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

Given:

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

Challenge:
Predict on new examples without going over 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

Basic `AdaBoostBT`

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

Modification 1

- 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

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

- Perhaps an aggressive assumption?

Modification 2

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

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

- Milder assumption should smooth optimization

Algorithm: AdaBoost with Budgeted Training

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

given: $(x_1, y_1), ..., (x_m, y_m) \in S$
initialize $D_1(i) = \frac{1}{m}$, $B_1 = B$
for $t = 1, \dots, T$ **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
  choose $\alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t}$, 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 $H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$

Experimental Results

data set	ocr17	ocr49	sonar	census	splice	ecoli	breast cancer	heart	ionosphere
num features	403	403	11196	131	240	356	82	371	8114
training size	1000	1000	100	1000	1000	200	500	100	300
test size	5000	5000	108	5000	2175	136	199	170	51

Table 1: Dataset sizes, and numbers of features, for training and test.

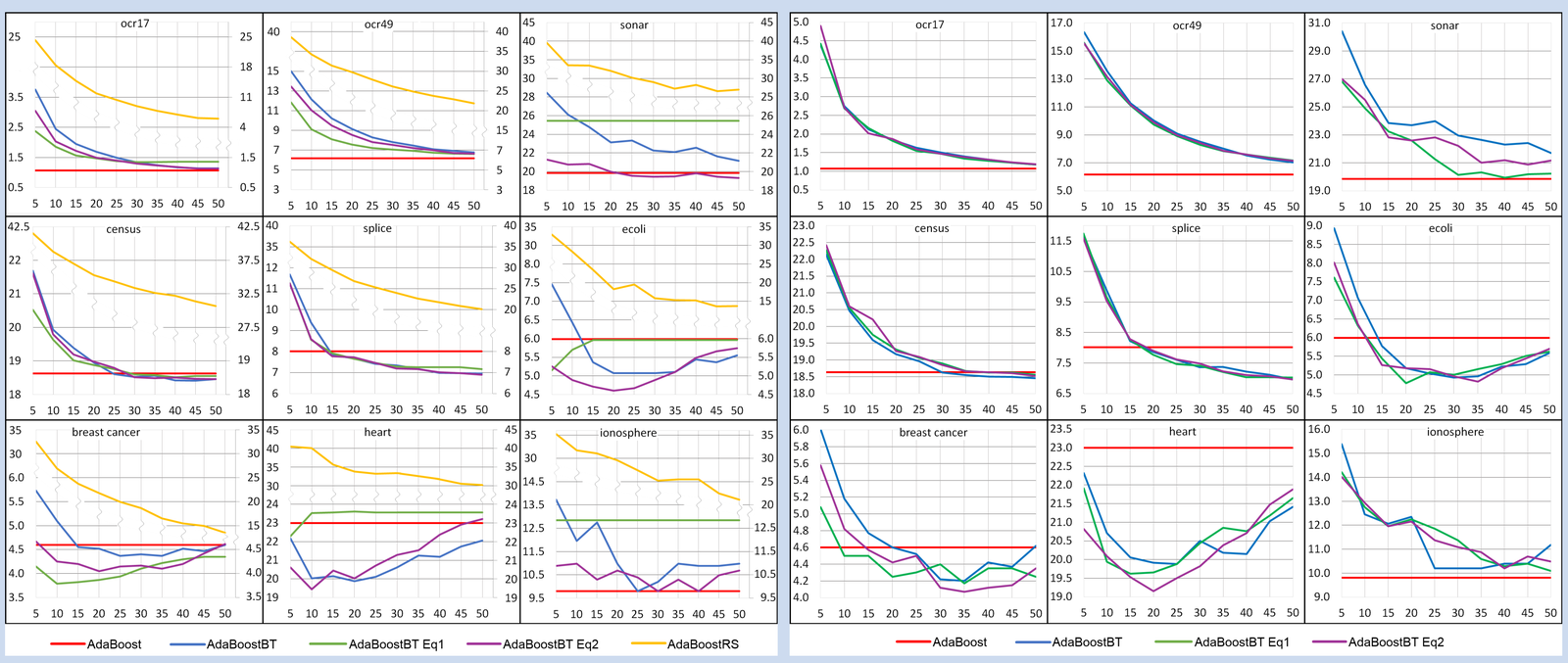

Figure 1: Experimental results compared to `AdaBoostRS` and `AdaBoost` using 500 rounds of boosting. Features costs distributions are Uniform[0,2] (left) and Normal($\mu = 1$, $\sigma = .25$) (right)

Decision Trees

Decision trees may seem an obvious solution, but they fail to deliver competitive generalization errors

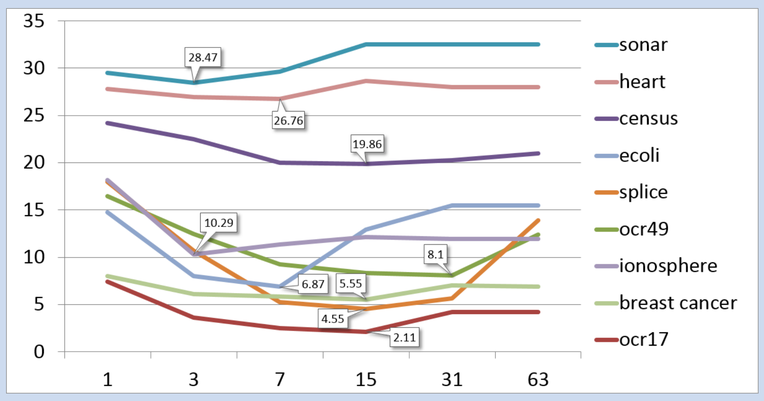

Figure 2: Error Rates of decision trees. The horizontal is these number of nodes (log scale in number of nodes, linear in expected tree depth). The vertical is percent error.

Main Reference

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

Observations

- Budgeted Training improves significantly on `AdaBoostRS`
- Modifying with Equations 1 and 2 tend to yield additional improvements
- When costs random, Equation 1 tends to win for small budgets.
- Too many cheap features can kill Equation 1 (ionosphere, sonar, heart, ecoli)
- Equation 2 avoids this trap as cost becomes less important at $t \to \infty$
- Equation 2 tends to win for larger budgets
- Both Equation 1 and 2 run higher risk of over-fitting than `AdaBoostBT`