Evaluating Hypotheses

Why bother?

- Want to decide whether or not to use it.
- Integral part of many learning algorithms, e.g., post-pruning.
- Clients want to know the accuracy of the learned hypothesis.

Given only a limited set of data, two key difficulties arise:

**Bias in the estimate:** Accuracy on training data is an optimistically biased estimate of the accuracy over future examples. Estimate accuracy on blind test set.

**Variance in the estimate:** Accuracy can vary from true accuracy depending on the makeup of the test examples.

---

Slide CS478–1

---

Evaluating Hypotheses

1. Methods for evaluating learned hypotheses
2. Methods for comparing the accuracy of two hypotheses
3. Methods for comparing the accuracy of two learning algorithms

---

Slide CS478–2
Definitions

$X$: space of possible instances

$\mathcal{D}$: unknown probability distribution that defines the probability of encountering each instance in $X$.

$f$: target concept/function

$H$: hypothesis space

$h$: hypothesis in $H$

---

Two Definitions of Error

The true error of hypothesis $h$ with respect to target function $f$ and distribution $\mathcal{D}$ is the probability that $h$ will misclassify an instance drawn at random according to $\mathcal{D}$.

$$ error_{\mathcal{D}}(h) = \Pr_{x \in \mathcal{D}}[f(x) \neq h(x)] $$

The sample error of $h$ with respect to target function $f$ and data sample $S$ is the proportion of examples $h$ misclassifies.

$$ error_S(h) = \frac{1}{n} \sum_{x \in S} \delta(f(x) \neq h(x)) $$

Where $\delta(f(x) \neq h(x))$ is 1 if $f(x) \neq h(x)$, and 0 otherwise.

How well does $error_S(h)$ estimate $error_{\mathcal{D}}(h)$?
Example

Hypothesis $h$ misclassifies 12 of the 40 examples in $S$

$$\text{error}_S(h) = \frac{12}{40} = .30$$

How good an estimate of $\text{error}_D(h)$ is $\text{error}_S(h)$?

Confidence Intervals

If

- $S$ contains $n$ examples, drawn independently of $h$ and each other
- $n \geq 30$

Then

- With approximately 95% probability, $\text{error}_D(h)$ lies in interval

$$\text{error}_S(h) \pm 1.96 \sqrt{\frac{\text{error}_S(h)(1 - \text{error}_S(h))}{n}}$$

95% confidence interval estimate: $0.30 \pm (1.96)(0.07) = 0.30 \pm 0.14$. 
Confidence Intervals

If (1) \( S \) contains \( n \) examples, drawn independently of \( h \) and each other, and (2) \( n \geq 30 \), then

- With approximately \( N\% \) probability, \( \text{error}_D(h) \) lies in
  interval

\[
\text{error}_S(h) \pm z_N \sqrt{\frac{\text{error}_S(h)(1 - \text{error}_S(h))}{n}}
\]

where

<table>
<thead>
<tr>
<th>( N% )</th>
<th>50%</th>
<th>68%</th>
<th>80%</th>
<th>90%</th>
<th>95%</th>
<th>98%</th>
<th>99%</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z_N )</td>
<td>0.67</td>
<td>1.00</td>
<td>1.28</td>
<td>1.64</td>
<td>1.96</td>
<td>2.33</td>
<td>2.58</td>
</tr>
</tbody>
</table>

Comparing Hypotheses

Test \( h_1 \) on sample \( S_1 \), test \( h_2 \) on \( S_2 \)

- Given \( h_1 \) and \( h_2 \), we can determine whether the difference in their error rates is meaningful or not.

\[
d = \text{error}_D(h_1) - \text{error}_D(h_2)
\]

- Estimator is the difference between the sample errors:

\[
\hat{d} = \text{error}_{S_1}(h_1) - \text{error}_{S_2}(h_2)
\]
• The variance of this distribution is the sum of the variances of $error_{S_1}(h_1)$ and $error_{S_2}(h_2)$:

$$
\sigma_d^2 \approx \frac{error_{S_1}(h_1)(1 - error_{S_1}(h_1))}{n_1} + \frac{error_{S_2}(h_2)(1 - error_{S_2}(h_2))}{n_2}
$$

• Can compute confidence interval estimate for $d$:

$$
d \pm z_N \sqrt{\frac{error_{S_1}(h_1)(1 - error_{S_1}(h_1))}{n_1} + \frac{error_{S_2}(h_2)(1 - error_{S_2}(h_2))}{n_2}}
$$

Slide CS478–9

---

**Comparing Learning Algorithms**

• We are often interested in comparing the performance of two learning algorithms, $L_A$ and $L_B$, instead of two specific hypotheses.

• Ideally, we’d like to measure the expected value of the difference in their error:

$$
E_{S \subset D}[error_D(L_A(S)) - error_D(L_B(S))]
$$

where $L(S)$ is the hypothesis output by learner $L$ using training set $S$ from distribution $D$.

• To estimate this difference, we need to average results over many different training and testing sets.

Slide CS478–10
K-fold Cross Validation

1. Partition data $D_0$ into $k$ disjoint test sets $T_1, T_2, \ldots, T_k$ of equal size, where this size is at least 30.

2. For $i$ from 1 to $k$
   use $T_i$ for the test set, and the remaining data for training set $S_i$
   
   $S_i \leftarrow \{D_0 - T_i\}$
   $h_A \leftarrow L_A(S_i), h_B \leftarrow L_B(S_i)$
   $\delta_i \leftarrow \text{error}_{T_i}(h_A) - \text{error}_{T_i}(h_B)$

3. Return the average difference in error: $\tilde{\delta} = \frac{1}{k} \sum_{i=1}^{k} \delta_i$

Slide CS478–11

McNemar’s Test

- For each example $x \in T$ (test set), record how it was classified.
- Construct the following contingency table:

\[
\begin{array}{cc}
   n_{00} & n_{01} \\
   n_{10} & n_{11}
\end{array}
\]

where

Slide CS478–12
- \( n_{00} \) = number of examples misclassified by both \( L_A \) and \( L_B \).
- \( n_{01} \) = number of examples misclassified by \( L_A \), but not by \( L_B \).
- \( n_{10} \) = number of examples misclassified by \( L_B \), but not by \( L_A \).
- \( n_{00} \) = number of examples misclassified by neither \( L_A \) nor \( L_B \).

McNemar’s test is based on a \( \chi^2 \) test for goodness-of-fit that compares the distribution of the observed counts to the counts expected when the learning algorithms have the the same performance.

Slide CS478–13

McNemar’s Test

Contingency table:

<table>
<thead>
<tr>
<th></th>
<th>( n_{00} )</th>
<th>( n_{01} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_{10} )</td>
<td>( n_{11} )</td>
<td></td>
</tr>
</tbody>
</table>

Expected counts:

<table>
<thead>
<tr>
<th></th>
<th>( n_{00} )</th>
<th>( (n_{01} + n_{10})/2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (n_{01} + n_{10})/2 )</td>
<td>( n_{11} )</td>
<td></td>
</tr>
</tbody>
</table>

If \( \frac{(n_{00} - n_{10})^2}{n_{00} + n_{10}} \) is greater than 3.841459, then the difference in error between \( L_A \) and \( L_B \) is statistically significant at or above the 95\% confidence level.

Slide CS478–14
Summary

- Statistical analysis is important to compare empirical learning results.
- No single procedure for comparing learning methods based on limited data satisfies all the constraints we would like.
- Statistical models rarely fit perfectly the practical constraints in testing learning algorithms when available data is limited.
- They do provide approximate confidence intervals that can be of great help in interpreting experimental comparisons of learning methods.