Robust Ensemble Learning for Data Mining[*]

G. Rätsch[1], B. Schölkopf[2], A.J. Smola[3], S. Mika[1],
T. Onoda[4], and K.-R. Müller[1]

[1] GMD FIRST, Kekuléstr. 7, D-12489 Berlin, Germany
[2] Microsoft Research, 1 Guildhall Street, Cambridge, UK
[3] Dep. of Engineering, ANU, Canberra ACT 0200, Australia
[4] CIRL CRIEPI, 2-11-1 Iwadokita, Komae-shi, Tokyo, 201-8511 Japan
{raetsch,mika,klaus}@first.gmd.de, bsc@microsoft.com,
Alex.Smola@anu.edu.au, onoda@criepi.denken.or.jp

Abstract. We propose a new boosting algorithm which similarly to ν-Support-Vector Classification allows for the possibility of a pre-specified fraction ν of points to lie in the margin area or even on the wrong side of the decision boundary. It gives a nicely interpretable way of controlling the trade-off between minimizing training error and capacity. Furthermore, it can act as a filter for finding and selecting informative patterns from a database.

1 Introduction

Boosting and related Ensemble learning methods have been recently used with great success in applications such as Optical Character Recognition [2, 3, 11]. The idea of a large (minimum) margin explains the good generalization performance of AdaBoost in the low noise regime. However, AdaBoost performs worse than other learning machines on noisy tasks [6, 7], such as the *iris* and the *breast cancer* benchmark data sets [5]. The present paper addresses the overfitting problem of AdaBoost in two ways. Primarily, it makes an *algorithmic* contribution to the problem of constructing regularized boosting algorithms. Secondly, it allows the user to roughly specify a hyper-parameter that controls the tradeoff between training error and capacity. This, in turn, is also appealing from a *theoretical* point of view, since it involves a parameter which controls a quantity that plays a crucial role in the generalization error bounds.

2 Boosting and the Linear Programming Solution

In this section, we briefly discuss the properties of the solution generated by standard AdaBoost and closely related Arc-GV[1], and discuss the relation to a linear programming (LP) solution over the class of base hypotheses G. Let $\{g_t(\mathbf{x}) : t = 1, \ldots, T\}$ be a sequence of hypotheses and $\boldsymbol{\alpha} = [\alpha_1 \ldots \alpha_T]$ their weights satisfying $\alpha_t \geq 0$. The hypotheses g_t are elements of a hypotheses class $G = \{g : \mathbf{x} \mapsto \{\pm 1\}\}$, which is defined by a base learning algorithm L.

The ensemble generates the label which is the weighted majority of the votes by $\operatorname{sign}(f(\mathbf{x}))$ where $f(\mathbf{x}) = \sum_t \frac{\alpha_t}{\|\boldsymbol{\alpha}\|_1} g_t(\mathbf{x})$. In order to express that f and therefore also the margin ρ depend on $\boldsymbol{\alpha}$ and for the ease of notation we define $\rho(\mathbf{z}, \boldsymbol{\alpha}) = yf(\mathbf{x})$, where $\mathbf{z} = (\mathbf{x}, y)$. Likewise we use the *normalized* margin

[*] This paper is a short version of [8].

$\rho(\boldsymbol{\alpha}) = \min_{1\le i\le m} \rho(\mathbf{z}_i, \boldsymbol{\alpha})$. The minimization objective function of AdaBoost can be expressed in terms of margins $\mathcal{G}(\boldsymbol{\alpha}) := \sum_{i=1}^m \exp(-\|\boldsymbol{\alpha}\|_1 \rho(\mathbf{z}_i, \boldsymbol{\alpha}))$. In every iteration AdaBoost tries to minimize this error by a stepwise maximization of the margin. It is believed (but not proven) that AdaBoost asymptotically approximates (up to scaling) the solution of the following linear programming problem over the complete hypothesis set G

$$\begin{array}{ll} \text{maximize} & \rho \\ \text{subject to } \rho(\mathbf{z}_i, \boldsymbol{\alpha}) \ge \rho & \text{for all } 1 \le i \le m \\ \qquad \alpha_t, \rho \ge 0 & \text{for all } 1 \le t \le |G| \\ \qquad \|\boldsymbol{\alpha}\|_1 = 1 \ . & \end{array} \tag{1}$$

Breiman [1] proposed a modification of AdaBoost, Arc-GV, making it possible to show the asymptotic convergence of $\rho(\boldsymbol{\alpha}^t)$ to the global solution: $\lim_{t\to\infty} \rho(\boldsymbol{\alpha}^t) = \rho^{\text{lp}}$, where ρ^{lp} is the maximum possible margin for a combined classifier from G.

3 ν-Arc

Let us consider the case where we are given a (finite) set $G = \{g : \mathbf{x} \mapsto [-1, 1]\}$ of T hypotheses. To find the coefficients $\boldsymbol{\alpha}$ for the combined hypothesis $f(\mathbf{x})$ we extend the LP-AdaBoost algorithm [4, 7] and solve the following linear optimization problem, similar in spirit to [10]:

$$\begin{array}{ll} \text{maximize} & \rho - \frac{1}{\nu m}\sum_{i=1}^m \xi_i \\ \text{subject to } \rho(\mathbf{z}_i, \boldsymbol{\alpha}) \ge \rho - \xi_i & \text{for all } 1 \le i \le m \\ \qquad \xi_i, \alpha_t, \rho \ge 0 & \text{for all } 1 \le t \le T \text{ and } 1 \le i \le m \\ \qquad \|\boldsymbol{\alpha}\|_1 = 1 \ . & \end{array} \tag{2}$$

This algorithm does not force all margins to be beyond zero and we get a *soft margin* classification with a regularization constant $\frac{1}{\nu m}$. Interestingly, it can be shown that ν is asymptotically proportional to the fraction of patterns in the margin area [8].

Suppose, we have a very large base hypothesis class G. Then it is very difficult to solve (2) as (1) directly. To this end, we propose an algorithm, ν-Arc, that can approximate the solution of (2). The optimal ρ for fixed margins $\rho(\mathbf{z}_i, \boldsymbol{\alpha})$ in (2) can be written as

$$\rho_\nu(\boldsymbol{\alpha}) := \operatorname*{argmax}_{\rho\in[0,1]} \left(\rho - \frac{1}{\nu m} \sum_{i=1}^m (\rho - \rho(\mathbf{z}_i, \boldsymbol{\alpha}))_+ \right) , \tag{3}$$

where $(\xi)_+ = \max(\xi, 0)$. Setting $\xi_i = (\rho_\nu(\boldsymbol{\alpha}) - \rho(\mathbf{z}_i, \boldsymbol{\alpha}))_+$ and subtracting $\frac{1}{\nu m}\sum_{i=1}^m \xi_i$ from the resulting inequality on both sides, yields (for all $1 \le i \le m$)

$$\rho(\mathbf{z}_i, \boldsymbol{\alpha}) + \xi_i \ge \rho_\nu(\boldsymbol{\alpha}) \tag{4}$$

$$\rho(\mathbf{z}_i, \boldsymbol{\alpha}) + \xi_i - \frac{1}{\nu m}\sum_{i=1}^m \xi_i \ge \rho_\nu(\boldsymbol{\alpha}) - \frac{1}{\nu m}\sum_{i=1}^m \xi_i \ . \tag{5}$$

In particular we have to get rid of the slack variables ξ_i again by absorbing them into quantities similar to $\rho(\mathbf{z}_i, \boldsymbol{\alpha})$ and $\rho(\boldsymbol{\alpha})$. This works as follows: on the right hand side of (5) we have the objective function (cf. (2)) and on the left hand side a term that depends nonlinearly on $\boldsymbol{\alpha}$. Defining

$$\tilde{\rho}_\nu(\boldsymbol{\alpha}) = \rho_\nu(\boldsymbol{\alpha}) - \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i, \quad \tilde{\rho}_\nu(\mathbf{z}_i, \boldsymbol{\alpha}) = \rho(\mathbf{z}_i, \boldsymbol{\alpha}) + \xi_i - \frac{1}{\nu m}\sum_{i=1}^{m}\xi_i, \qquad (6)$$

which we substitute for $\rho(\boldsymbol{\alpha})$ and $\rho(\mathbf{z}, \boldsymbol{\alpha})$ in (1), respectively, we obtain a new optimization problem. Note that $\tilde{\rho}_\nu(\boldsymbol{\alpha})$ and $\tilde{\rho}_\nu(\mathbf{z}_i, \boldsymbol{\alpha})$ play the role of a *corrected* or *virtual* margin. We obtain a non-linear min-max problem in terms of $\tilde{\rho}$

$$\begin{array}{ll} \text{maximize} & \tilde{\rho}(\boldsymbol{\alpha}) \\ \text{subject to} & \tilde{\rho}(\mathbf{z}_i, \boldsymbol{\alpha}) \geq \tilde{\rho}(\boldsymbol{\alpha}) \text{ for all } 1 \leq i \leq m \\ & \alpha_t \geq 0 \text{ for all } 1 \leq t \leq T \\ & \|\boldsymbol{\alpha}\|_1 = 1 \; , \end{array} \qquad (7)$$

which we refer to as ν-*Arc.*

We can now state interesting properties for ν-Arc by using Theorem 5 of [9] that bounds the generalization error $R(f)$ for ensemble methods. In our case $R_\rho(f) \leq \nu$ by construction, thus we get the following simple reformulation of this bound:

$$R(f) \leq \nu + \sqrt{\frac{c}{m}\left(\frac{h\log^2(m/h)}{\rho_\nu^2} + \log\left(\frac{1}{\delta}\right)\right)} \; . \qquad (8)$$

The tradeoff in minimizing the right hand side between the first and the second term is controlled directly by an easy interpretable regularization parameter ν.

4 Experiments

We show a set of toy experiments to illustrate the general behavior of ν-Arc-GV. As base hypothesis class G we use RBF networks [7], and as data a two-class problem generated from several 2D Gauss blobs. We obtain the following results:

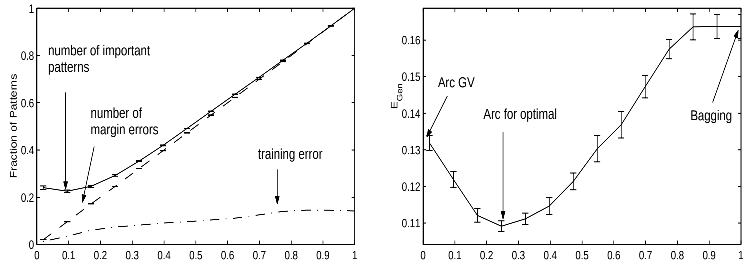

Fig. 1. Toy experiment: the left shows the average fraction of *important* patterns, the average fraction of margin errors and the average training error for different values of the regularization constant ν for ν-Arc. The bottom plots show the corresponding generalization error. The parameter ν allows us to reduce the test errors to values about 20% (relatively) lower than for the hard margin algorithm (for $\nu = 0$ we recover Arc-GV/AdaBoost and for $\nu = 1$ we get Bagging.)

- ν-Arc leads to approximately νm patterns that are effectively used in the training of the base learner: Figure 1 (left) shows the fraction of patterns that have high average weights during the learning process.
- ν-Arc leads to the fraction ν of margin errors (cf. dashed line in Figure 1) exactly.
- The (estimated) test error, averaged over 10 training sets, exhibits a rather flat minimum in ν (Figure 1 (right)).

5 Conclusion

We analyzed the AdaBoost algorithm and found that Arc-GV and AdaBoost are suitable for approximating the solution of non-linear min-max problems over huge hypothesis classes. We introduced a new regularization constant ν that controls the fraction of patterns inside the margin area. The new parameter is highly intuitive and has to be tuned only within a fixed interval $[0, 1]$.

We found empirically that the generalization performance in ν-Arc is robust against changes around the optimal choice of the regularization parameter ν. This finding makes model selection (e.g. via cross-validation) much easier.

As the patterns in the margin area correspond to interesting, difficult and informative patterns, future research will focus on using Boosting and Support Vector methods for data mining purposes.

References

1. L. Breiman. Prediction games and arcing algorithms. Technical Report 504, Statistics Department, University of California, December 1997.
2. H. Drucker, R. Schapire, and P. Simard. Boosting performance in neural networks. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 7:705 – 719, 1993.
3. Y. LeCun et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural Networks*, pages 261–276, 1995.
4. A. Grove and D. Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. In *Proc. of the 15th Nat. Conf. on AI*, pages 692–699, 1998.
5. C. J. Merz and P. M. Murphy. UCI repository of machine learning databases, 1998. [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. Irvine, CA.
6. J. R. Quinlan. Boosting first-order learning (invited lecture). *Lecture Notes in Computer Science*, 1160:143, 1996.
7. G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. Technical Report NC-TR-1998-021, NeuroColt, 1998. To appear in Machine Learning.
8. G. Rätsch, B. Schökopf, A. Smola, S. Mika, T. Onoda, and K.-R. Müller. Robust ensemble learning. In A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 207–219. MIT Press, Cambridge, MA, 1999.
9. R. Schapire, Y. Freund, P. L. Bartlett, and W. Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26(5):1651–1686, 1998.
10. B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1083 – 1121, 2000.
11. H. Schwenk and Y. Bengio. Training methods for adaptive boosting of neural networks. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Inf. Processing Systems*, volume 10. The MIT Press, 1998.