Problem Setting

Given:
- Training examples $S \subset X \times \{-1, +1\}$
- Feature cost function $c : [1, \ldots, n] \rightarrow \mathbb{R}^+$
- 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 $\gamma_t$
- 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
  \[ h_t = \arg\min_{h \in H} \left( 1 - \gamma_t(h)^2 \right) \]  
  (1)
- Perhaps an aggressive assumption?

Modification 2
- Estimate $T$ by assuming future rounds will incur average cost
- Base learner is chosen to minimize
  \[ h_t = \arg\min_{h \in H} \left( 1 - \gamma_t(h)^2 \right) \]  
  (2)
- Milder assumption should smooth optimization

Algorithm: AdaBoost with Budgeted Training

\[ \text{AdaBoostBT}\left(S, B, C\right), \text{ where: } S \subset X \times \{-1, +1\}, B > 0, C : [1, \ldots, n] \rightarrow \mathbb{R}^+ \]
1: given: $(x_1, y_1), \ldots, (x_m, y_m) \in S$
2: initialize $D_1(i) = \frac{1}{m}$, $B_1 = B$
3: for $t = 1, \ldots, T$
5: get $h_t \in H : X \rightarrow \{-1, +1\}$
6: if the total cost of the unpaid features of $h_t$ exceeds $B_t$ then
7: \hspace{2em} set $T = t - 1$ and end for
8: else set $B_{t+1}$ as $B_t$ minus the total cost of the unpaid features of $h_t$, mark them as paid
9: \hspace{2em} choose $a_t = \frac{\ln \frac{1}{\beta}}{d}$, where $\gamma_t = \sum_i D_t(i) y_i h_t(x_i)$.
10: \hspace{2em} update $D_{t+1}(i) = D_t(i) \exp(a_t y_i h_t(x_i)) / Z_t$, where $Z_t$ is the normalization factor
11: end for
12: output the final classifier $H(x) = \text{sign} \left( \sum_{t=1}^{T} a_t h_t(x) \right)$

Experimental Results

<table>
<thead>
<tr>
<th>data set</th>
<th>orbit</th>
<th>cec19</th>
<th>sonar</th>
<th>census</th>
<th>splice</th>
<th>ecoli</th>
<th>breast cancer</th>
<th>heart</th>
<th>ionosphere</th>
</tr>
</thead>
<tbody>
<tr>
<td>num features</td>
<td>403</td>
<td>403</td>
<td>11196</td>
<td>131</td>
<td>240</td>
<td>256</td>
<td>89</td>
<td>571</td>
<td>8114</td>
</tr>
<tr>
<td>training size</td>
<td>1000</td>
<td>1000</td>
<td>100</td>
<td>1000</td>
<td>1000</td>
<td>200</td>
<td>500</td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>test size</td>
<td>5000</td>
<td>5000</td>
<td>108</td>
<td>5000</td>
<td>2175</td>
<td>136</td>
<td>199</td>
<td>170</td>
<td>51</td>
</tr>
</tbody>
</table>

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

Main Reference


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 $t \rightarrow \infty$
- Equation 2 tends to win for larger budgets
- Both Equation 1 and 2 run higher risk of over-fitting than AdaBoostBT