Overfitting, Cross Validation, MDL, Structural Risk Minimization, Using unlabeled data

Tom M. Mitchell
Machine Learning
10-701
October, 2003
Many Ways to Address Overfitting

• Cross validation
  – K-fold
  – Leave One Out cross validation
• Structural risk minimization
• Minimum description length “principle” (MDL)
• Bayesian Information Criterion (BIC)
• Using unlabeled data
Cross Validation

- Separate data into train, validation sets
- Learn hypothesis using training set
- Use validation set to prune/select hypothesis
Cross Validation

- Separate data into train, validation sets
- Learn hypothesis using training set
- Use validation set to prune/select hypothesis
  - Choose validation set large enough to obtain low-variance estimate of true error

- When \( h \) is a boolean function, and \( S \) is a sample of data containing \( n \geq 30 \) examples drawn independently of each other and of \( h \), the 95% confidence interval for the true error of \( h \) is approx

\[
error_S(h) \pm 1.96 \sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}
\]
Cross Validation

• Note we distrust training error of \( h \) as an estimate of true error of \( h \) because our choice of \( h \) is dependent on the training data
  – Training error gives optimistically biased estimate

• Then why trust validation set error of \( h \) if we are using it to prune/select \( h \)?
  – Good way to prune
  – Optimistic way to estimate resulting error

• We shouldn’t really…
  – Though the estimate provided by the validation set is usually less biased (why?)
Cross Validation

• So the proper way to learn, prune/select, then obtain an unbiased estimate of true error is:

Separate data into 3 sets:
  – Use *training set* to learn hypothesis (e.g. decision tree, neural net)
  – Use *validation set* to prune/select the hypothesis
  – Use *test set* to obtain unbiased estimate of error
K-fold Cross Validation

Problem: When training data limited, withholding data for validation set hurts. We want to use it for training!

K-fold cross validation (to estimate error):
- Partition m available examples into k disjoint subsets (called ‘folds’)
- For i=1 to k
  - Train using all folds except fold i
  - Use fold i to obtain unbiased estimate of true error
When finished, output mean error over all folds

When k=m, we have leave-one-out cross validation
- Which allows training on m-1 examples repeatedly
- Most efficient use of data/most computationally expensive
- Some contention remains over whether/when this is best approach…
Minimum Description Length Principle

Occam’s razor: prefer the shortest hypothesis

MDL: prefer the hypothesis $h$ that minimizes

$$h_{MDL} = \arg\min_{h \in H} L_{C_1}(h) + L_{C_2}(D|h)$$

where $L_C(x)$ is the description length of $x$ under encoding $C$

Example: $H =$ decision trees, $D =$ training data labels

- $L_{C_1}(h)$ is $\#$ bits to describe tree $h$
- $L_{C_2}(D|h)$ is $\#$ bits to describe $D$ given $h$
  - Note $L_{C_2}(D|h) = 0$ if examples classified perfectly by $h$. Need only describe exceptions
- Hence $h_{MDL}$ trades off tree size for training errors
Minimum Description Length Principle

\[
h_{MAP} = \arg \max_{h \in H} P(D|h)P(h) = \arg \max_{h \in H} \log_2 P(D|h) + \log_2 P(h) = \arg \min_{h \in H} -\log_2 P(D|h) - \log_2 P(h)
\]  

(1)

Interesting fact from information theory:

The optimal (shortest expected coding length) code for an event with probability \( p \) is \(- \log_2 p\) bits.

So interpret (1):

- \(- \log_2 P(h)\) is length of \( h \) under optimal code
- \(- \log_2 P(D|h)\) is length of \( D \) given \( h \) under optimal code

\(\rightarrow\) prefer the hypothesis that minimizes

\[\text{length}(h) + \text{length(misclassifications)}\]
Structural Risk Minimization

From PAC theory (Vapnik, 1995) we know that with probability \((1 - \delta)\)

\[
err_D \leq err_D + \sqrt{\frac{VC(H)(\log(2m/VC(H)) + 1)}{m}} - \log(\delta/4)
\]

- \(err_D\) is true error of \(h\)
- \(err_D\) is error of \(h\) on training set \(D\)
- \(m\) is number of training examples in \(D\)
- \(VC(H)\) is VC dimension of hypothesis space \(H\)

So, choose among \(H\)’s with different \(VC(H)\) to minimize this!

- e.g., \(H_k\) = decision trees of depth \(k\)
- often used to train Support Vector Machines
Summary of Overfitting

- **Empirical**: Cross-validation methods use data to make decision of which hypothesis is best

- **Theoretical**: MDL and Structural Risk Minimization are theory-based methods that use assumptions about which hypotheses are a priori most likely (together with the data)
  - BIC and AIC are two other theory-based methods

- Note there is no free lunch! – Without prior assumptions of some kind, one can never generalize beyond the observed data
Define metric over $H \cup \{f\}$

$$d(h_1, h_2) \equiv \int \delta(h_1(x) \neq h_2(x)) p(x) dx$$

$$\hat{d}(h_1, f) = \frac{1}{|L|} \sum_{x_i \in L} \delta(h_1(x_i) \neq y_i)$$

$$\hat{d}(h_1, h_2) = \frac{1}{|U|} \sum_{x \in U} \delta(h_1(x) \neq h_2(x))$$

Organize $H$ into complexity classes, sorted by $P(h)$

Let $h_i^*$ be hypothesis with lowest $\hat{d}(h, f)$ in $H_i$

Prefer $h_1^*$, $h_2^*$, or $h_3^*$?
• Definition of distance metric
  – Non-negative: \( d(f,g) \geq 0; \)
  – Symmetric: \( d(f,g) = d(g,f); \)
  – Triangle inequality: \( d(f,g) \leq d(f,h) + d(h,g) \)

• Classification with zero-one loss:
  \[
  d(h_1, h_2) \equiv \int \delta(h_1(x) \neq h_2(x))p(x)dx
  \]

• Regression with squared loss:
  \[
  d(h_1, h_2) \equiv \sqrt{\int (h_1(x) - h_2(x))^2p(x)dx}
  \]
Idea: Use $U$ to Avoid Overfitting

Biased estimates based on training data

Unbiased estimate based on unlabeled data, not used for training

Note:
- $\hat{d}(h_i^*, f)$ optimistically biased (too short)
- $\hat{d}(h_i^*, h_j^*)$ unbiased
- Distances must obey triangle inequality!
  \[
  d(h_1, h_2) \leq d(h_1, f) + d(f, h_2)
  \]

→ Heuristic:
- Continue training until $\hat{d}(h_i, h_{i+1})$ fails to satisfy triangle inequality
**Procedure TRI**

- Given hypothesis sequence $h_0, h_1, ...$
- Choose the last hypothesis $h_\ell$ in the sequence that satisfies the triangle inequality $d(h_k, h_\ell) \leq d(h_k, P_{y|x}) + d(h_\ell, P_{y|x})$ with every preceding hypothesis $h_k$, $0 \leq k < \ell$. (Note that the inter-hypothesis distances $d(h_k, h_\ell)$ are measured on the unlabeled training data.)
Experimental Evaluation of TRI
[Schuurmans & Southey, MLJ 2002]

• Use it to select degree of polynomial for regression
• Compare to alternatives such as cross validation, structural risk minimization, …

Figure 5: Target functions used in the polynomial curve fitting experiments (in order): step$(x \geq 0.5)$, sin$(1/x)$, sin$^2(2\pi x)$, and a fifth degree polynomial.
Generated \( y \) values contain zero mean Gaussian noise

\[ Y = f(x) + \varepsilon \]

Figure 4: An example of minimum squared error polynomials of degrees 1, 2, and 9 for a set of 10 training points. The large degree polynomial demonstrates erratic behavior off the training set.
**Approximation ratio:**

<table>
<thead>
<tr>
<th></th>
<th>true error of selected hypothesis</th>
<th>true error of best hypothesis considered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>approximated f + noise</td>
<td>approximated f + noise</td>
</tr>
</tbody>
</table>

**Results using 200 unlabeled, t labeled**: 

Cross validation (Ten-fold)

Structural risk minimization

<table>
<thead>
<tr>
<th>t = 20</th>
<th>TRI</th>
<th>CVT</th>
<th>SRM</th>
<th>RIC</th>
<th>GCV</th>
<th>BIC</th>
<th>AIC</th>
<th>FPE</th>
<th>ADJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>1.00</td>
<td>1.06</td>
<td>1.14</td>
<td>7.54</td>
<td>5.47</td>
<td>15.2</td>
<td>22.2</td>
<td>25.8</td>
<td>1.02</td>
</tr>
<tr>
<td>50</td>
<td>1.06</td>
<td>1.17</td>
<td>1.39</td>
<td>224</td>
<td>118</td>
<td>394</td>
<td>585</td>
<td>590</td>
<td>1.12</td>
</tr>
<tr>
<td>75</td>
<td>1.17</td>
<td>1.42</td>
<td>3.62</td>
<td>5.8e3</td>
<td>3.9e3</td>
<td>9.8e3</td>
<td>1.2e4</td>
<td>1.2e4</td>
<td>1.24</td>
</tr>
<tr>
<td>95</td>
<td>1.44</td>
<td>6.75</td>
<td>56.1</td>
<td>6.1e5</td>
<td>3.7e5</td>
<td>7.8e5</td>
<td>9.2e5</td>
<td>8.2e5</td>
<td>1.54</td>
</tr>
<tr>
<td>100</td>
<td>2.41</td>
<td>1.1e4</td>
<td>2.2e4</td>
<td>1.5e8</td>
<td>6.5e7</td>
<td>1.5e8</td>
<td>1.5e8</td>
<td>8.2e7</td>
<td>3.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>t = 30</th>
<th>TRI</th>
<th>CVT</th>
<th>SRM</th>
<th>RIC</th>
<th>GCV</th>
<th>BIC</th>
<th>AIC</th>
<th>FPE</th>
<th>ADJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>1.00</td>
<td>1.08</td>
<td>1.17</td>
<td>4.69</td>
<td>1.51</td>
<td>5.41</td>
<td>5.45</td>
<td>2.72</td>
<td>1.06</td>
</tr>
<tr>
<td>50</td>
<td>1.08</td>
<td>1.17</td>
<td>1.54</td>
<td>34.8</td>
<td>9.19</td>
<td>39.6</td>
<td>40.8</td>
<td>19.1</td>
<td>1.14</td>
</tr>
<tr>
<td>75</td>
<td>1.19</td>
<td>1.37</td>
<td>9.68</td>
<td>258</td>
<td>91.3</td>
<td>266</td>
<td>266</td>
<td>159</td>
<td>1.25</td>
</tr>
<tr>
<td>95</td>
<td>1.45</td>
<td>6.11</td>
<td>419</td>
<td>4.7e3</td>
<td>2.7e3</td>
<td>4.8e3</td>
<td>5.1e3</td>
<td>4.0e3</td>
<td>1.51</td>
</tr>
<tr>
<td>100</td>
<td>2.18</td>
<td>643</td>
<td>1.6e7</td>
<td>1.6e7</td>
<td>1.6e7</td>
<td>1.6e7</td>
<td>1.6e7</td>
<td>1.6e7</td>
<td>2.10</td>
</tr>
</tbody>
</table>

Table 1: Fitting $f(x) = \text{step}(x \geq 0.5)$ with $P_x = U(0, 1)$ and $\sigma = 0.05$. Tables give distribution of approximation ratios achieved at training sample size $t = 20$ and $t = 30$, showing percentiles of approximation ratios achieved in 1000 repeated trials.
<table>
<thead>
<tr>
<th>$t = 20$</th>
<th>TRI</th>
<th>CVT</th>
<th>SRM</th>
<th>RIC</th>
<th>GCV</th>
<th>BIC</th>
<th>AIC</th>
<th>FPE</th>
<th>ADJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>2.04</td>
<td>1.03</td>
<td>1.00</td>
<td>1.00</td>
<td>1.06</td>
<td>1.00</td>
<td>1.01</td>
<td>1.58</td>
<td>1.02</td>
</tr>
<tr>
<td>50</td>
<td>3.11</td>
<td>1.37</td>
<td>1.33</td>
<td>1.34</td>
<td>1.94</td>
<td>1.35</td>
<td>1.61</td>
<td>18.2</td>
<td>1.32</td>
</tr>
<tr>
<td>75</td>
<td>3.87</td>
<td>2.23</td>
<td>2.30</td>
<td>2.13</td>
<td>10.0</td>
<td>2.75</td>
<td>4.14</td>
<td>1.2e3</td>
<td>1.83</td>
</tr>
<tr>
<td>95</td>
<td>5.11</td>
<td>9.45</td>
<td>8.84</td>
<td>8.26</td>
<td>5.0e3</td>
<td>11.8</td>
<td>82.9</td>
<td>1.8e5</td>
<td>3.94</td>
</tr>
<tr>
<td>100</td>
<td>8.92</td>
<td>105</td>
<td>526</td>
<td>105</td>
<td>2.0e7</td>
<td>2.1e3</td>
<td>2.7e5</td>
<td>2.4e7</td>
<td>6.30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$t = 30$</th>
<th>TRI</th>
<th>CVT</th>
<th>SRM</th>
<th>RIC</th>
<th>GCV</th>
<th>BIC</th>
<th>AIC</th>
<th>FPE</th>
<th>ADJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>1.50</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.02</td>
<td>1.01</td>
</tr>
<tr>
<td>50</td>
<td>3.51</td>
<td>1.16</td>
<td>1.03</td>
<td>1.05</td>
<td>1.11</td>
<td>1.02</td>
<td>1.08</td>
<td>1.45</td>
<td>1.27</td>
</tr>
<tr>
<td>75</td>
<td>4.15</td>
<td>1.64</td>
<td>1.45</td>
<td>1.48</td>
<td>2.02</td>
<td>1.39</td>
<td>1.88</td>
<td>6.44</td>
<td>1.60</td>
</tr>
<tr>
<td>95</td>
<td>5.51</td>
<td>5.21</td>
<td>5.06</td>
<td>4.21</td>
<td>26.4</td>
<td>5.01</td>
<td>19.9</td>
<td>295</td>
<td>3.02</td>
</tr>
<tr>
<td>100</td>
<td>9.75</td>
<td>124</td>
<td>1.4e3</td>
<td>20.0</td>
<td>9.1e3</td>
<td>28.4</td>
<td>9.4e3</td>
<td>1.0e4</td>
<td>8.35</td>
</tr>
</tbody>
</table>

Table 4: Fitting $f(x) = \sin^2(2\pi x)$ with $P_x = U(0, 1)$ and $\sigma = 0.05$. Tables give distribution of approximation ratios achieved at training sample size $t = 20$ and $t = 30$, showing percentiles of approximation ratios achieved in 1000 repeated trials.
Proposition 1 Let $h_m$ be the optimal hypothesis in the sequence $h_0, h_1, \ldots$ (that is, $h_m = \arg\min_{h_k} d(h_k, P_{Y|X})$) and let $h_\ell$ be the hypothesis selected by TRI. If (i) $m \leq \ell$ and (ii) $d(h_m, P_{Y|X}) \leq d(h_m, P_{Y|X})$ then

$$d(h_\ell, P_{Y|X}) \leq 3d(h_m, P_{Y|X})$$

(6)
Extension to TRI:
Adjust for expected bias of training data estimates

[Schuurmans & Southey, MLJ 2002]

Procedure ADJ

- Given hypothesis sequence $h_0, h_1, ...$
- For each hypothesis $h_\ell$ in the sequence
  - multiply its estimated distance to the target $d(h_\ell, \widehat{P}_{Y|X})$ by the worst ratio of unlabeled and labeled distance to some predecessor $h_k$ to obtain an adjusted distance estimate $d(h_\ell, \widehat{P}_{Y|X}) = d(h_\ell, \widehat{P}_{Y|X}) \frac{d(h_k, h_\ell)}{d(h_k, h_\ell)}$.
- Choose the hypothesis $h_n$ with the smallest adjusted distance $d(h_n, \widehat{P}_{Y|X})$.

Experimental results: averaged over multiple target functions, outperforms TRI
Summary

- Unlabeled data provides unbiased estimate of how often two hypotheses disagree
- Use this to identify suspiciously low disagreement over labeled training data overfitting

Different use of unlabeled data $U$

Can use $U \rightarrow \hat{P}(X)$ to alter optimization problem

- Wish to find
  \[
  \hat{f} \leftarrow \underset{h \in H}{\text{argmin}} \sum_{x \in X} \delta(h(x) \neq f(x))P(x)
  \]

- Often approximate as
  \[
  \hat{f} \leftarrow \underset{h \in H}{\text{argmin}} \frac{1}{|L|} \sum_{(x,y) \in L} \delta(h(x) \neq y)
  \]

- Can use $U$ for improved approximation:
  \[
  \hat{f} \leftarrow \underset{h \in H}{\text{argmin}} \sum_{x \in X} \delta(h(x) \neq f(x)) \frac{n(x,L)}{|L|} + \frac{n(x,U)}{|U|}
  \]