Weak and Strong Learning

• Suppose we have a learning algorithm that always gives reasonable but not necessarily great performance (e.g., accuracy $\geq 0.6$).

• Can we somehow use this algorithm to do better? How?

Some General and Specialized Alg

- Bagging

- Bagging of Decision Trees

- Random Trees

- Random Forests

Improving over Decision Trees

Table 2. All pairwise combinations of the four ensemble methods. Each cell contains the number of wins, losses, and ties between the algorithm in that row and the algorithm in that column.

<table>
<thead>
<tr>
<th>Noise</th>
<th>C4.5</th>
<th>Adaboost C4.5</th>
<th>Bagged C4.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>5 - 0</td>
<td>1 - 6</td>
<td>3 - 3</td>
</tr>
<tr>
<td>5%</td>
<td>3 - 0</td>
<td>1 - 5</td>
<td>5 - 1</td>
</tr>
<tr>
<td>10%</td>
<td>2 - 0</td>
<td>1 - 4</td>
<td>5 - 3</td>
</tr>
</tbody>
</table>

[D00]
Stability of Base Classifiers

- Which of these classifiers are stable/sensitive?
  - kNN
  - Decision Trees
  - Linear Threshold Elements (SVM)
  - Naive Bayes
  - “ZeroR”
  - “OneR”

Forcing Classifier Diversity

- Can we force the hypotheses produced by different runs to be different (even when base classifiers are not sensitive)?
  - How?

Confidence Rated Adaboost

Given: \{(x_1, y_1), ..., (x_m, y_m)\}: \[x_i \in X, y_i \in \{-1, +1\}\]
Initialize \[D_0(i) = \frac{1}{m}\]
For \[t = 1, ..., T\]:
  - Train weak learner using distribution \(D_t\).
  - Get weak hypothesis \(h_t: X \rightarrow \{-1, 1\}\)
  - Choose \(\alpha_t \in \mathbb{R}\)
  - Update:
    \[
    D_{t+1}(i) = \frac{D_t(i) \exp(-y_ith_t(x_i))}{Z_t}
    \]
    where \(Z_t\) is a normalization factor (chosen so that \(D_{t+1}\) will be a distribution).

Output the final hypothesis:

\[
H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_th_t(x) \right).
\]

Comparisons and Explanations

- Training set + generalization analysis
- Boosting vs Bagging vs Random Forests
- Revisit accuracy slides above
- Note sensitivity to noise
- Bias-Variance effect of bagging and boosting
- Margin explanation

Train/Test Error and Margin

<table>
<thead>
<tr>
<th></th>
<th>Boosting</th>
<th>Bagging</th>
</tr>
</thead>
<tbody>
<tr>
<td>letter</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C4.5

- Error: \(\frac{1}{2} \ln \frac{1 + r_t}{1 - r_t}\)
- When predictions of \(h_t\) are in \([-1,1]\)
  - Update is such that:
    - Error of \(h_1\) on \(D_{t+1}\) is 0.5
using decision stumps as the base learning algorithm. Results are given for the letter, satimage and vehicle datasets.

Figure 5: Error curves and margin distribution graphs for three voting methods (bagging, boosting and ECOC).

Same plot type: base learner C4.5; 20% noise

CLT says that this implies good performance

But attempts at algorithms to optimize cumulative margin directly not as successful

Sick dataset; base learner C4.5; no noise

Kappa(1 → same hyp; 0 → independent; -1 → reverse labels)

Sick dataset; base learner C4.5; 20% noise

Kappa(1 → same hyp; 0 → independent; -1 → reverse labels)
### Adaboost vs SVM

- Similar final hyp when $h_i$ is one feature
- But different optimization setting
- And different criterion:
  - Max min margin
  - Exponentially weighted cumulative margin (exponential loss)

### Ensemble Methods

- Main idea: voting among diverse set of hypothesis can help reduce errors
- Different schemes to take advantage of and/or force diversity
- Bagging, Random Forests, Ada-Boosting
- Not fully understood - and many variants exist
- Other ways of combining classifiers are also possible