(t)}(x_i)-y_i|}$
Output the hypothesis $h_f = HS\left(\sum_{k=1}^{T} (\log \frac{1}{\beta_t}) h_t(x) - 1/2 \sum_{k=1}^{T} (\log \frac{1}{\beta_t})\right)$

An upper bound to the error $\epsilon = \sum_{k=1}^{N} D(k) \cdot \mid h_f(x_k) - y_k) \mid$ is given by the following

Theorem 1 (Freund-Schapire). *Suppose the learning algorithm* $\mathcal{A}$*, when called by ADABOOST, generates hypotheses with errors* $\epsilon_1, \ldots, \epsilon_T$*. Then the error* $\epsilon = \sum_{k=1}^{N} D(k) \cdot \mid h_f(x_k) - y_k \mid$ *of the final hypothesis* h_f *output by ADABOOST is bounded by*

$$\epsilon \leq 2^T \prod_{t=1}^{T} \sqrt{\epsilon_t(1-\epsilon_t)}.$$

ADABOOST can be applied to classification problems with 2 classes. It has been generalized to multiclass problems ($Y = \{1, \ldots, K\}$) and to regression problems ($Y = [0,1]$) [1]. Roughly speaking, the main idea of the algorithm ADABOOST.**R** designed by Freund and Shapire for regression problems, is that of transforming the regression problem into a classification one using the total order relation $\leq$ on the real numbers. For example, every hypothesis $h : X \to [0,1]$ is transformed into the boolean function $\widehat{h} : X \times [0,1] \to \{0,1\}$ with

$$\widehat{h}(x,y) = \begin{cases} 1 & y \geq h(x) \\ 0 & \text{otherwise} \end{cases}$$

In the next paragraph we present a different way of transforming a regression problem for functions with values in $[0,1]^m$, into a classification problem. It develops an idea presented in [5] and it is based on the notion of norm in R^m.

3 The algorithm ADABOOST-RΔ

In this section we show a boosting algorithm ADABOOST-RΔ for regression problems. In this setting, Y is $[0,1]^m$ instead of $\{0,1\}$; as before, a sample S, chosen at random according to a probability distribution $\mathcal{P}$ on $X^m \times Y$, is given to a learning algorithm $\mathcal{A}$, that outputs a hypothesis $h : X \to [0,1]^m$.

Given a norm $\|\ \|$ on R^m, fixed $\Delta > 0$, we say that x and $\tilde{x}$ "Δ- agree" if $\|x - \tilde{x}\| \leq \Delta$. We consider as generalization error ϵ_g^Δ of a hypothesis h the probability that y and $h(x)$ does not "Δ- agree", that is $\epsilon_g^\Delta = \int HS(\|h_(x) - y\| - \Delta)dP$. Analogously, the empirical error ϵ^Δ on a sample $S = \langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$ is $\epsilon^\Delta = \sum (HS(\|h(x_k - y_k\| - \Delta))$.

Algorithm ADABOOST-RΔ

Input: a $N-$sample $S = \langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$, a distribution D on S, a learning algorithm $\mathcal{A}$, an integer T, a real numberΔ.

Initialize the weight vector $w_i^1 = D(i)$ for $i = 1, \ldots, N$

Do for $t = 1, 2, \ldots, T$

1. Set $\mathbf{p}^{(t)} = \frac{\mathbf{w}^{(t)}}{\sum_{i=1}^{N} w_i^{(t)}}$

2. Choose randomly with distribution $\mathbf{p}^{(t)}$ the sample $S^{(t)}$ from S; call the learning algorithm $\mathcal{A}$ and get the hypothesis $h_t = \mathcal{A}_{S^{(t)}}$

3. Calculate the error $\epsilon_t = \sum_{i=1}^{N} p_i^{(t)} HS(\|h_t(x_i) - y_i\| - \Delta)$
if $\epsilon_t > 1/2$ then $T = t - 1$ and abort loop

4. Calculate $\beta_t = \frac{\epsilon_t}{(1-\epsilon_t)}$
5. Set the new weights vector to be $w_i^{(t+1)} = w_i^{(t)}\beta_t^{1-HS(\|h_t(x_i)-y_i\|-\Delta)}$
Output the hypothesis $h_f = argmax_{y\in[0,1]^m} \sum \alpha_t HS(\Delta - \|h_t(x_i)-y_i\|)$ where $\alpha_t = \log \frac{1}{\beta}_t$

An upper bound to the error $\epsilon^{2\Delta} = \sum(HS(\|h_f(x_k - y_k\| - 2\Delta))$ is given by following

Theorem 2. *Suppose the learning algorithm $\mathcal{A}$, when called by ADABOOST-RΔ, generates hypotheses h_t with errors $\epsilon_1, \ldots, \epsilon_T < 1/2$ and let h_f be the final hypothesis output by ADABOOST-RΔ. Then*

$$\sum_{\|h_f(x_k)-y_k\|>2\Delta} D(k) \leq 2^T \prod_{t=1}^{T} \sqrt{\epsilon_t(1-\epsilon_t)}$$

.

Proof. (Outline) We transform the regression problem $X \to [0,1]^m$ into a classification problem $X \times [0,1]^m \to \{0,1\}$

- the sample $S = \langle (x_i, y_i) \mid i = 1, N\rangle$ is transformed into the sample $\widehat{S} = \langle ((x_i, y_i), 0) \mid i = 1, N\rangle$
- the distribution $D(i)$ is transformed into $\widehat{D}(i) = D(i)$ on the sample $\widehat{S}$
- the algorithm $\mathcal{A}$ with input S is transformed into the algorithm $\widehat{\mathcal{A}}$ with input $\widehat{S}$ with the rule: $\widehat{\mathcal{A}}_{\widehat{S}}(x_i, y_i) = HS(\|\mathcal{A}_S(x_i) - y_i\| - \Delta)$

Let $w^{(t)}$ and $\epsilon^{(t)}$ be respectively the weights vector and the error at step t of the algorithm ADABOOST-RΔ and let $\widehat{w}^{(t)}$ and $\widehat{\epsilon}^{(t)}$ $\epsilon^{(t)}$ be respectively the weight vector and the error at step t of the algorithm ADABOOST applied to the associated classification problem. By induction it can be proved that

$$\widehat{w}^{(t)} = w^{(t)}, \quad \widehat{\epsilon}^{(t)} = \epsilon^{(t)} \quad (1 \leq t \leq T)$$

Let $\widehat{h}_f(x,y)$ be the final hypothesis given by ADABOOST and $h_f(x)$ be the final hypothesis given by ADABOOST-RΔ. If $\|h_f(x_i) - y_i\| \geq 2\Delta$ then $\widehat{h}_f(x_i, y_i) = 1$. Let us suppose, on the contrary, that $\widehat{h}_f(x_i, y_i) = 0$, then

$$\sum_t \alpha_t HS(\|h_t(x_i) - y_i\| - \Delta) < 1/2 \sum_t \alpha_t \qquad (1)$$

Let $I = \{t \mid \|h_t(x_i) - y_i\| < \Delta\}$, the inequality (1) becomes $\sum_{t \notin I} \alpha_t < 1/2 \sum_t \alpha_t$ and the following relation hold

$$\sum_{t\in I} \alpha_t > \sum_{t \notin I} \alpha_t \qquad (2)$$

Since $h_f = argmax_{y\in[0,1]^m} \sum \alpha_t HS(\Delta - \|h_t(x_i) - y_i\|)$ then

$$\sum_t \alpha_t HS(\Delta - ||h_t(x_i) - h_f(x_i)||) \geq \sum_t \alpha_t HS(\Delta - ||h_t(x_i) - y_i||) \quad (3)$$

From (2) and (3) follows

$$\sum_{t\in I} \alpha_t HS(\Delta - ||h_t(x_i) - h_f(x_i)||) > \sum_{t\notin I} \alpha_t (1 - HS(\Delta - ||h_t(x_i) - h_f(x_i)||) \geq 0 \quad (4)$$

From the inequality (4) one can assert that there exists $\tilde{t} \in I$ such that $HS(\Delta - ||h_{\tilde{t}}(x_i) - h_f(x_i)||) = 1$; for such $\tilde{t}$ it holds that $||h_f(x_i) - h_{\tilde{t}}(x_i)|| < \Delta$ and $||h_{\tilde{t}}(x_i) - y_i|| < \Delta$. Hence by the triangular inequality we obtain: $||h_f(x_i) - y_i|| < 2\Delta$ but this is against the hypothesis.

Since $||\widehat{h}_f(x_i) - y_i)|| > 2\Delta$ implies $\widehat{h}_f(x_i, y_i) = 1$, using Theorem 1 we conclude

$$\sum_{||\widehat{h}_f(x_i) - y_i)|| > 2\Delta} D(i) \leq \sum_{\widehat{h}_f(x_i, y_i) = 1} D(i) \leq 2^T \prod_{t=1}^{T} \sqrt{\epsilon_t(1-\epsilon_t)}$$

□

4 Experimental results

In this section we present some preliminary results to evaluate the performance of ADABOOST-RΔ in terms of learning accuracy and computational efficiency. The experiments have been done as follows:

- the functions to be learned are functions $f : R \rightarrow R$ or $g : R^2 \rightarrow R^2$
- the "weak algorithm" $\mathcal{A}$, given in input to ADABOOST-RΔ, is backpropagation on neural networks of fixed architecture
- the parameter Δ has been set at 1.5δ, where δ is the error on the training set made by backpropagation and preliminarly computed.

The experiments show that ADABOOST-RΔ exhibits better convergence properties than backpropagation: after the same number of epochs the accuracy on either the training and the test sets is higher. A qualitative example of this behaviour is shown in Figure 1. Figure 1a and 1b show respectively the interpolation of the function $f(x) = (\sin(10 \cdot x) + 2)/4 + \sin(50 \cdot (x + 0.5)^2)/15 + 0.1N(0, 0.1)$ made by backpropagation and by ADABOOST-RΔ.

Quantitative results for a function $g : R^2 \rightarrow R^2$, described by the the expression $g(x, y) = \sin(\frac{6}{5\cdot(x+0.5)}) \cdot \sin(\frac{6}{5\cdot(y+0.5)})/3 + 0.5$ with gaussian noise $0.2 \cdot N(0, 0.2)$ added, are given in Table 1 (left). Two different network architectures (a denotes the architecture 2-10-10-2, b denotes the architecture 2-15-15-2) have been trained for different numbers of epochs.

For functions $f : R \rightarrow R$ we have also compared the performance of ADABOOST-RΔ with the algorithm ADABOOST-R. Table 1 (right) shows the results relative to the function $f(x) = \sin(\frac{1}{(0.03\cdot x)})/5 + 0.2 \cdot N(0, 0.2)$

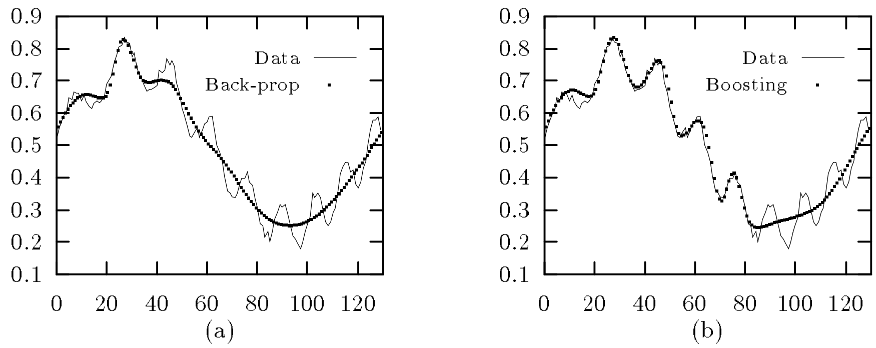

Fig. 1. Comparison between back-propagation and ADABOOST-RΔ.

str	data	epochs	T	mean err.	max err.
a	tr	1000	5	0.0428	0.1607
a	test	1000	5	0.0433	0.1545
a	tr	5000	1	0.0471	0.1739
a	test	5000	1	0.0475	0.1664
b	tr	1500	4	0.0417	0.1687
b	test	1500	4	0.0425	0.2101
b	tr	6000	1	0.0434	0.2054
b	test	6000	1	0.044 0	0.2404

alg.	data	epochs	T	mean err.	max err.
F.S.	tr	1000	6	0.021	0.111
F.S.	test	1000	6	0.020	0.128
Δ	tr	1000	5	0.017	0.058
Δ	test	1000	5	0.018	0.073
b.pr.	tr	6000	1	0.031	0.204
b.pr.	test	6000	1	0.036	0.228

Table 1.

References

[1] Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Internal Report of AT & T, September (1995)

[2] Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. Machine Learning: Proc. of Thirteenth Int. Conf. (1996) 148–156

[4] V.N. Vapnik (1982) Estimation of Dependences Based on Empirical Data. Springer-Verlag.

[3] V.N. Vapnik, A.Y. Chervonenkis (1971) On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2): 264-280.

[5] Freund, Y.: Boosting a weak learning algorithm by majority. Information and Computation **121 (2)** (1995) 256–285

This article was processed using the LaTeX macro package with LLNCS style