Boosting Algorithms for Maximizing the Soft Margin

Manfred Warmuth[1], Karen Glocer[1], Gunnar Rätsch[2]

1: UC Santa Cruz

2: Friedrich Miescher Laboratory of the Max Planck Society

manfred,kag@soe.ucsc.edu, gunnar.raetsch@tuebingen.mpg.de

1. Introduction

- Boosting algorithm when there is no consistent convex combination of base hypotheses
- In $\Theta(\frac{1}{\delta^2}\log\frac{N}{\nu})$ iterations produces a convex combination with soft margin within δ of the maximum

Boosting protocol:

- Set of examples $S = \langle(x_1, y_1), \ldots, (x_N, y_N)\rangle$
- Maintains distribution $\mathbf{d}$ on examples
- At iteration t:
  - Given current distribution d^{t-1}, oracle provides hypothesis h_t of edge $\gamma_t = \mathbf{d}^{t-1} \cdot \mathbf{u}^t \geq g$, where $u_i^t = y_i h_t(x_i)$
  - Guarantee $g > 0$ not known to algorithm
  - Update distribution $\mathbf{d}^{t-1}$ to $\mathbf{d}^t$

LPBoost computes $\mathbf{d}^t$ by solving:

Primal	Dual
$\min_{\mathbf{d},\gamma} \quad \gamma$ s.t. $\mathbf{d} \cdot \mathbf{u}^m \leq \gamma,\ 1{\leq}m{\leq}t,$ $\mathbf{d} \in \mathcal{P}^N,\ \mathbf{d} \leq \frac{1}{\nu}\mathbf{1}.$	$\max_{\mathbf{w},\rho} \quad \rho + \frac{1}{\nu}\sum_{n=1}^N \xi_n$ s.t. $\sum_{i=1}^t w_i y_i h_i + \xi_n \geq \rho,\ 1{\leq}n{\leq}N,$ $\mathbf{w} \in \mathcal{P}^M,\ \boldsymbol{\psi} \geq \mathbf{0}.$
minimize maximum edge	maximize minimum soft margin

Non-standard LPBoost formulation

- Totally corrective
- Capping probabilities in primal $\leftrightarrow$ soft margin in dual

2. LPBoost does not have $\Omega(\log N)$ iteration bounds.

- LPBoost (Schuurmans et al) works well in practice
- No bounds have been proved for it
- In our counter examples LPBoost takes $\Omega(N)$ iterations to achieve margin precision $\delta \approx 1$ for separable case.
  - Forces LPBoost to concentrate its distribution on single example
  - Holds regardless of LP optimization algorithm
  - Shows need for regularization

$n \setminus m$	1	2	3	4	5
1	$+1$	$-1+5\epsilon$	$-1+7\epsilon$	$-1+9\epsilon$	$-1+\epsilon$
2	$+1$	$-1+5\epsilon$	$-1+7\epsilon$	$-1+9\epsilon$	$-1+\epsilon$
3	$+1$	$-1+5\epsilon$	$-1+7\epsilon$	$-1+9\epsilon$	$-1+\epsilon$
4	$+1$	$-1+5\epsilon$	$-1+7\epsilon$	$-1+9\epsilon$	$-1+\epsilon$
5	$\mathbf{-1+2}\epsilon$	$+1$	$-1+7\epsilon$	$-1+9\epsilon$	$+1-\epsilon$
6	$-1+3\epsilon$	$\mathbf{-1+4}\epsilon$	$+1$	$-1+9\epsilon$	$+1-\epsilon$
7	$-1+3\epsilon$	$-1+5\epsilon$	$\mathbf{-1+6}\epsilon$	$+1$	$+1-\epsilon$
8	$-1+3\epsilon$	$-1+5\epsilon$	$-1+7\epsilon$	$\mathbf{-1+8}\epsilon$	$+1-\epsilon$

- The counter example suggests that a good algorithm should employ two tricks:
  - Cap the weight on any example
  - Spread the weight on the examples via a regularization such as the relative entropy

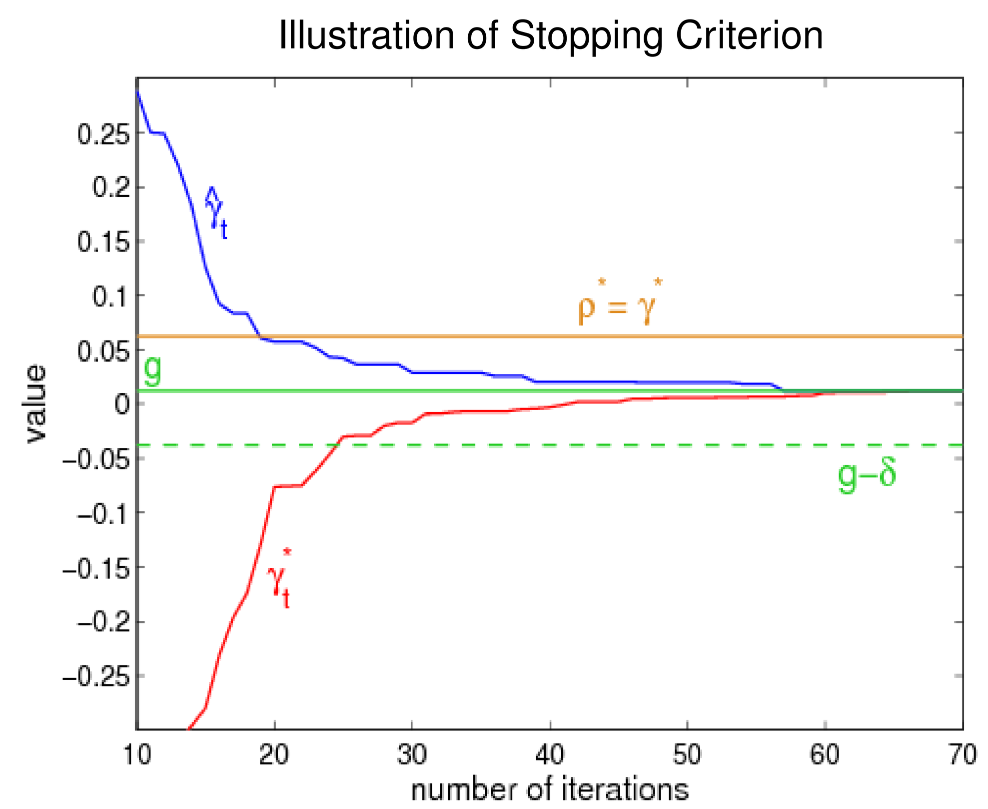
These two tricks used by the SoftBoost algorithm make it possible to obtain iteration bounds that grow logarithmic in N.

3. SoftBoost

- Designed for data that is not necessarily separable by convex combinations of base hypotheses
- Achieves robustness by *capping* the the weight on any example to be at most $\frac{1}{\nu}$
- Capping the weights on the examples prevents the algorithm from focusing excessively on a few examples that it can't hope to get right
- Produces a convex combination of hypotheses whose *soft margin* is within δ of the optimum
- SoftBoost terminates after at most $\lceil\frac{2}{\delta^2}\ln(N/\nu)\rceil$ iterations.
- The algorithm does not need to know the guarantee g on the base hypotheses

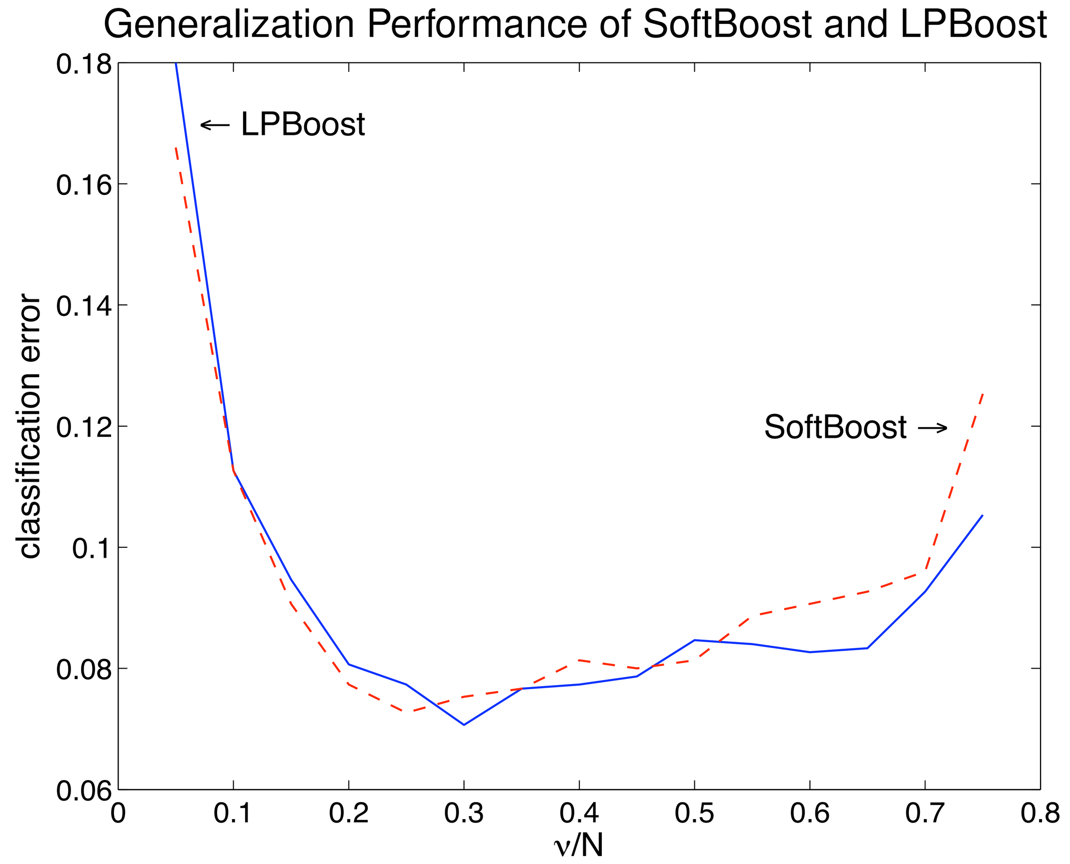
Algorithm 1: *SoftBoost*

1. **Input:** $S = \langle(x_1, y_1), \ldots, (x_N, y_N)\rangle$, desired accuracy δ, and capping parameter $\nu \in [1, N]$.
2. **Initialize:** d_n^0 to the uniform distribution
3. **Do for** $t = 1, \ldots$
   (a) Train classifier on $\mathbf{d}^{t-1}$ and $\{\mathbf{u}_1, \ldots, \mathbf{u}_{t-1}\}$ and obtain hypothesis h^t. Set $u_n^t = h^t(x_n)y_n$.
   (b) Calculate the edge γ_t of $h^t : \gamma_t = \mathbf{d}^t \cdot \mathbf{u}^t$
   (c) Set $\widehat{\gamma}_t = (\min_{m=1\ldots t}\gamma_m) - \delta$
   (d) Set $\gamma^* =$ solution to the primal linear programming problem.
   (e) Update
   $$\begin{aligned} \mathbf{d}^{t+1} &= \underset{\mathbf{d}}{\operatorname{argmin}} \sum_{n=1}^N d_n \log\frac{d_n}{d_n^{t-1}} \\ \text{s.t.} \quad & \mathbf{d}\cdot\mathbf{u}^m \leq \widehat{\gamma}_t - \delta, \text{ for } 1 \leq m \leq t \\ & \textstyle\sum_n d_n = 1,\ \mathbf{d} \leq \frac{1}{\nu}\mathbf{1}. \end{aligned}$$
   (f) **If** $\widehat{\gamma}_t - \gamma_t^* \leq \delta$ then $T = t - 1$ and break
4. **Output:** $f_{\mathbf{w}}(x) = \sum_{m=1}^T \mathbf{w}_m h^m(x)$, where the coefficients $\mathbf{w}_m$ maximize the soft margin over the hypothesis set $\{h^1, \ldots, h^t\}$ using the LP problem.



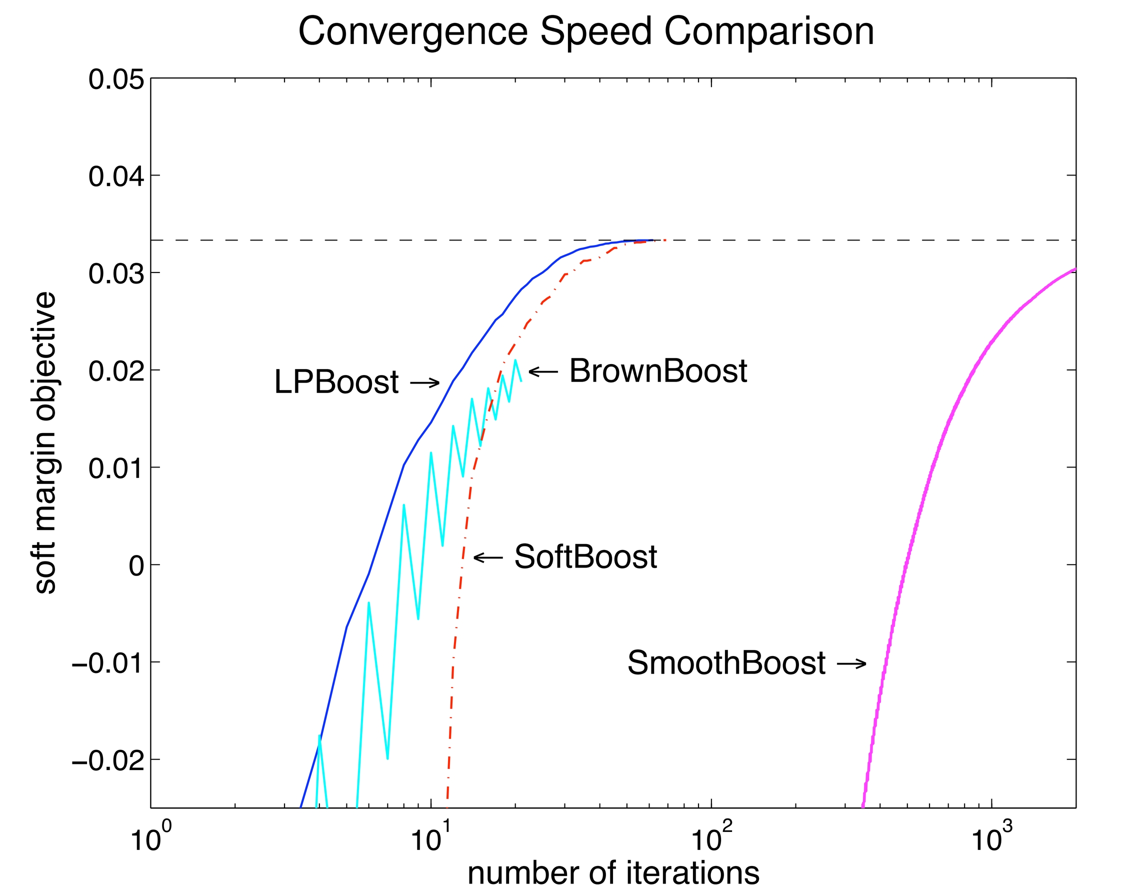
- $\widehat{\gamma}_t := (\min_{m=1\ldots t}\gamma_m) - \delta$ is nonincreasing and $\widehat{\gamma}_t \geq g$
- γ^* (solution to (1)) is non decreasing and $\gamma^* \leq g$
- Algorithm terminates when they are sufficiently close together

4. Experimental Results



- Generalization performance of SoftBoost (red) and LPBoost (blue) for different values of ν
- The data is a synthetic data set with with 10% label noise in the training set

- If ν is too small, the algorithm concentrates on a very few, presumably wrongly labeled examples and does not generalize well



- SoftBoost starts more slowly than LPBoost
- Both converge to within δ of guarantee g in approximately the same number of iterations
- BrownBoost, which was designed to deal with noisy data but is not a smooth boosting algorithm, does not maximize the soft margin
- SmoothBoost, the best of previously existing smooth boosting algorithms, converges much more slowly and does not achieve the optimal soft margin

	AdaBoost	LPBoost	SoftBoost	BrownBoost
Banana	13.3 ± 0.7	11.1 ± 0.6	11.1 ± 0.5	12.9 ± 0.7
B.Cancer	32.1 ± 3.8	27.8 ± 4.3	28.0 ± 4.5	30.2 ± 3.9
Diabetes	27.9 ± 1.5	24.4 ± 1.7	24.4 ± 1.7	27.2 ± 1.6
German	26.9 ± 1.9	24.6 ± 2.1	24.7 ± 2.1	24.8 ± 1.9
Heart	20.1 ± 2.7	18.4 ± 3.0	18.2 ± 2.7	20.0 ± 2.8
Ringnorm	1.9 ± 0.3*	1.9 ± 0.2	1.8 ± 0.2	1.9 ± 0.2
F.Solar	36.1 ± 1.5	35.7 ± 1.6	35.5 ± 1.4	36.1 ± 1.4
Thyroid	4.4 ± 1.9*	4.9 ± 1.9	4.9 ± 1.9	4.6 ± 2.10
Titanic	22.8 ± 1.0	22.8 ± 1.0	23.0 ± 0.8	22.8 ± 0.8
Waveform	10.5 ± 0.4	10.1 ± 0.5	9.8 ± 0.5	10.4 ± 0.4

- Generalization error estimates and standard deviations for ten UCI benchmark data sets
- SoftBoost and LPBoost outperform AdaBoost and BrownBoost on most data sets
- SoftBoost and LPBoost perform similarly

NIPS 2007, Neural Information Processing Systems, 3-6 December 2007, Vancouver, Canada