Instructor: Jessica Wu -- Harvey Mudd College

The instructor gratefully acknowledges Eric Eaton (UPenn), David Kauchak (Pomona), and the many others who made their course materials freely available online.

Experimental Procedure
Learning Goals

- Describe how cross-validation ($k$-fold, leave-one-out) is used to evaluate model and optimize hyperparameters
- Describe how to compare models statistically using the $t$-test
- Describe how bootstrapping is used to evaluate test performance
Proper Evaluation?

Current plan
- Learn algorithm on training data (subset of full data)
- Evaluate on test data (subset of full data)
- Repeat until happy with results

Is this okay?
- No! Although we are not explicitly looking at test data, we are still “cheating” by biasing our algorithm to test data
- Once you look at / use test data, it is no longer test data!

So, how can we evaluate our algorithm during development?

Classification Evaluation

Based on slide by David Kauchak

Based on slide by Eric Eaton
$k$-Fold Cross-Validation

- Why just choose one particular “split” of data?
  - in principle, we should do this multiple times since performance may be different for each split

- $k$-Fold Cross-Validation (e.g., $k = 10$)
  - randomly partition all training data of $n$ instances into $k$ disjoint subsets (each roughly of size $n/k$)
  - choose each fold in turn as validation set; train model on the other $k-1$ folds and evaluate
  - compute statistics over $k$ test performances, or choose best of $k$ models

---

Example: 3-Fold CV

- All Training Data
  - 1st Partition
    - 1st Validation Data
    - 1st Training Data
  - 2nd Partition
    - 2nd Validation Data
    - 2nd Training Data
  - 3rd Partition
    - 3rd Training Data
    - 3rd Validation Data

- learn
- learn
- learn

- report CV performance (summary statistics over $k$ performances)
- choose model with best validation performance

- Test Data
- evaluate
- report test performance

---

Based on slide by Eric Eaton
Optimizing Model Parameters

Can also use CV to choose value of model parameter $P$

- Search over space of parameter values $p \in \text{values}(P)$
  - evaluate model with $P = p$ on validation set
- Choose value $p'$ with highest validation performance
- Learn model on full training set with $P = p'$

Based on slide by Eric Eaton

Example: Comparing Models

<table>
<thead>
<tr>
<th>Split 1</th>
<th>Split 2</th>
<th>Split 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Based on slide by David Kauchak
Comparing Models

Is model 2 better than model 1?

<table>
<thead>
<tr>
<th>Sample 1</th>
<th>Sample 2</th>
<th>Sample 3</th>
<th>Sample 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>split</td>
<td>M1</td>
<td>M2</td>
<td>split</td>
</tr>
<tr>
<td>1</td>
<td>87</td>
<td>88</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>85</td>
<td>84</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>83</td>
<td>84</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>80</td>
<td>79</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>88</td>
<td>89</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>85</td>
<td>85</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>83</td>
<td>81</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>87</td>
<td>86</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>88</td>
<td>89</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>84</td>
<td>85</td>
<td>10</td>
</tr>
<tr>
<td>avg</td>
<td>85</td>
<td>85</td>
<td>avg</td>
</tr>
</tbody>
</table>

How do we decide if model 2 is better than model 1?

Statistical tests

Setup

– assume some default hypothesis about data that you would like to disprove, called the null hypothesis
– e.g. model 1 and model 2 are not statistically different in performance

Test

– calculate test statistic from data (often assuming something about data)
– calculate p-value from test statistic
  • p-value = probability of seeing test statistic at least as extreme as one actually observed given null hypothesis is true
– compare p-value to threshold $\alpha$ (significance level)
– reject null hypothesis if $p < \alpha$
– note that statistically significant difference is not necessarily a large-magnitude difference

Based on slide by David Kauchak
**t-test**

Determines whether two samples come from same underlying distribution or not

**Null hypothesis**
- model 1 and model 2 accuracies are no different, i.e. come from same distribution

**Assumptions**
- there are a number that often are not completely true, but we are often not too far off

**Our formulation**
- do “paired t-test”
  - values can be thought of as pairs, calculated under same conditions (in our case, same train/test split)
  - gives more power than unpaired t-test (we have more information)
- for almost all experiments, do “two-tailed” version
  - no *a priori* knowledge of which model is better

Based on slide by David Kauchak

---

**Comparing Models**

*Is model 2 better than model 1?*

<table>
<thead>
<tr>
<th>Sample 1</th>
<th>Sample 2</th>
<th>Sample 3</th>
<th>Sample 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>split</strong></td>
<td><strong>M1</strong></td>
<td><strong>M2</strong></td>
<td><strong>split</strong></td>
</tr>
<tr>
<td>1</td>
<td>87</td>
<td>88</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>85</td>
<td>84</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>83</td>
<td>84</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>80</td>
<td>79</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>88</td>
<td>89</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>85</td>
<td>85</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>83</td>
<td>81</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>87</td>
<td>86</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>88</td>
<td>89</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>84</td>
<td>85</td>
<td>10</td>
</tr>
<tr>
<td><strong>avg</strong></td>
<td><strong>85</strong></td>
<td><strong>85</strong></td>
<td><strong>avg</strong></td>
</tr>
<tr>
<td><strong>sdev</strong></td>
<td><strong>5.9</strong></td>
<td><strong>3.9</strong></td>
<td><strong>sdev</strong></td>
</tr>
</tbody>
</table>

Based on slide by David Kauchak
Leave-One-Out CV (LOOCV)

- Special case where $k = n$
  - Each partition now one example
  - Train using $n - 1$ examples, validate on remaining example
  - Repeat $n$ times, each with different validation example
  - Finally, choose model with smallest average validation error
- When is it used?
  - Can be expensive for large $n$, so typically used when $n$ is small
  - Useful in domains with limited training data (maximizes data used for training)

Summary: Cross-Validation

- Cross-validation generates an approximate estimate of how well the classifier will do on “unseen” data
  - as $k \rightarrow n$, model becomes more accurate (more training data)
  - ... but, CV becomes more computationally expensive (have to train $k$ models)
  - choosing $k < n$ is a compromise
- It is an even better idea to do CV repeatedly!
Multiple Trials of $k$-Fold CV

1) Loop for $t$ trials:
   a.) Randomize Data Set
   b.) Perform $k$-fold CV

2) Compute statistics over $t \times k$ validation performances

Comparing Multiple Classifiers

1) Loop for $t$ trials:
   a.) Randomize Data Set
   b.) Perform $k$-fold CV

2) Compute statistics over $t \times k$ validation performances

Based on slide by Eric Eaton
Statistical Tests on Test Data

- All Data
- All Training Data
- Training Data
- Validation Data
- Test Data

Based on slide by David Kauchak

Bootstrapping

- Given set of $n$ examples
- Sample $n$ elements from this set with replacement to create new training set
- Use set of examples not selected as validation set
- Repeat $t$ times

Based on slide by Piyush Rai
Experimentation Good Practices

Never look at your test data!

During development
  – compare different models / hyperparameters on development data
  – use cross-validation to get more consistent results
  – if you want to be confident with results, use t-test

For final evaluation, use bootstrap resampling combined with t-test to compare

Avoiding Pitfalls

• Is my held-aside test data really representative of going out to collect new data?
• Did I repeat my entire data processing procedure on every fold of cross-validation, using only training data for that fold?
• Have I modified my algorithm so many times, or tried so many approaches, on this same data set that I (human) am overfitting it?
The Short Way
(that Many People Actually Use)

• Split into only training data + validation data
• Train on training data, evaluate on validation data
• Report cross-validation performance
  – possibly also training performance

• Why is this used?
  – might not be enough data to create held-out test set
  – you cannot trust that authors did not peek at test data anyway =P