Model Selection and Assessment

CS4780/5780 – Machine Learning
Fall 2013
Thorsten Joachims
Cornell University

Reading:
Mitchell Chapter 5
(http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.3325)

Outline

- Model Selection
  - Controlling overfitting in decision trees
  - Train, validation, test
  - K-fold cross validation
- Evaluation
  - What is the true error of classification rule $h$?
  - Is rule $h_1$ more accurate than $h_2$?
  - Is learning algorithm A1 better than A2?

Overfitting

![Graph showing overfitting](image)

- Note: Accuracy = 1.0 - Error

Reduced-Error Pruning

![Graph showing reduced-error pruning](image)

Reduced-Error Pruning

- Early Stopping: Stop growing the tree and introduce leaf when splitting no longer “reliable”.
  - Restrict size of tree (e.g., number of nodes, depth)
  - Minimum number of examples in node
  - Threshold on splitting criterion
- Post Pruning: Grow full tree, then simplify.
  - Reduced-error tree pruning
  - Rule post-pruning

Model Selection

![Diagram of model selection process](image)

- Training: Run learning algorithm $m$ times (e.g. different parameters).
- Validation Error: $\text{Errors}_{\text{Err}}(h)$ is an estimate of $\text{Err}(h)$ for each $h_i$.
- Selection: Use $h_i$ with min $\text{Errors}_{\text{Err}}(h_i)$ for prediction on test examples.
K-fold Cross Validation

- Given
  - Sample of labeled instances S
  - Learning Algorithms A
- Compute
  - Randomly partition S into k equally sized subsets $S_1$ ... $S_k$
  - For $i$ from 1 to k
    - Train A on $S_1$ ... $S_i$ $S_{i+1}$ ... $S_k$ and get $\hat{h}$.
    - Apply $h$ to $S_i$ and compute $Err(h)$.
- Estimate
  - Average $Err(h)$ is estimate of average prediction error of rules produced by A, namely $E(Err(A(S_{train}))$

Evaluate Learned Hypotheses

- Real-world Process
  - Split S randomly
  - Draw i.i.d.
  - Evaluation: Learner $\rightarrow$ Test Sample $S_{test}$
  - Learner: $\rightarrow$ $S_{train}$ (incl. ModSel)

- Goal: Find $h$ with small prediction error $Err(h)$ over $P(X)$.
- Question: How good is $Err(h)$ of $h$ found on training sample $S_{train}$?

- Training Error: $Err_{train}(h)$ on training sample.
- Test Error: $Err_{test}(h)$ is an estimate of $Err(h)$.

Text Classification Example: “Corporate Acquisitions” Results

- Unpruned Tree (ID3 Algorithm):
  - Size 437 nodes
  - Training Error: 0.0%
  - Test Error: 11.0%
- Early Stopping Tree (ID3 Algorithm):
  - Size 299 nodes
  - Training Error: 2.6%
  - Test Error: 9.8%
- Reduced-Error Tree Pruning (C4.5 Algorithm):
  - Size 167 nodes
  - Training Error: 4.0%
  - Test Error: 10.8%
- Rule Post-Pruning (C4.5 Algorithm):
  - Size 164 tests
  - Training Error: 3.1%
  - Test Error: 10.3%

What is the True Error of a Hypothesis?

- Given
  - Sample of labeled instances S
  - Learning Algorithm A
- Setup
  - Partition S randomly into $S_{train}$ (70%) and $S_{test}$ (30%)
  - Train learning algorithm A on $S_{train}$, result is $h$.
  - Apply $h$ to $S_{test}$ and compare predictions against true labels.
- Test
  - Error on test sample $Err_{test}(h)$ is estimate of true error $Err(h)$.
  - Compute confidence interval.

Text Classification Example: Results

- Data
  - Training Sample: 2000 examples
  - Test Sample: 600 examples
- Unpruned Tree:
  - Size 437 nodes
  - Training Error: 0.0%
  - Test Error: 11.0%
- Early Stopping Tree:
  - Size 299 nodes
  - Training Error: 2.6%
  - Test Error: 9.8%
- Post-Pruned Tree:
  - Size 167 nodes
  - Training Error: 4.0%
  - Test Error: 10.8%
- Rule Post-Pruning:
  - Size 164 tests
  - Training Error: 3.1%
  - Test Error: 10.3%

Binomial Distribution

- The probability of observing $x$ heads in a sample of $n$ independent coin tosses, where in each toss the probability of heads is $p$, is
  \[ P(X = x | n, p) = \binom{n}{x} p^x (1-p)^{n-x} \]
- Normal approximation: For $np(1-p)>5$ the binomial can be approximated by the normal distribution with
  - Expected value: $E(X)=np$
  - Variance: $Var(X)=np(1-p)$
- With probability $\delta$, the observation $x$ falls in the interval
  \[ z_{\frac{1}{2}} \leq z \leq z_{\frac{1}{2}+\delta} \]

<table>
<thead>
<tr>
<th>$z$</th>
<th>0.50</th>
<th>0.60</th>
<th>0.70</th>
<th>0.80</th>
<th>0.90</th>
<th>0.95</th>
<th>0.98</th>
<th>0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.645</td>
<td>1.96</td>
<td>2.33</td>
<td>2.58</td>
<td>1.285</td>
<td>1.645</td>
<td>1.965</td>
<td>2.335</td>
<td>2.575</td>
</tr>
</tbody>
</table>
Is Rule $h_1$ More Accurate than $h_2$?  
(Same Test Sample)
- Given
  - Sample of labeled instances $S$
  - Learning Algorithms $A_1$ and $A_2$
- Setup
  - Randomly partition $S$ into $S_{\text{train}}$ (70%) and $S_{\text{test}}$ (30%)
  - Train learning algorithms $A_1$ and $A_2$ on $S_{\text{train}}$ result are $\hat{h}_1$ and $\hat{h}_2$.
  - Apply $\hat{h}_1$ and $\hat{h}_2$ to $S_{\text{test}}$ and compute $Err_{\text{test}}(\hat{h}_1)$ and $Err_{\text{test}}(\hat{h}_2)$.
- Test
  - Decide, if $Err_{\text{test}}(\hat{h}_1) = Err_{\text{test}}(\hat{h}_2)$?
  - Null Hypothesis: $Err_{\text{test}}(\hat{h}_1)$ and $Err_{\text{test}}(\hat{h}_2)$ come from binomial distributions with same p.
  - $\rightarrow$ Binomial Sign Test (McNemar’s Test)

Is Rule $h_1$ More Accurate than $h_2$?  
(Different Test Samples)
- Given
  - Samples of labeled instances $S_1$ and $S_2$
  - Learning Algorithms $A_1$ and $A_2$
- Setup
  - Randomly partition $S_1$ into $S_{\text{train1}}$ (70%) and $S_{\text{test1}}$ (30%)
  - Randomly partition $S_2$ into $S_{\text{train2}}$ (70%) and $S_{\text{test2}}$ (30%)
  - Train learning algorithms $A_1$ on $S_{\text{train1}}$ and $A_2$ on $S_{\text{train2}}$ result are $\hat{h}_1$ and $\hat{h}_2$.
  - Apply $\hat{h}_1$ to $S_{\text{test1}}$ and $\hat{h}_2$ to $S_{\text{test2}}$ and get $Err_{\text{test1}}(\hat{h}_1)$ and $Err_{\text{test2}}(\hat{h}_2)$.
- Test
  - Decide, if $Err_{\text{test1}}(\hat{h}_1) = Err_{\text{test2}}(\hat{h}_2)$?
  - Null Hypothesis: $Err_{\text{test1}}(\hat{h}_1)$ and $Err_{\text{test2}}(\hat{h}_2)$ come from binomial distributions with same p.
  - $\rightarrow$ t-Test ($z$-Test) [$\rightarrow$ see Mitchell book]

Is Learning Algorithm $A_1$ better than $A_2$?
- Given
  - $k$ samples $S_1, ..., S_k$ of labeled instances, all i.i.d. from $P(X,Y)$.
  - Learning Algorithms $A_1$ and $A_2$
- Setup
  - For $i$ from 1 to $k$
    - Randomly partition $S_i$ into $S_{\text{train}}$ (70%) and $S_{\text{test}}$ (30%)
    - Train learning algorithms $A_1$ and $A_2$ on $S_{\text{train}}$ result are $\hat{h}_1$ and $\hat{h}_2$.
    - Apply $\hat{h}_1$ and $\hat{h}_2$ to $S_{\text{test}}$ and compute $Err_{\text{test}}(\hat{h}_1)$ and $Err_{\text{test}}(\hat{h}_2)$.
- Test
  - Decide, if $E_i(Err_{\text{test}}(\hat{h}_1)) = E_i(Err_{\text{test}}(\hat{h}_2))$?
  - Null Hypothesis: $Err_{\text{test}}(A_1(S_{\text{train}}))$ and $Err_{\text{test}}(A_2(S_{\text{train}}))$ come from same distribution over samples $S$.
  - $\rightarrow$ t-Test ($z$-Test) or Wilcoxon Signed-Rank Test [$\rightarrow$ see Mitchell book]

Approximation via K-fold Cross Validation
- Given
  - Sample of labeled instances $S$
  - Learning Algorithms $A_1$ and $A_2$
- Compute
  - Randomly partition $S$ into $k$ equally sized subsets $S_1, ..., S_k$
    - For $i$ from 1 to $k$
      - Train $A_1$ and $A_2$ on $S_1, ..., S_{i-1}, S_{i+1}, ..., S_k$ and get $\hat{h}_1$ and $\hat{h}_2$.
      - Apply $\hat{h}_1$ and $\hat{h}_2$ to $S_i$ and compute $Err_{\text{test}}(\hat{h}_1)$ and $Err_{\text{test}}(\hat{h}_2)$.
- Estimate
  - Average $Err_{\text{test}}(\hat{h}_1)$ is estimate of $E_i(Err_{\text{test}}(A_1(S_{\text{train}})))$
  - Average $Err_{\text{test}}(\hat{h}_2)$ is estimate of $E_i(Err_{\text{test}}(A_2(S_{\text{train}})))$
  - Count how often $Err_{\text{test}}(\hat{h}_1) > Err_{\text{test}}(\hat{h}_2)$ and $Err_{\text{test}}(\hat{h}_1) < Err_{\text{test}}(\hat{h}_2)$