Ensemble Learning

• what is an ensemble?
• why use an ensemble?
• selecting component classifiers
• selecting combining mechanism
• some results

Key Ensemble Questions

Which components to combine?
• different learning algorithms
• same learning algorithm trained in different ways
• same learning algorithm trained the same way

How to combine classifications?
• majority vote
• weighted (confidence of classifier) vote
• weighted (confidence in classifier) vote
• learned combiner

What makes a good (accurate) ensemble?

Why Do Ensembles Work?

Hansen and Salamon, 1990

If we can assume classifiers are random in predictions and accuracy > 50%, can push accuracy arbitrarily high by combining more classifiers

Key assumption: classifiers are independent in their predictions
• not a very reasonable assumption
• more realistic: for data points where classifiers predict with > 50% accuracy, can push accuracy arbitrarily high (some data points just too hard)

What Makes a Good Ensemble?

Krogh and Vedelsby, 1995

Can show that the accuracy of an ensemble is mathematically related:

\[ \hat{E} = \bar{E} - D \]

\( \hat{E} \) is the error of the entire ensemble
\( \bar{E} \) is the average error of the component classifiers
\( D \) is a term measuring the diversity of the components

Effective ensembles have accurate and diverse components

Ensemble Mechanisms - Components

• Separate learning methods
  – not often used
  – very effective in certain problems (e.g., protein folding, Rost and Sander, Zhang)

• Same learning method
  – generally still need to vary something externally
    • exception, some good results with neural networks
  – most often, data set used for training varied:
    • Bagging (Bootstrap and Aggregate), Breiman
    • Boosting, Freund & Schapire
    • Arcing, Breiman
Ensemble Mechanisms - Combiners

- Voting
- Averaging (if predictions not 0,1)
- Weighted Averaging
  - base weights on confidence in component
- Learning combiner
  - Stacking, Wolpert
  - general combiner
  - RegionBoosting, Maclin
  - piecewise combiner

Bagging

Varies data set
Each training set a bootstrap sample
- bootstrap sample - select set of examples (with replacement) from original sample

Algorithm:
for $k = 1$ to $\#\text{classifiers}$
  - $train^k$ - bootstrap sample of train set
  - create classifier using $train^k$ as training set
  - combine classifications using simple voting

Weak Learning

Schapire showed that a set of weak learners (learners with > 50% accuracy, but not much greater) could be combined into a strong learner

Idea: weight the data set based on how well we have predicted data points so far
  - data points predicted accurately - low weight
  - data points mispredicted - high weight

Result: focuses components on portion of data space not previously well predicted

Boosting - Ada

Varies weights on training data

Algorithm:
for each data point: weight $w_i$ to $1/\#\text{datapoints}$
for $k = 1$ to $\#\text{classifiers}$
  - generate classifier$_k$ with current weighted train set
  - $\varepsilon_k = \text{sum of } w_i$'s of misclassified points
  - $\beta_k = 1 - \varepsilon_k / \varepsilon_k$
  - multiply weights of all misclassified points by $\beta_k$
  - normalize weights to sum to 1

combine: weighted vote, weight for classifier$_k$ is $\log(\beta_k)$

Q: what to do if $\varepsilon_k = 0.0$ or $\varepsilon_k > 0.5$?

Boosting - Arcing

Sample data set (like Bagging), but probability of data point being chosen weighted (like Boosting)

$m_i = \#\text{number of mistakes made on point } i$ by previous classifiers

probability of selecting point $i$:

$$ prob_i = \frac{1 + m_i}{\sum_{j=1}^{N} 1 + m_j} $$

Value 4 chosen empirically
Combine using voting

Some Results - BP, C4.5 Components

<table>
<thead>
<tr>
<th>Dataset</th>
<th>C4.5</th>
<th>BP</th>
<th>BagC4</th>
<th>BagBP</th>
<th>AdaC4</th>
<th>AdaBP</th>
<th>ArcC4</th>
<th>ArcBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>letter</td>
<td>14.0</td>
<td>18.0</td>
<td>7.0</td>
<td>10.5</td>
<td>4.1</td>
<td>5.7</td>
<td>3.9</td>
<td>4.6</td>
</tr>
<tr>
<td>segment</td>
<td>3.7</td>
<td>6.6</td>
<td>3.0</td>
<td>5.4</td>
<td>1.7</td>
<td>3.5</td>
<td>1.5</td>
<td>3.3</td>
</tr>
<tr>
<td>promoter</td>
<td>12.8</td>
<td>5.3</td>
<td>10.6</td>
<td>4.0</td>
<td>6.8</td>
<td>4.5</td>
<td>6.4</td>
<td>4.6</td>
</tr>
<tr>
<td>kr-vs-kp</td>
<td>0.6</td>
<td>2.3</td>
<td>0.6</td>
<td>0.8</td>
<td>0.3</td>
<td>0.4</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td>splice</td>
<td>5.9</td>
<td>4.7</td>
<td>5.4</td>
<td>3.9</td>
<td>5.1</td>
<td>4.0</td>
<td>5.3</td>
<td>4.2</td>
</tr>
<tr>
<td>breastc</td>
<td>5.0</td>
<td>3.4</td>
<td>3.7</td>
<td>3.4</td>
<td>3.5</td>
<td>3.8</td>
<td>3.5</td>
<td>4.0</td>
</tr>
<tr>
<td>housev</td>
<td>3.6</td>
<td>4.9</td>
<td>3.6</td>
<td>4.1</td>
<td>5.0</td>
<td>5.1</td>
<td>4.8</td>
<td>5.3</td>
</tr>
</tbody>
</table>
Some Theories on Bagging/Boosting

Error = Bayes Optimal Error + Bias + Variance
Bayes Optimal Error = noise error
Theories:
- Bagging can reduce variance part of error
- Boosting can reduce variance AND bias part of error
- Bagging will hardly ever increase error
- Boosting may increase error
- Boosting susceptible to noise
- Boosting’s increases margins

Combiner - Stacking

Idea:
- generate component (level 0) classifiers with part of the data (half, three quarters)
- train combiner (level 1) classifier to combine predictions of components using remaining data
- retrain component classifiers with all of training data
In practice, often equivalent to voting

Combiner - RegionBoost

- Train “weight” classifier for each component classifier
- “weight” classifier predicts how likely point will be predicted correctly
- “weight” classifiers: k-Nearest Neighbor, Backprop
- Combiner, generate component classifier prediction and weight using corresponding “weight” classifier
- Small gains in accuracy