Boosting on a Feature Budget

ICML/COLT 2010 Budgeted Learning Workshop

Lev Reyzin

Yahoo! Research

Motivation

- Looking at **features** may be expensive.
  - medical diagnosis
  - Internet applications
  - etc.

- In many applications, this is especially true during **prediction**, more so than in training.

Overview

- Goal is to do supervised learning, using a limited number of features.
  - Given a **budget** on each example, the learner must look at no more features than allowed by the budget.
  - Only limited in test data, not training.
  - We call this **feature efficient** prediction.

- The idea is to modify **boosting** for this task.

A Reminder of Boosting

- Combines many "moderately inaccurate" weak learners into an ensemble predictor.
  - These are often feature efficient.
- Generates a new **weak learner** on each round.
- Outputs a **weighted distribution** of weak learners (from its hypothesis class).
- Our idea is to **sample** from the distribution over weak learners instead of taking the full vote.
  - If we don't need too many samples, the final predictor will also be feature efficient.

AdaBoostRS, an Algorithm

1) Train AdaBoost [Freund & Schapire '97].

2) To **predict** on a new example, instead of taking the full vote, **sample** the weak learners' votes according to their weights.
   - Take as many samples as your budget allows.

3) Take the **unweighted** vote of the samples as the prediction.

Margin Bound

- A **margin** is the weighted fraction of weak learners voting for the correct label.
- [Schapire et al. '98]: for any weighted vote, the generalization error is at most:

the VC dimension of the base classifier

$$\underbrace{\hat{\Pr}\left[\text{margin}_f(x,y) \leq \theta\right]}_{\text{Fraction training examples with margin below theta}} + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right)$$

number of training examples

for any value of theta

On Margins

- If AdaBoost has high margins, then AdaBoostRS doesn't need to sample too many weak learners to be in agreement.

- If AdaBoost has low margins then we don't really care how AdaBoostRS predicts.

- As AdaBoostRS takes more and more samples, in the limit its vote agrees with AdaBoost's.

Some Margin Bounds

- AdaBoost:

$$\boldsymbol{P}_D[yf(x) \le 0] \le \boldsymbol{P}_S[yf(x) \le \theta] + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right)$$

- AdaBoostRS:

$$\boldsymbol{P}_D[yf(x) \le 0] \quad \le \quad \boldsymbol{P}_S[yf(x) \le \theta] + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right) + e^{-N\theta^2/2}$$

d is VC dimension, m is number of training examples, N is number of (weak learner) samples AdaBoostRS takes

More on AdaBoost RS

- Using Decision Stumps as weak learners, each tree looks at only 1 feature. This means we can take as many samples as the budget.

- The situation is even better because of the "birthday paradox."
  - Some samples are "free."

Some Experiments

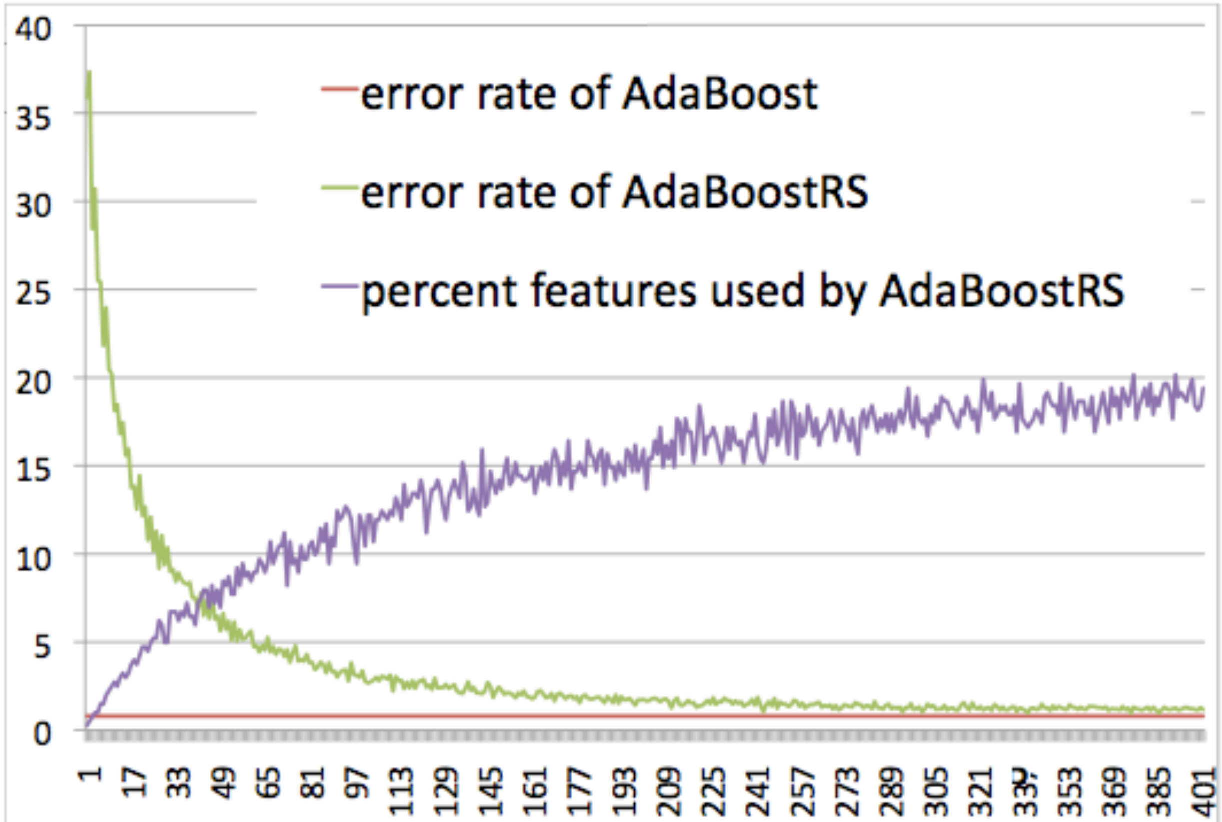

Figure 1. A graph of the error rate of AdaBoostRS on the ocr17 dataset and the percent of features it is using. The horizontal axis is the number of samples drawn by AdaBoostRS.

Some More Experiments

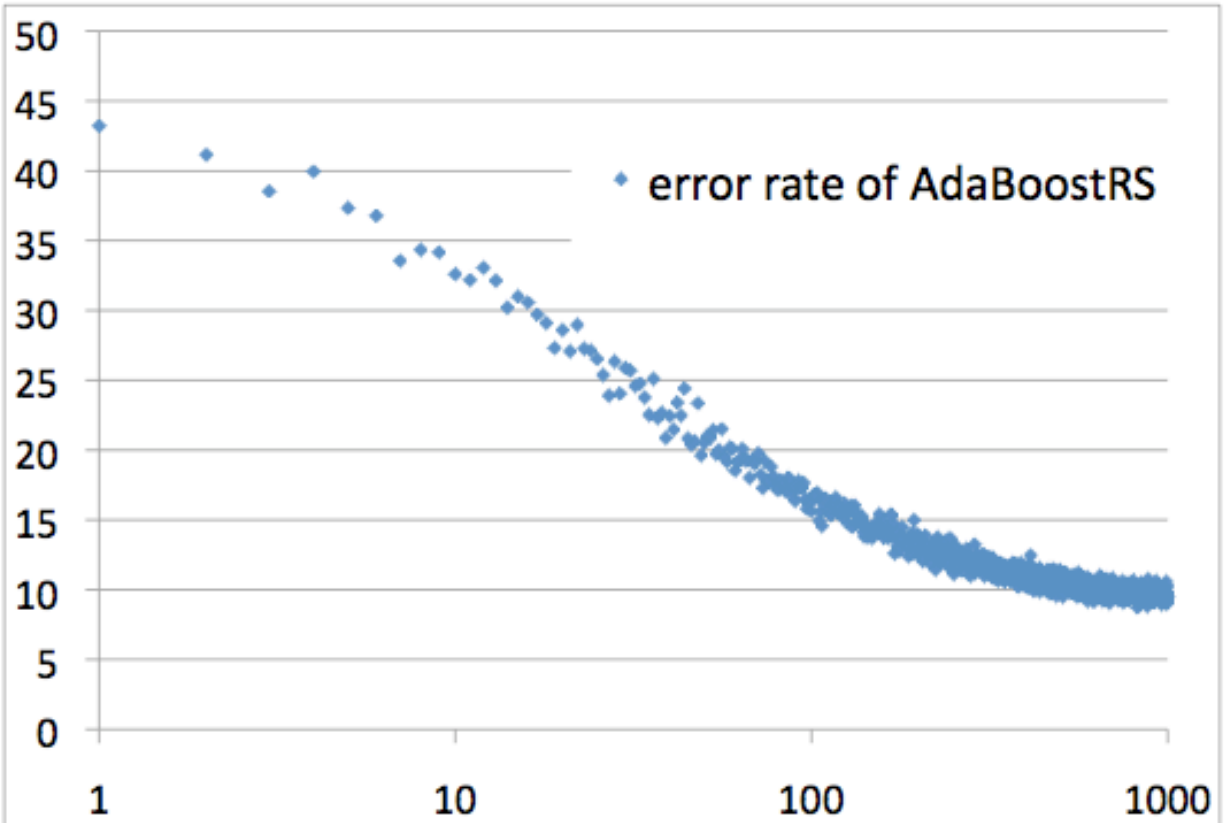

Figure 2. A graph of the error rate of AdaBoostRS on the splice dataset, as a function of the number of samples. The horizontal axis is on a log scale.

Number of Samples Needed to Reach Given Relative Error Rates to AdaBoost.

	100%	50%	25%	10%
census (18.3)	15	77	211	637
splice (8.1)	97	205	421	823
ocr17 (0.8)	178	244	339	505
ocr49 (6.5)	167	296	694	1020

Percent of Features Used to Reach Given Relative Error Rates to AdaBoost.

	100%	50%	25%	10%
census (18.3)	10.0	29.2	40.8	48.5
splice (8.1)	26.8	38.9	52.7	61.5
ocr17 (0.8)	16.4	18.4	19.9	20.6
ocr49 (6.5)	21.6	26.4	31.3	32.3

Discussion

- This method is generic -- AdaBoost can be replaced by your favorite voting algorithm.
- AdaBoostRS's budget does not have to be known in advance.
- Can be done online.
- Works with a variety of weak learners, can be non-linear.
- Sampling is "non-adaptive."

Open Problems

- How do we handle non-uniform feature costs?
- How does this compare to just taking the "top" weak learners?
- What's the best boosting algorithm to use here?
- Lots of interesting questions left to answer.