Training-Time Optimization of a Budgeted Booster

Yi Huang Brian Powers Lev Reyzin

University of Illinois at Chicago

{yhuang,bpower6,lreyzin}@math.uic.edu

December 10, 2013

Motivation

Observing features may incur a cost

- Time, Money, Risk
- Medical diagnosis
- Internet applications

We need to classify test examples on a budget.

Feature-Efficient Learners

Goal: Supervized Learning with:

- budget $B > 0$
- feature costs $C : [i, \ldots, n] \rightarrow \mathbb{R}^+$
- Limited by budget at test time

We call such a learner **feature-efficient**

Related Work

- Determining when to stop sequential clinical trials
  Wald ('47)
- PAC-learnability with incomplete features
  Ben-David and Dichterman ('93), Greiner ('02)
- Robust predictors resilient to missing/corrupted features
  Globerson and Roweis ('06)
- Linear Predictor only accessing few features per example
  Cesa-Bianchi ('10)
- Dynamic feature selection using an MDP
  He et al. ('12)
- Feature-efficient prediction by randomly sampling from a full ensemble
  Reyzin ('11)

Reyzin's AdaBoostRS

1. Run AdaBoost to produce an ensemble predictor
2. Sample from ensemble randomly until budget is reached
3. Take unweighted average vote of samples

An Obvious Solution

There's a simpler alternative:

Stop boosting early!

Our Method: Budgeted Training

Modify AdaBoost to stop training early when budget runs out.

The resulting predictor will be feature-efficient.

Modify base learner selection when costs are non-uniform.

Algorithm: AdaBoost

`AdaBoost (S )` where: $S \subset X \times \{-1,+1\}$

given: $(x_1, y_1), \ldots, (x_m, y_m) \in S$
initialize $D_1(i) = \frac{1}{m}$
for $t = 1, \ldots, T$ **do**
    train base learner using distribution D_t.
    get $h_t \in \mathcal{H} : X \to \{-1,+1\}$.
    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 $D_{t+1}(i) = D_t(i)\exp(\alpha_t y_i h_t(x_i))/Z_t$,
end for
output the final classifier $H(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Algorithm: AdaBoost with Budgeted Training

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

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, marking 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 $D_{t+1}(i) = D_t(i)\exp(\alpha_t y_i h_t(x_i))/Z_t$,
end for
output the final classifier $H(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Optimizing for Non-Uniform Costs

- `AdaBoost` normally choses a base learner that maximizes γ_t (i.e. minimizes error rate)
- What about non-uniform costs?
- How should cost influence base learner selection?

Modified Optimization 1

Training error of `AdaBoost` is bounded by [Freund & Schapire '97]

$$\hat{\Pr}[H(x) \neq y] \leq \prod_{t=1}^{T} \sqrt{1 - \gamma_t^2}$$

- Driven down by both high γ_ts and high T (ie low costs)
- To estimate T we may make an assumption
- **If in round t we choose hypothesis h_t, assume we can find base learners with same c on future rounds.**

Modified Optimization 1: Deriving Tradeoff

Minimize training error bound

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

Modified Optimization 1: Deriving Tradeoff

If all $\gamma_i = \gamma_t(h)$

$$\text{minimize} \ \left(1 - \gamma_t(h)^2\right)^{\frac{T}{2}}$$

Modified Optimization 1: Deriving Tradeoff

$T = \frac{B}{c(h)}$ by assumption

$$\text{minimize } (1 - \gamma_t(h)^2)^{\frac{B}{2c(h)}}$$

Modified Optimization 1: Deriving Tradeoff

$\frac{B}{2}$ can be removed from exponent

$$\text{minimize } (1 - \gamma_t(h)^2)^{\frac{1}{c(h)}}$$

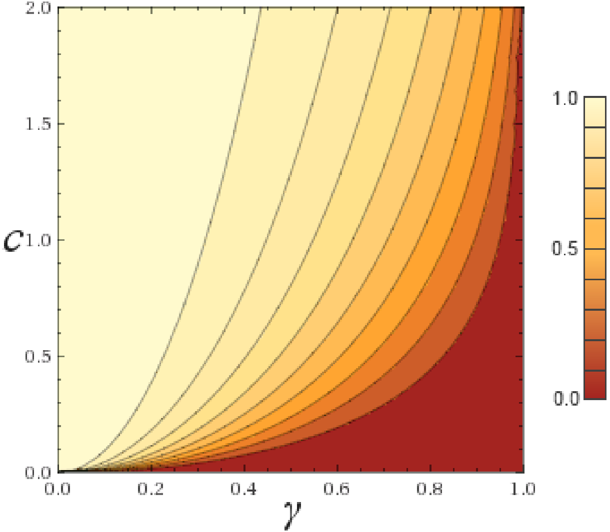
Modified Optimization 1

- We may now choose a base learner satisfying

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

Tradeoff Contours

Contour Plot of $(1-\gamma^2)^{\frac{1}{c}}$



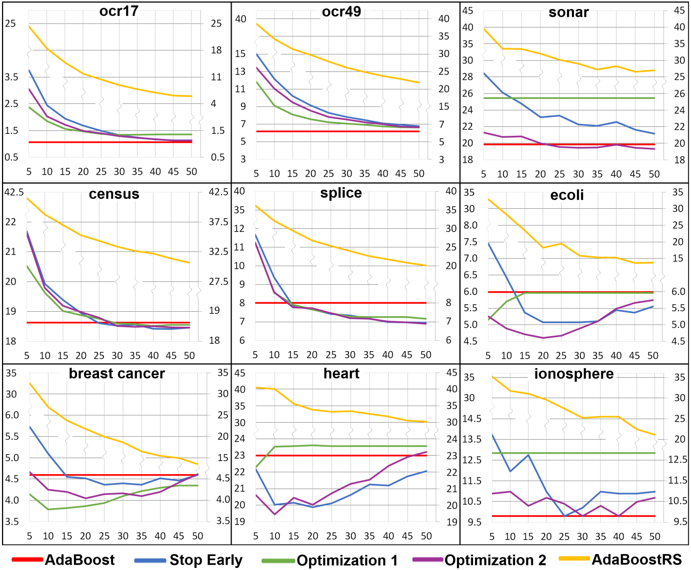
Modified Optimization 2

- Alternate estimate of T based on milder assumption
- **If in round t we choose hypothesis h_t, assume we can find base learners with c equal to the average base learner cost.**
- Average cost of base learners is $\frac{(B-B_t)+c}{t}$
- Choose a base learner satisfying

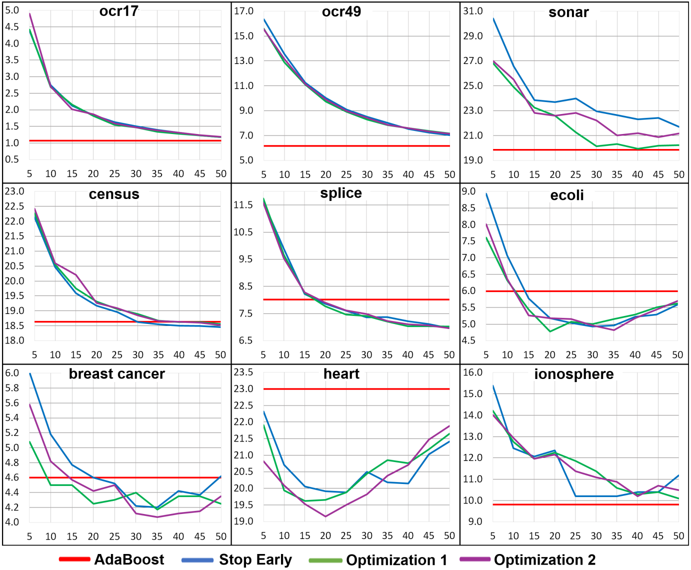
$$h_t = \operatorname{argmin}_{h \in \mathcal{H}} \left(\left(1 - \gamma_t(h)^2\right)^{\frac{1}{(B-B_t)+c(h)}} \right) \tag{2}$$

- Average cost should produce a smoother optimization

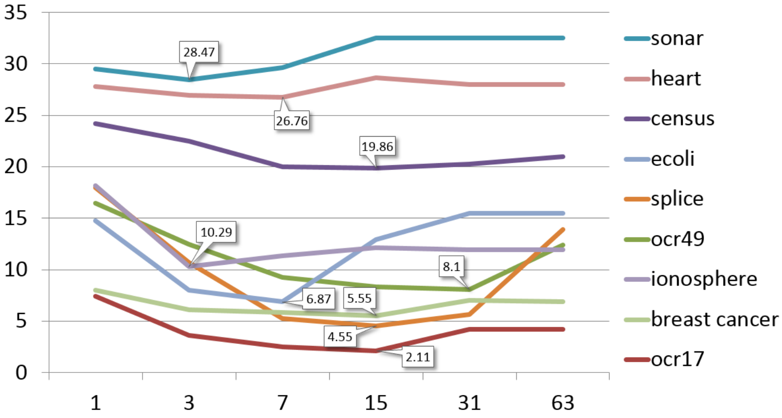
Experimental Results: $C \sim Unif(0, 2)$



Experimental Results: $C \sim N(1, .25)$



Compare to Decision Trees



Observations

- Budgeted training improves significantly on `AdaBoostRS`
- Modifying with optimizations 1 and 2 tend to yield additional improvements
- With non-uniform costs:
  - Optimization 1 tends to win for small budgets
  - Optimization 2 tends to win for larger budgets

Observations

- Too many cheap features can kill optimization 1 (ionosphere, sonar, heart, ecoli)
- Optimization 2 avoids this trap, since cost becomes less important as $t \to \infty$
- Both optimizations 1 and 2 run higher risk of over-fitting than `AdaBoostBT`

Future Work

- Improve optimization for cost distributions with few cheap features
- Consider adversarial cost models
- Boost using weak learners other than decision stumps (e.g. decision trees)
- Extend our ideas to confidence-rated predictions [Schapire & Singer '99]
- Refine optimizations by considering the complexity term in AdaBoost's generalization error bound
- Study making other machine learning algorithms feature-efficient through budgeted training

References

Shai Ben-David and Eli Dichterman (1993)
Learning with restricted focus of attention
COLT 12(3), 297 – 296.

Nicolò Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir (2010)
Efficient learning with partially observed attributes
CoRR abs/1004.4421.

Yoav Freund and Robert E. Schapire (1997)
A decision-theoretic generalization of on-line learning and an application to boosting
J. Comput. Syst. Sci., 55(1):119–139.

Amir Globerson and Sam T. Roweis (2006)
Nightmare at test time: robust learning by feature deletion
ICML, pages 353–360.

References

Russell Greiner, Adam J. Grove, and Dan Roth (2002)
Learning cost-sensitive active classifers
Artif. Intell., 139(2):137–174.

He He, Hal Daumé III, and Jason Eisner (2012)
Imitation learning by coaching
NIPS, pages 3158–3166.

Lev Reyzin (2011)
Boosting on a budget: Sampling for feature-efficient prediction
ICML, pages 529–536.

Robert E. Schapire and Yoram Singer (1999)
Improved boosting algorithms using confidence-rated predictions.
Machine Learning, 37(3):297–336.

Abraham Wald (1947)
Sequential Analysis. Wiley.

Thank You

Appendix: AdaBoost Generalization Error Bound

Occam's Razor bound gives us

$$\text{generalization error} \leq \text{training error} + \tilde{O}\left(\sqrt{\frac{dT}{m}}\right)$$

m is the number of training examples
T is the number of boosting rounds
d is the VC dimension of the base classifier