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:

\[
\Pr \left[ \text{margin}_f(x, y) \leq \theta \right] + \tilde{O} \left( \sqrt{\frac{d}{m\theta^2}} \right)
\]
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:

\[ P_D[yf(x) \leq 0] \leq P_S[yf(x) \leq \theta] + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right) \]

• AdaBoostRS:

\[ P_D[yf(x) \leq 0] \leq P_S[yf(x) \leq \theta] + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right) + e^{-N\theta^2/2} \]

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.

<table>
<thead>
<tr>
<th></th>
<th>100%</th>
<th>50%</th>
<th>25%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>census (18.3)</td>
<td>15</td>
<td>77</td>
<td>211</td>
<td>637</td>
</tr>
<tr>
<td>splice (8.1)</td>
<td>97</td>
<td>205</td>
<td>421</td>
<td>823</td>
</tr>
<tr>
<td>ocr17 (0.8)</td>
<td>178</td>
<td>244</td>
<td>339</td>
<td>505</td>
</tr>
<tr>
<td>ocr49 (6.5)</td>
<td>167</td>
<td>296</td>
<td>694</td>
<td>1020</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>100%</th>
<th>50%</th>
<th>25%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>census (18.3)</td>
<td>10.0</td>
<td>29.2</td>
<td>40.8</td>
<td>48.5</td>
</tr>
<tr>
<td>splice (8.1)</td>
<td>26.8</td>
<td>38.9</td>
<td>52.7</td>
<td>61.5</td>
</tr>
<tr>
<td>ocr17 (0.8)</td>
<td>16.4</td>
<td>18.4</td>
<td>19.9</td>
<td>20.6</td>
</tr>
<tr>
<td>ocr49 (6.5)</td>
<td>21.6</td>
<td>26.4</td>
<td>31.3</td>
<td>32.3</td>
</tr>
</tbody>
</table>
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.