Training-Time Optimization of a Budgeted Booster

Yi Huang, Brian Powers, Lev Reyzin
{yhuang,bpower6,lreyzin}@math.uic.edu

UIC UNIVERSITY OF ILLINOIS AT CHICAGO

Problem Setting

Normal supervised learning with feature costs
Given:

- Training examples $S \subset X \times \{-1,+1\}$
- Feature cost function $c : [i \ldots n] \to \mathbb{R}^+$
- Test time budget $B > 0$

Challenge:
Predict on new examples under budget

Random Sampling

`AdaBoostRS` by Reyzin [1]

1. Train a classifier using AdaBoost
2. Randomly sample from ensemble predictors
3. Pay for each unpaid feature until budget is reached
4. Use weighted vote of sampled predictors

Budgeted Training

- Consider costs during training
- Cease training as soon as budget is reached
- Resulting classifier will obey budget
- We can easily modify `AdaBoost` for budgeted training

Cost Tradeoff Equations

Stop `AdaBoost` Early

- Choose h_t with maximum γ_t
- Does not prefer cheaper hypotheses

Modification 1 (Greedy)

- Goal: choose hypotheses to drive down training error bound

$$\prod_{t=1}^{T} \sqrt{1-\gamma_t^2}$$

- Last training round T is unknown
- Estimate T by assuming future rounds will have same cost as current
- Base learner is chosen to minimize

$$h_t = \underset{h \in \mathcal{H}}{\text{argmin}} \left((1-\gamma_t(h)^2)^{\frac{1}{c(h)}} \right) \qquad (1)$$

- Perhaps an aggressive assumption?

Modification 2 (Smoothed)

- Estimate T by assuming future rounds will incur average cost
- Base learner is chosen to minimize

$$h_t = \underset{h \in \mathcal{H}}{\text{argmin}} \left((1-\gamma_t(h)^2)^{\frac{1}{(B-B_t)+c(h)}} \right) \qquad (2)$$

- Milder assumption should smooth optimization

A Margin Bound Justification

Does opting for "quantity" of weak learners over "quality" lead to a predictor that won't generalize well? Margin bounds [2] suggest not.
The margin bound is

$$\Pr[yf(x) \le 0] \quad \le \quad \widehat{\Pr}[yf(x) \le \theta] + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right),$$

where $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$. The first term can be bounded [3]

$$\widehat{\Pr}[yf(x) \le \theta] \le e^{\theta \sum \alpha_i} \prod_{t=1}^{T} Z_t,$$

For small θ this tends to

$$\prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} \sqrt{1-\gamma_t^2}.$$

Algorithm: AdaBoost with Budgeted Training

`AdaBoostBT(S,B,C)`, where: $S \subset X \times \{-1,+1\}$, $B > 0$, $C : [i \ldots n] \to \mathbb{R}^+$

given: $(x_1, y_1), \ldots, (x_m, y_m) \in S$
initialize $D_1(i) = \frac{1}{m}$, $B_1 = B$
for $t = 1, \ldots, T$ **do**
  train base learner using distribution D_t, get $h_t \in \mathcal{H} : X \to \{-1,+1\}$
  if the total cost of the unpaid features of h_t exceeds B_t **then**
    set $T = t-1$ and **end for**
  else set B_{t+1} as B_t minus the total cost of the unpaid features of h_t, mark them as paid
  set $\alpha_t = \frac{1}{2}\ln\frac{1+\gamma_t}{1-\gamma_t}$, where $\gamma_t = \sum_i D_t(i) y_i h_t(x_i)$.
  update $D_{t+1}(i) = D_t(i)\exp(\alpha_t y_i h_t(x_i))/Z_t$, where Z_t is the normalization factor
end for
output the final classifier $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Experimental Results

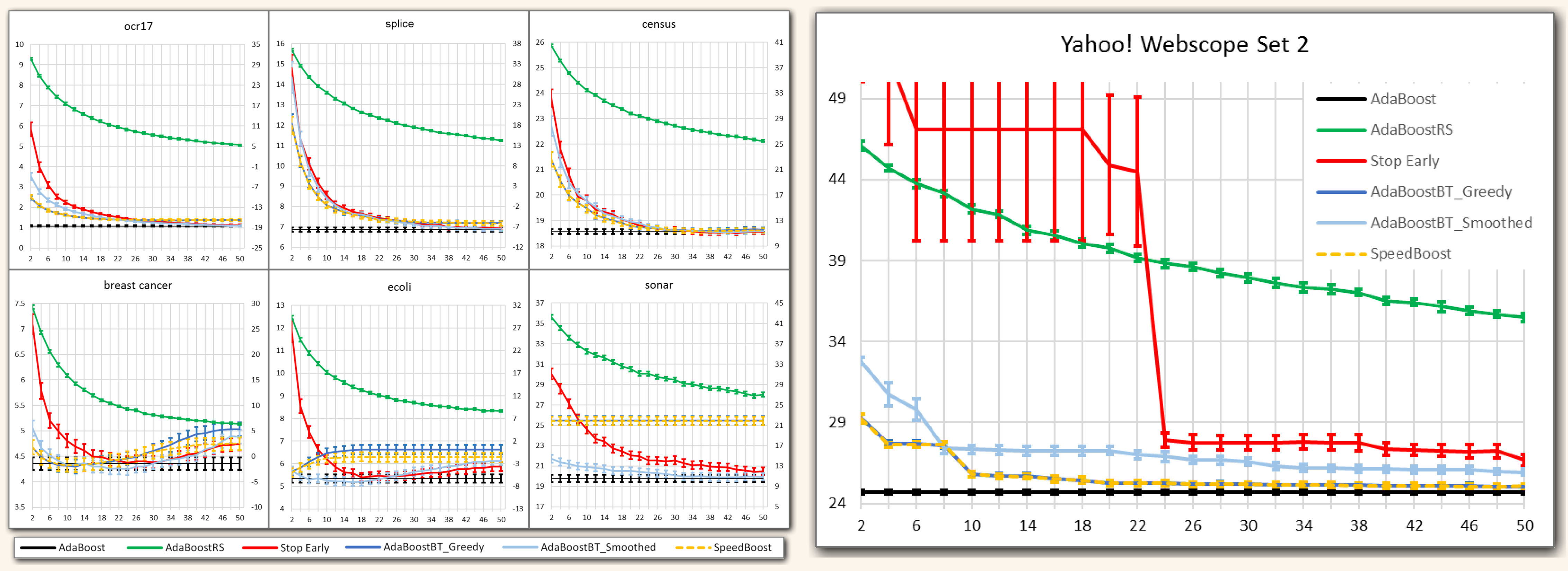

Figure 1: Experimental results with 95% confidence interval bars comparing our approaches to `AdaBoostRS` and `SpeedBoost`. Test error is calculated at budget increments of 2. The feature costs are uniformly distributed in the interval [0,2] (left) and actual (right). Horizontal axis is budget, vertical is test error rate. `AdaBoostRS` error rate uses the right-hand vertical axis for most data sets.

A Look at SpeedBoost

`SpeedBoost` [4] and `AdaBoostBT_Greedy` perform almost identically–Why?

`AdaBoostBT_Greedy`

$$\text{Find } \underset{h \in \mathcal{H}}{\text{argmin}} \, (1-\gamma(h)^2)^{\frac{1}{c(h)}}$$

`SpeedBoost` (exponential loss)

$$\text{Find } \underset{h \in \mathcal{H}}{\text{argmin}} \, \frac{1-\sqrt{1-\gamma(h)^2}}{c(h)}$$

$$\min_{h \in \mathcal{H}} (1-\gamma(h)^2)^{\frac{1}{c(h)}} = \max_{h \in \mathcal{H}} \frac{-\ln\sqrt{1-\gamma(h)^2}}{c(h)},$$

and the Taylor series of $-\ln(x)$ is

$$(1-x) + \frac{1}{2}(1-x)^2 - o\left((1-x)^2\right)$$

When $\gamma(h)$ is close to 0 the two perform very similar optimizations.

Decision Trees

CART Decision trees, an obvious solution, fail to deliver competitive generalization errors

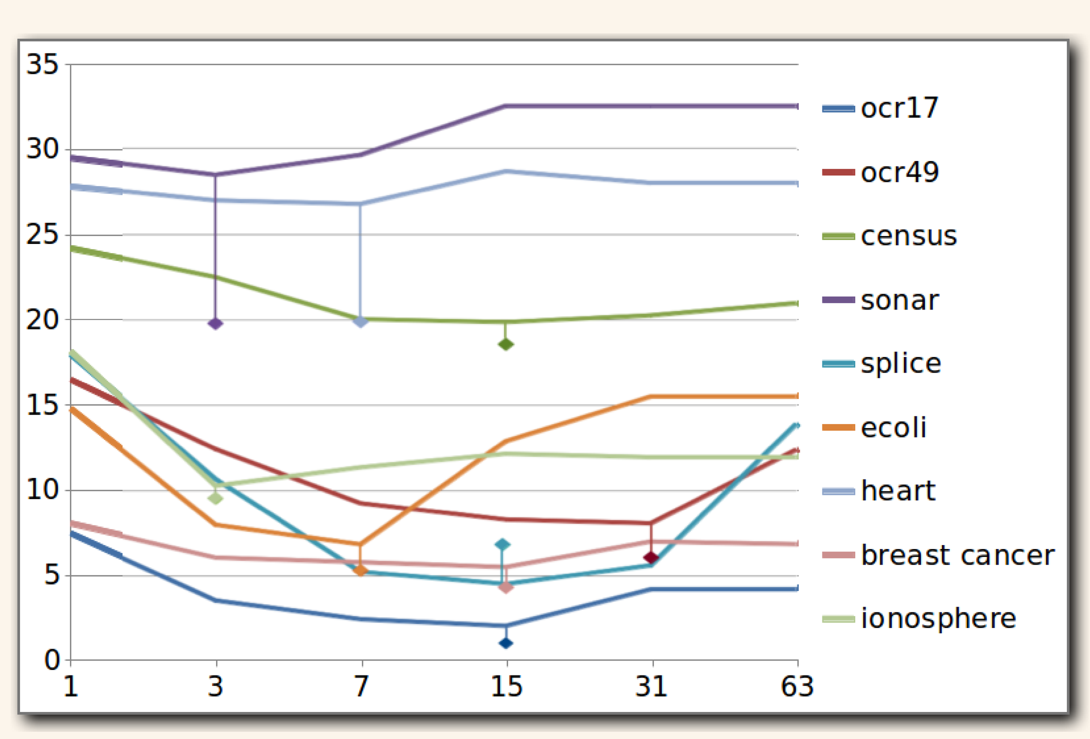

Figure 2: Error Rates of decision trees. The horizontal axis is these number of nodes. The vertical axis is percent error. Diamonds show the `AdaBoost` error rate for easy comparison.

Observations

Comparison to `AdaBoostRS`

- Budgeted Training improves significantly on `AdaBoostRS`
- Greedy and Smoothed modifications tend to yield additional improvements

Impact of Budget Size

- Greedy tends to win for small budgets
- Smoothed tends to win for larger budgets
- Both run higher risk of over-fitting than `AdaBoostBT`

The Cheap Feature Trap

- Too many cheap features can kill Greedy optimization (sonar, ecoli)
- Smoothed avoids this trap as cost becomes less important when $t \to \infty$

Yahoo! Webscope Data

- One highly predictive feature with a cost of 20
- Dramatic difference between `AdaBoostBT` and the modified algorithms
- Greedy and Smoothed create powerful low-budget classifiers

Benefits over `SpeedBoost`

- Take into account future rounds (Smoothed)
- Computational issues are avoided

References

[1] Lev Reyzin. Boosting on a budget: Sampling for feature-efficient prediction. In *ICML*, pages 529–536, 2011.

[2] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *the Annals of Statistics*, 26(5):1651–1686, 1998.

[3] R.E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. Adaptive computation and machine learning. MIT Press, 2012.

[4] Alexander Grubb and Drew Bagnell. Speedboost: Anytime prediction with uniform near-optimality. In *AISTATS*, pages 458–466, 2012.