Evaluating Machine Learning Methods: Part 1

CS 760@UW-Madison
Goals for the lecture

you should understand the following concepts

- bias of an estimator
- learning curves
- stratified sampling
- cross validation
- confusion matrices
- TP, FP, TN, FN
- ROC curves
Goals for the next lecture

you should understand the following concepts

• PR curves
• confidence intervals for error
• pairwise $t$-tests for comparing learning systems
• scatter plots for comparing learning systems
• lesion studies
Bias of an estimator

\[ \hat{\theta} = E[\hat{\theta}] - \theta \]

e.g. polling methodologies often have an inherent bias

<table>
<thead>
<tr>
<th>POLLSTER</th>
<th>LIVE CALLER WITH CELLPHONES</th>
<th>INTERNET</th>
<th>NCPP/ AAPOR/ ROPER</th>
<th>POLLS ANALYZED</th>
<th>SIMPLE AVERAGE ERROR</th>
<th>RACES CALLED CORRECTLY</th>
<th>ADVANCED +/-</th>
<th>PREDICTIVE +/-</th>
<th>538 GRADE</th>
<th>BANNED BY 538</th>
<th>MEAN-REVERTED BIAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SurveyUSA</td>
<td></td>
<td></td>
<td></td>
<td>763</td>
<td>4.6</td>
<td>90%</td>
<td>-1.0</td>
<td>-0.8</td>
<td>A</td>
<td>D+0.1</td>
<td></td>
</tr>
<tr>
<td>YouGov</td>
<td></td>
<td></td>
<td></td>
<td>707</td>
<td>6.7</td>
<td>93%</td>
<td>-0.3</td>
<td>+0.1</td>
<td>B</td>
<td>D+1.6</td>
<td></td>
</tr>
<tr>
<td>Rasmussen Reports/ Pulse Opinion Research</td>
<td></td>
<td></td>
<td></td>
<td>657</td>
<td>5.3</td>
<td>79%</td>
<td>+0.4</td>
<td>+0.7</td>
<td>C+</td>
<td>R+2.0</td>
<td></td>
</tr>
<tr>
<td>Zogby Interactive/JZ Analytics</td>
<td></td>
<td></td>
<td></td>
<td>465</td>
<td>5.6</td>
<td>78%</td>
<td>+0.8</td>
<td>+1.2</td>
<td>C-</td>
<td>R+0.8</td>
<td></td>
</tr>
<tr>
<td>Mason-Dixon Polling &amp; Research, Inc.</td>
<td></td>
<td></td>
<td></td>
<td>415</td>
<td>5.2</td>
<td>86%</td>
<td>-0.4</td>
<td>-0.2</td>
<td>B+</td>
<td>R+1.0</td>
<td></td>
</tr>
<tr>
<td>Public Policy Polling</td>
<td></td>
<td></td>
<td></td>
<td>383</td>
<td>4.9</td>
<td>82%</td>
<td>-0.5</td>
<td>-0.1</td>
<td>B+</td>
<td>R+0.2</td>
<td></td>
</tr>
<tr>
<td>Research 2000</td>
<td></td>
<td></td>
<td></td>
<td>279</td>
<td>5.5</td>
<td>88%</td>
<td>+0.2</td>
<td>+0.6</td>
<td>F</td>
<td>×</td>
<td>D+1.4</td>
</tr>
</tbody>
</table>
How can we get an unbiased estimate of the accuracy of a learned model?
Test sets revisited

How can we get an unbiased estimate of the accuracy of a learned model?

- when learning a model, you should pretend that you don’t have the test data yet (it is “in the mail”)

- if the test-set labels influence the learned model in any way, accuracy estimates will be biased
Learning curves

How does the accuracy of a learning method change as a function of the training-set size?

This can be assessed by plotting learning curves.

Figure from Perlich et al. *Journal of Machine Learning Research*, 2003
Learning curves

given training/test set partition

• for each sample size \( s \) on learning curve
  • (optionally) repeat \( n \) times
    • randomly select \( s \) instances from training set
    • learn model
    • evaluate model on test set to determine accuracy \( a \)
  • plot \((s, a)\) or \((s, \text{avg. accuracy and error bars})\)
Limitations of a single training/test partition

• we may not have enough data to make sufficiently large training and test sets
  • a larger test set gives us more reliable estimate of accuracy (i.e. a lower variance estimate)
  • but… a larger training set will be more representative of how much data we actually have for learning process

• a single training set doesn’t tell us how sensitive accuracy is to a particular training sample
Using multiple training/test partitions

• two general approaches for doing this
  • random resampling
  • cross validation
Random resampling

We can address the second issue by repeatedly randomly partitioning the available data into training and test sets.
Stratified sampling

When randomly selecting training or validation sets, we may want to ensure that class proportions are maintained in each selected set.

This can be done via stratified sampling: first stratify instances by class, then randomly select instances from each class proportionally.
Cross validation

Partition data into \( n \) subsamples

Iteratively leave one subsample out for the test set, train on the rest

<table>
<thead>
<tr>
<th>iteration</th>
<th>train on</th>
<th>test on</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( s_2, s_3, s_4, s_5 )</td>
<td>( s_1 )</td>
</tr>
<tr>
<td>2</td>
<td>( s_1, s_3, s_4, s_5 )</td>
<td>( s_2 )</td>
</tr>
<tr>
<td>3</td>
<td>( s_1, s_2, s_4, s_5 )</td>
<td>( s_3 )</td>
</tr>
<tr>
<td>4</td>
<td>( s_1, s_2, s_3, s_5 )</td>
<td>( s_4 )</td>
</tr>
<tr>
<td>5</td>
<td>( s_1, s_2, s_3, s_4 )</td>
<td>( s_5 )</td>
</tr>
</tbody>
</table>
Suppose we have 100 instances, and we want to estimate accuracy with cross validation.

<table>
<thead>
<tr>
<th>iteration</th>
<th>train on</th>
<th>test on</th>
<th>correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>s_2 s_3 s_4 s_5</td>
<td>s_1</td>
<td>11 / 20</td>
</tr>
<tr>
<td>2</td>
<td>s_1 s_3 s_4 s_5</td>
<td>s_2</td>
<td>17 / 20</td>
</tr>
<tr>
<td>3</td>
<td>s_1 s_2 s_4 s_5</td>
<td>s_3</td>
<td>16 / 20</td>
</tr>
<tr>
<td>4</td>
<td>s_1 s_2 s_3 s_5</td>
<td>s_4</td>
<td>13 / 20</td>
</tr>
<tr>
<td>5</td>
<td>s_1 s_2 s_3 s_4</td>
<td>s_5</td>
<td>16 / 20</td>
</tr>
</tbody>
</table>

accuracy = 73/100 = 73%
10-fold cross validation is common, but smaller values of $n$ are often used when learning takes a lot of time.

In leave-one-out cross validation, $n = \# \text{ instances}$.

In stratified cross validation, stratified sampling is used when partitioning the data.

CV makes efficient use of the available data for testing.

Note that whenever we use multiple training sets, as in CV and random resampling, we are evaluating a learning method as opposed to an individual learned hypothesis.
Confusion matrices

How can we understand what types of mistakes a learned model makes?

**task: activity recognition from video**

![Confusion Matrix](vision.jhu.edu)

- **actual class**
- **predicted class**

figure from vision.jhu.edu
### Confusion Matrix for 2-Class Problems

The confusion matrix for 2-class problems is a table that categorizes predictions into true positives, false positives, false negatives, and true negatives. The matrix is structured as follows:

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Predicted</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Positive</strong></td>
<td>TP (TP)</td>
<td>FP (FN)</td>
</tr>
<tr>
<td><strong>Negative</strong></td>
<td>FN (FP)</td>
<td>TN (TN)</td>
</tr>
</tbody>
</table>

Where:
- **TP** (True Positives): Correctly predicted positive class.
- **FP** (False Positives): Incorrectly predicted positive class.
- **FN** (False Negatives): Incorrectly predicted negative class.
- **TN** (True Negatives): Correctly predicted negative class.

**Accuracy** is defined as:

$$\text{accuracy} = \frac{\text{TP + TN}}{\text{TP + FP + FN + TN}}$$

**Error** is defined as:

$$\text{error} = 1 - \text{accuracy} = 1 - \frac{\text{TP + TN}}{\text{TP + FP + FN + TN}}$$
Is accuracy an adequate measure of predictive performance?

accuracy may not be useful measure in cases where

• there is a large class skew
  • Is 98% accuracy good when 97% of the instances are negative?

• there are differential misclassification costs – say, getting a positive wrong costs more than getting a negative wrong
  • Consider a medical domain in which a false positive results in an extraneous test but a false negative results in a failure to treat a disease

• we are most interested in a subset of high-confidence predictions
### Other accuracy metrics

- **True positives** ($TP$)
- **True negatives** ($TN$)
- **False positives** ($FP$)
- **False negatives** ($FN$)

#### Table

<table>
<thead>
<tr>
<th>Predicted Class</th>
<th>Actual Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>False negatives ($FN$)</td>
</tr>
<tr>
<td>Negative</td>
<td>True negatives ($TN$)</td>
</tr>
</tbody>
</table>
Other accuracy metrics

<table>
<thead>
<tr>
<th>predicted class</th>
<th>actual class</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive</td>
<td>true positives (TP)</td>
</tr>
<tr>
<td>negative</td>
<td>false negatives (FN)</td>
</tr>
</tbody>
</table>

true positive rate (recall) = \[ \frac{TP}{\text{actual pos}} = \frac{TP}{TP + FN} \]
Other accuracy metrics

true positive rate (recall)  = \[ \frac{TP}{TP + FN} \]

false positive rate  = \[ \frac{FP}{TN + FP} \]
A Receiver Operating Characteristic (ROC) curve plots the TP-rate vs. the FP-rate as a threshold on the confidence of an instance being positive is varied.

Different methods can work better in different parts of ROC space.
Algorithm for creating an ROC curve

let \( \{ (y^{(1)}, c^{(1)}) \ldots (y^{(m)}, c^{(m)}) \} \) be the test-set instances sorted according to predicted confidence that each instance is positive.

let \( num\_neg, num\_pos \) be the number of negative/positive instances in the test set.

\( TP = 0, \quad FP = 0 \)

\( last\_TP = 0 \)

for \( i = 1 \) to \( m \)

// find thresholds where there is a pos instance on high side, neg instance on low side

if \( (i > 1) \) and \( (c^{(i)} \neq c^{(i-1)}) \) and \( (y^{(i)} == \text{neg}) \) and \( (TP > last\_TP) \)

\( FPR = FP / num\_neg, \quad TPR = TP / num\_pos \)

output \((FPR, TPR)\) coordinate

\( last\_TP = TP \)

if \( y^{(i)} == \text{pos} \)

++\( TP \)

else

++\( FP \)

\( FPR = FP / num\_neg, \quad TPR = TP / num\_pos \)

output \((FPR, TPR)\) coordinate.
Plotting an ROC curve

<table>
<thead>
<tr>
<th>instance</th>
<th>confidence</th>
<th>correct class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex 9</td>
<td>.99</td>
<td>+</td>
</tr>
<tr>
<td>Ex 7</td>
<td>.98</td>
<td>+</td>
</tr>
<tr>
<td>Ex 1</td>
<td>.72</td>
<td>-</td>
</tr>
<tr>
<td>Ex 2</td>
<td>.70</td>
<td>+</td>
</tr>
<tr>
<td>Ex 6</td>
<td>.65</td>
<td>+</td>
</tr>
<tr>
<td>Ex 10</td>
<td>.51</td>
<td>-</td>
</tr>
<tr>
<td>Ex 3</td>
<td>.39</td>
<td>-</td>
</tr>
<tr>
<td>Ex 5</td>
<td>.24</td>
<td>+</td>
</tr>
<tr>
<td>Ex 4</td>
<td>.11</td>
<td>-</td>
</tr>
<tr>
<td>Ex 8</td>
<td>.01</td>
<td>-</td>
</tr>
</tbody>
</table>

True positive rate vs. False positive rate

TPR = True Positive Rate
FPR = False Positive Rate
ROC curve example

task: recognizing genomic units called operons

figure from Bockhorst et al., Bioinformatics 2003
The best operating point depends on the relative costs of FN and FP misclassifications.

- Best operating point when FN costs $10 \times$ FP
- Best operating point when cost of misclassifying positives and negatives is equal
- Best operating point when FP costs $10 \times$ FN
THANK YOU

Some of the slides in these lectures have been adapted/borrowed from materials developed by Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Elad Hazan, Tom Dietterich, and Pedro Domingos.