More Machine Learning

(Chapter 18)

Another Information Example

<table>
<thead>
<tr>
<th></th>
<th>stock</th>
<th>rolling</th>
<th>the</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>3</td>
<td>40</td>
<td>other</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>8</td>
<td>35</td>
<td>finance</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>7</td>
<td>25</td>
<td>other</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>7</td>
<td>14</td>
<td>other</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>2</td>
<td>20</td>
<td>finance</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>4</td>
<td>25</td>
<td>finance</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>6</td>
<td>20</td>
<td>finance</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>2</td>
<td>35</td>
<td>other</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>11</td>
<td>25</td>
<td>finance</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>15</td>
<td>28</td>
<td>other</td>
</tr>
</tbody>
</table>
Gain(rolling) = 1 - \left\{ \frac{4}{10}H(1/2, 1/2) + \frac{4}{10}H(1/2, 1/2) + \frac{2}{10}H(1/2, 1/2) \right\} = 0

ML in Practice: General Approach

- Formulate task
- (Prior model (parameters, structure))
- Obtain data
- What representation should be used? (attribute/value pairs)
- Annotate data
- Learn/refine model with data (training)
- Use model for classification or prediction on unseen data (testing)
- Measure accuracy
Issues

- Representation
  - How to map from a representation in the domain to a representation used for learning?
- Training data
  - How can training data be acquired?
- Amount of training data
  - How well does the algorithm do as we vary the amount of data?
- Which attributes influence learning most?
- Does the learning algorithm provide insight into the generalizations made?

Other Decision Tree cases

- What if class is discrete valued, not binary?

- What if an attribute has many values (e.g., 1 per instance)?
Training vs. Testing

- A learning algorithm is good if it uses its learned hypothesis to make accurate predictions on unseen data
  - Collect a large set of examples (with classifications)
  - Divide into two disjoint sets: the training set and the test set
  - Apply the learning algorithm to the training set, generating hypothesis $h$
  - Measure the percentage of examples in the test set that are correctly classified by $h$
  - Repeat for different sizes of training sets and different randomly selected training sets of each size.

Division into 3 sets

- Inadvertent peeking
  - Parameters that must be learned (e.g., how to split values)
  - Generate different hypotheses for different parameter values on training data
  - Choose values that perform best on testing data
  - Why do we need to do this for selecting best attributes?
Overfitting

- Learning algorithms may use irrelevant attributes to make decisions
  - For news, day published and newspaper

- Decision tree pruning
  - Prune away attributes with low information gain
  - Use statistical significance to test whether gain is meaningful

K-fold Cross Validation

- To reduce overfitting

- Run k experiments
  - Use a different $1/k$ of data for testing each time
  - Average the results

- 5-fold, 10-fold, leave-one-out
Ensemble Learning

• Learn from a collection of hypotheses
• Majority voting
• Enlarges the hypothesis space

Boosting

• Uses a weighted training set
  – Each example as an associated weight \( w_j \geq 0 \)
  – Higher weighted examples have higher importance
• Initially, \( w_j = 1 \) for all examples
• Next round: increase weights of misclassified examples, decrease other weights
• From the new weighted set, generate hypothesis \( h_2 \)
• Continue until \( M \) hypotheses generated
• Final ensemble hypothesis = weighted-majority combination of all \( M \) hypotheses
  • Weight each hypothesis according to how well it did on training data
AdaBoost

• If input learning algorithm is a weak learning algorithm
  – L always returns a hypothesis with weighted error on training slightly better than random
• Returns hypothesis that classifies training data perfectly for large enough M
• *Boosts* the accuracy of the original learning algorithm on training data