Ensemble Methods: Bagging, Boosting, etc.

Lucila Ohno-Machado
HST951

Topics

• Bagging
• Boosting
  – Ada-Boosting
  – Arcing
• Stacked Generalization
• Mixture of Experts

Bias plus variance decomposition

• Geman, 1992

• Bias: how close the average classifier is to the gold standard
• Variance: how often classifiers disagree
• Intrinsic target noise: error of Bayes optimal classifier

Combining classifiers

• Examples: classification trees and neural networks, several neural networks, several classification trees, etc.
• Average results from different models
• Why?
  – Better classification performance than individual classifiers
  – More resilience to noise
• Why not?
  – Time consuming
  – Overfitting

Bagging

• Breiman, 1996
• Derived from bootstrap (Efron, 1993)

• Create classifiers using training sets that are bootstrapped (drawn with replacement)
• Average results for each case

Bagging Example (Opitz, 1999)

<table>
<thead>
<tr>
<th>Original</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set 1</td>
<td>2</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
**Bagging Example (Opitz, 1999)**

Original

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set 1</td>
<td>2</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Training set 2</td>
<td>7</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>1</td>
</tr>
</tbody>
</table>

**Boosting**

- A family of methods
- Sequential production of classifiers
- Each classifier is dependent on the previous one, and focuses on the previous one’s errors
- Examples that are incorrectly predicted in previous classifiers are chosen more often or weighted more heavily

**Boosting Example (Opitz, 1999)**

Original

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set 1</td>
<td>2</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Training set 2</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>
Ada-Boosting

- Freund and Schapire, 1996
- Two approaches
  - Select examples according to error in previous classifier (more representatives of misclassified cases are selected) – more common
  - Weigh errors of the misclassified cases higher (all cases are incorporated, but weights are different) – not for all algorithms

\[ \varepsilon_k = \sum \text{probabilities for misclassified instances for current classifier } C_k \]

Multiply probability of selecting misclassified cases by
\[ \beta_k = \frac{(1 - \varepsilon_k)}{\varepsilon_k} \]

"Renormalize" probabilities (i.e., rescale so that it sums to 1)

Combine classifiers \( C_1 \ldots C_k \) using weighted voting where \( C_k \) has weight \( \log(\beta_k) \)

Arcing

- Arcing-x4 (Breiman, 1996)
- For the \( i \)th example in the training set, \( m_i \) refers to the number of times that it was misclassified by the previous \( K \) classifiers
- Probability \( p_i \) of selecting example \( i \) in the next classifier is
\[ p_i = \frac{1 + m_i}{\sum_{j=1}^{N} 1 + m_j} \]
- Empirical determination

Empirical comparison (Opitz, 1999)

- 23 data sets from UCI repository
- 10-fold cross validation
- Backpropagation neural nets
- Classification trees
- Single, Simple (multiple NNs with different initial weights), Bagging, Ada-boost, Arcing
- # of classifiers in ensemble
- "Accuracy" as evaluation measure

Correlation coefficients

<table>
<thead>
<tr>
<th></th>
<th>Neural Net</th>
<th>Classification Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Simple</td>
<td>Bagging</td>
</tr>
<tr>
<td>Simple NN</td>
<td>.88</td>
<td>.87</td>
</tr>
<tr>
<td>Bagging NN</td>
<td>.88</td>
<td>.78</td>
</tr>
<tr>
<td>Arcing NN</td>
<td>.87</td>
<td>.78</td>
</tr>
<tr>
<td>Ada NN</td>
<td>.85</td>
<td>.78</td>
</tr>
<tr>
<td>Bagging CT</td>
<td>-.11</td>
<td>.14</td>
</tr>
<tr>
<td>Arcing CT</td>
<td>.38</td>
<td>.35</td>
</tr>
<tr>
<td>Ada CT</td>
<td>.37</td>
<td>.35</td>
</tr>
</tbody>
</table>

Results

• Ensembles generally better than single, but not so different from “Simple” (NNs with different initial random weights)
• Ensembles within NNs and CTs are strongly correlated
• Ada-boosting and arcing strongly correlated even across different algorithms (boosting depends more on data set than type of classifier algorithm)
• 40 networks in ensemble were sufficient
• NNs generally better than CTs

More results

• Created data sets with different levels of noise (random selection of possible value for a feature or outcome) from the 23 sets
• Created artificial data with noise

Conclusion:
• Boosting worse with more noise

Other work

• Opitz and Shavlik
  – Genetic search for classifiers that are accurate yet different
• Create diverse classifiers by:
  – Using different parameters
  – Using different training sets


Stacked Generalization

• Wolpert, 1992
• Level-0 models are based on different learning models and use original data (level-0 data)
• Level-1 models are based on results of level-0 models (level-1 data are outputs of level-0 models) -- also called “generalizer”

Empirical comparison

• Ting, 1999
• Compare SG to best model and to arcing and bagging
• Stacked C4.5, naïve Bayes, and a nearest neighbor learner
• Used multi-response linear regression as generalizer

Results

• SG had better performance (accuracy) than best level-0 model
• Use of continuous estimates better than use of predicted class
• Better than majority vote
• Similar performance as arcing and bagging
• Good for parallel computation (like bagging)

Related work

• Decomposition of problem into subtasks
• Mixture of experts (Jacobs, 1991)
  – Each expert here takes care of a certain input space
• Hierarchical neural networks
  – Difference is that cases are routed to pre-defined expert networks


Ideas for final projects

• Compare single, bagging, and boosting on other classifiers (e.g., logistic regression, rough sets)
• Use other data sets for comparison
• Use other performance measures
• Study the effect of voting scheme
• Try to find a relationship between initial performance, number of cases, and number of classifiers within an ensemble
• Genetic search for good diverse classifiers
• Analyze effect of prior outlier removal on boosting

Some software

• Classification trees, association rules
  – http://www.rulequest.com/
• Neural nets
  – http://brain.unr.edu/FILES_PHP/show_papers.php#software
• Logistic Regression
  – R, SAS, SPSS
• SVM
  – http://www.support-vector.net/software.html
• Rough Sets
  – http://www.idi.ntnu.no/~aleks/rosetta/download/