CS639: Data Management for Data Science

Lecture 17: Evaluating Machine Learning Methods

Theodoros Rekatsinas
Announcements

1. You can see your exams during office hours

2. Homework will be announced later this week; we will have only two more projects not three
Today

1. Evaluating ML models
How can we get an unbiased estimate of the accuracy of a learned model?
Test sets

• 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

• 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 learning curves
Learning curves

• Given a training/test set partition
  • For each sample size s on the learning curve
    • (optionally) repeat n times
    • Randomly select s instances from the training set
    • Learn the model
    • Evaluate the model on the test set to determine accuracy a
  • Plot (s,a)
Validation (tuning) sets

• Suppose we want unbiased estimates of accuracy during the learning process (e.g. to choose the best level of decision-tree pruning)?

Partition training data into separate training/validation sets
Limitations of using 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 estimates 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 does not tell us how sensitive accuracy is to a particular training sample
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>
Cross validation example

- 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 = $\frac{73}{100} = 73\%$
Cross validation example

• 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 model
Internal cross validation

• Instead of a single validation set, we can use cross-validation within a training set to select a model (e.g. to choose the best level of decision-tree pruning)
Confusion matrices

- How can we understand what types of mistakes a learned model makes?
Confusion matrix for 2-class problems

<table>
<thead>
<tr>
<th></th>
<th>Actual Class</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Predicted</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Class</strong></td>
<td><strong>Positive</strong></td>
</tr>
<tr>
<td><strong>Positive</strong></td>
<td><strong>True Positives</strong> (TP)</td>
</tr>
<tr>
<td><strong>Negative</strong></td>
<td><strong>False Negatives</strong> (FN)</td>
</tr>
</tbody>
</table>

**Accuracy** = \[
\frac{TP + TN}{TP + FP + FN + TN}
\]
Is accuracy an adequate measure of predictive performance?

- Accuracy may not be a useful measure in cases where:
  - There is a large class skew
    - Is 98% accuracy good if 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

<table>
<thead>
<tr>
<th></th>
<th>actual class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>positive</td>
</tr>
<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) = \( \frac{TP}{\text{actual pos}} = \frac{TP}{TP + FN} \)

false positive rate = \( \frac{FP}{\text{actual neg}} = \frac{FP}{TN + FP} \)
ROC curves

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. This depends on cost of false + vs. false -
ROC curve example
ROC curves and misclassification costs

Thyroid anomaly detection

- Best operating point when FN costs 10x FP
- Best operating point when cost of misclassifying positives and negatives is equal
- Best operating point when FP costs 10x FN
Algorithm for creating an ROC curve

1. sort test-set predictions according to confidence that each instance is positive

2. step through sorted list from high to low confidence
   i. locate a *threshold* between instances with opposite classes (keeping instances with the same confidence value on the same side of threshold)
   ii. compute TPR, FPR for instances above threshold
   iii. output (FPR, TPR) coordinate
Other accuracy metrics

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

predicted class

- recall (TP rate) = \( \frac{TP}{\text{actual pos}} = \frac{TP}{TP + FN} \)
- precision = \( \frac{TP}{\text{predicted pos}} = \frac{TP}{TP + FP} \)
Precision/recall curves

A precision/recall curve plots the precision vs. recall (TP-rate) as a threshold on the confidence of an instance being positive is varied.

- Ideal point
- Default precision determined by the fraction of instances that are positive
To Avoid Cross-Validation Pitfalls

1. Is my held-aside test data really representative of going out to collect new data?
   - Even if your methodology is fine, someone may have collected features for positive examples differently than for negatives – should be randomized
   - Example: samples from cancer processed by different people or on different days than samples for normal controls
To Avoid Cross-Validation Pitfalls

• 2. Did I repeat my entire data processing procedure on every fold of cross-validation, using only the training data for that fold?
  – On each fold of cross-validation, did I ever access in any way the label of a test case?
  – Any preprocessing done over entire data set (feature selection, parameter tuning, threshold selection) must not use labels
To Avoid Cross-Validation Pitfalls

3. Have I modified my algorithm so many times, or tried so many approaches, on this same data set that I (the human) am overfitting it?
   - Have I continually modified my preprocessing or learning algorithm until I got some improvement on this data set?
   - If so, I really need to get some additional data now to at least test on
Ablation Studies

We can gain insight into what contributes to a learning system’s performance by removing (lesioning) components of it.

The ROC curves here show how performance is affected when various feature types are removed from the learning representation.