Comparing AdaBoost, AdaBoost*, Multivariate naïve Bayes, and Multinomial Binary naïve Bayes

Brian Madden
AdaBoost Algorithm

• For each feature:
  • Compute edge \( r = w \cdot u \)
  • Where:
    • \( u = y \cdot h \)
    • \( y = \text{label} \)
    • \( h = \text{feature} \)
  • Select max edge and compute weight as:
    • \( \alpha = \frac{1}{2} \ln \left( \frac{1+r}{1-r} \right) \)
  • Update weights:
    • \( w = \frac{w \cdot e^{-\alpha \cdot y \cdot h}}{Z} \)
AdaBoost* Algorithm

• For each feature:
  • Compute edge \( r = w \cdot u \)
  • Where:
    • \( u = y \cdot h \)
    • \( y = \) label
    • \( h = \) feature
  • Select max edge and compute weight as:
    • \( p = r - v \)
    • \( \alpha = \frac{1}{2} \ln\left(\frac{1+r}{1-r}\right) - \frac{1}{2} \ln\left(\frac{1+p}{1-p}\right) \)
  • Update weights:
    • \( w = \frac{w \cdot e^{(-\alpha \cdot y \cdot h)}}{Z} \)
Naïve Bayes (Multivariate)

\[
p(\vec{x} \mid c) = \prod_{i=1}^{m} p(t_i \mid c)^{x_i} \cdot (1 - p(t_i \mid c))^{(1-x_i)}.
\]

\[
\frac{p(c_s) \cdot \prod_{i=1}^{m} p(t_i \mid c_s)^{x_i} \cdot (1 - p(t_i \mid c_s))^{(1-x_i)}}{\sum_{c \in \{c_s, c_h\}} p(c) \cdot \prod_{i=1}^{m} p(t_i \mid c)^{x_i} \cdot (1 - p(t_i \mid c))^{(1-x_i)}} > T,
\]

\[
p(t \mid c) = \frac{1 + M_{t,c}}{2 + M_c},
\]

(Equations taken from paper by Metsis, Androutsopoulos, and Paliouras because I ran out of time!)
Average Accuracy vs. Ensemble Length Over 5 Folds

Average Accuracy

Ensemble Length

Average Test Accuracy
Average Training Accuracy
AdaBoost Accuracy over 10 runs

- **Accuracy over 10 runs**
- **Average accuracy over 10 runs**
- **Standard Deviation**
- **Standard Deviation**
<table>
<thead>
<tr>
<th></th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Positive</strong></td>
<td>3056 (18.7%)</td>
<td>2400 (14.7%)</td>
</tr>
<tr>
<td><strong>Negative</strong></td>
<td>10035 (61.6%)</td>
<td>790 (4.8%)</td>
</tr>
</tbody>
</table>