Cross-validation for detecting and preventing overfitting

Andrew W. Moore/Anna Goldenberg
School of Computer Science
Carnegie Mellon University

Want to learn model from data

Toy example

Can we learn f(x) that fits our data, y=f(x)+noise?
Which is best?

Why not choose the method with the best fit to the data?

What do we really want?

Why not choose the method with the best fit to the data?

“How well are you going to predict future data drawn from the same distribution?”
The test set method

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Learn model from the training set

(Linear function example)
The test set method

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Learn the model from the training set
4. Estimate your future performance with the test set

How to tell how good is the prediction?

MSE (Mean Squared Error)

\[ \text{MSE} = \frac{(y_{1p} - y_{1o})^2 + (y_{2p} - y_{2o})^2 + (y_{3p} - y_{3o})^2}{3} \]

(Linear function example)
**MSE (Mean Squared Error)**

In general,

\[
\text{MSE} = \frac{\sum_{i=1}^{n} (\text{prediction}_i - \text{observation}_i)^2}{n}
\]

**The test set method**

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Learn the model from the training set
4. Estimate your future performance with the test set

\[
\text{MSE} = 2.4
\]
The test set method

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform model fitting on the training set
4. Estimate your future performance with the test set

(Quadratic function)
Mean Squared Error = 0.9

The test set method

1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform model fitting on the training set
4. Estimate your future performance with the test set

(Piecewise Linear function)
Mean Squared Error = 2.2
The test set method

Good news:
• Very very simple
• Can simply choose the method with the best test-set score

Bad news:
• What’s the downside?

The test set method

Good news:
• Very very simple
• Can simply choose the method with the best test-set score

Bad news:
• Wastes data: best model is estimated using 30% less data
• If we don’t have much data, our test-set might just be lucky or unlucky

We say the “test-set estimator of performance has high variance”
LOOCV (Leave-one-out Cross Validation)

For \( k = 1 \) to \( R \), \( R = \# \) of records

1. Let \((x_k, y_k)\) be the \( k^{th} \) record

2. Temporarily remove \((x_k, y_k)\) from the dataset
LOOCV (Leave-one-out Cross Validation)

For \( k=1 \) to \( R \)

1. Let \((x_k, y_k)\) be the \( k^{th} \) record
2. Temporarily remove \((x_k, y_k)\) from the dataset
3. Train on the remaining \( R-1 \) datapoints

4. Note your error \((x_k, y_k)\)
LOOCV (Leave-one-out Cross Validation)

For $k=1$ to $R$
1. Let $(x_k, y_k)$ be the $k^{th}$ record
2. Temporarily remove $(x_k, y_k)$ from the dataset
3. Train on the remaining $R-1$ datapoints
4. Note your error $(x_k, y_k)$

When you’ve done all points, report the mean error.
LOOCV for Quadratic Function

For k=1 to R
1. Let \((x_k, y_k)\) be the \(k^{th}\) record
2. Temporarily remove \((x_k, y_k)\) from the dataset
3. Train on the remaining \(R-1\) datapoints
4. Note your error \((x_k, y_k)\)

When you’ve done all points, report the mean error.

\[ \text{MSE}_{\text{LOOCV}} = 0.962 \]

LOOCV for Join The Dots

For k=1 to R
1. Let \((x_k, y_k)\) be the \(k^{th}\) record
2. Temporarily remove \((x_k, y_k)\) from the dataset
3. Train on the remaining \(R-1\) datapoints
4. Note your error \((x_k, y_k)\)

When you’ve done all points, report the mean error.

\[ \text{MSE}_{\text{LOOCV}} = 3.33 \]
Which kind of Cross Validation?

<table>
<thead>
<tr>
<th></th>
<th>Downside</th>
<th>Upside</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test-set</td>
<td>Variance: unreliable estimate of future performance</td>
<td>Cheap</td>
</tr>
<tr>
<td>Leave-one-out</td>
<td>Expensive. Has some weird behavior</td>
<td>Doesn’t waste data</td>
</tr>
</tbody>
</table>

..can we get the best of both worlds?

k-fold Cross Validation

Randomly break the dataset into k partitions (in our example we’ll have k=3 partitions colored Red Green and Blue)
k-fold Cross Validation

Randomly break the dataset into $k$ partitions (in our example we’ll have $k=3$ partitions colored Red Green and Blue)

For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.
Randomly break the dataset into \( k \) partitions (in our example we’ll have \( k = 3 \) partitions colored Red Green and Blue)

For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

For the blue partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.

Then report the mean error

\[
MSE_{3\text{FOLD}} = 2.05
\]
k-fold Cross Validation

Randomly break the dataset into $k$ partitions (in our example we’ll have $k=3$ partitions colored Red Green and Blue)

For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

For the green partition: Train on all the points not in the green partition. Find the test-set sum of errors on the green points.

For the blue partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.

Then report the mean error

Quadratic model
$MSE_{3\text{FOLD}}=1.11$

Piecewise linear model
$MSE_{3\text{FOLD}}=2.93$
### Which kind of Cross Validation?

<table>
<thead>
<tr>
<th></th>
<th>Downside</th>
<th>Upside</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test-set</strong></td>
<td>Variance: unreliable estimate of future performance</td>
<td>Cheap</td>
</tr>
<tr>
<td><strong>Leave-one-out</strong></td>
<td>Expensive. Has some weird behavior</td>
<td>Doesn’t waste data</td>
</tr>
<tr>
<td><strong>10-fold</strong></td>
<td>Wastes 10% of the data. 10 times more expensive than test set</td>
<td>Only wastes 10%. Only 10 times more expensive instead of R times.</td>
</tr>
<tr>
<td><strong>3-fold</strong></td>
<td>Wastier than 10-fold. Expensivier than test set</td>
<td>Slightly better than test-set</td>
</tr>
<tr>
<td><strong>R-fold</strong></td>
<td>Identical to Leave-one-out</td>
<td></td>
</tr>
</tbody>
</table>

Copyright © 2001, Andrew W. Moore

Cross-Validation: Slide 31

---

### Which kind of Cross Validation?

<table>
<thead>
<tr>
<th></th>
<th>Downside</th>
<th>Upside</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Test-set</strong></td>
<td>Variance: unreliable estimate of future performance</td>
<td>Cheap</td>
</tr>
<tr>
<td><strong>Leave-one-out</strong></td>
<td>Expensive. Has some weird behavior</td>
<td>Doesn’t waste data</td>
</tr>
<tr>
<td><strong>10-fold</strong></td>
<td>Wastes 10% of the data. 10 times more expensive than test set</td>
<td>Only wastes 10%. Only 10 times more expensive instead of R times.</td>
</tr>
<tr>
<td><strong>3-fold</strong></td>
<td>Wastier than 10-fold. Expensivier than test set</td>
<td>Slightly better than test-set</td>
</tr>
<tr>
<td><strong>R-fold</strong></td>
<td>Identical to Leave-one-out</td>
<td></td>
</tr>
</tbody>
</table>

But note: can use algorithmic tricks to make these cheap

Copyright © 2001, Andrew W. Moore

Cross-Validation: Slide 32
CV-based Model Selection

- We’re trying to decide which algorithm to use.
- We train each machine and make a table...

<table>
<thead>
<tr>
<th>i</th>
<th>f_i</th>
<th>TRAINERR</th>
<th>10-FOLD-CV-ERR</th>
<th>Choice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>f_1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>f_2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>f_3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>f_4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>f_5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>f_6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alternatives to CV-based model selection

- Model selection methods:
  
  - AIC (Akaike Information Criterion) — asymptotically equivalent to LOOCV
  - BIC (Bayesian Information Criterion) — used for finding best structure rather than for classification, asymptotically equivalent to carefully chosen k-fold
  - VC-dimension (Vapnik-Chervonenkis Dimension) — only directly applicable to choosing classifiers, wildly conservative

  Their advantage over CV — you only need the training error
  Many alternatives---including proper Bayesian approaches
Other Cross-validation

- Can do “leave all pairs out” or “leave-all-n-tuples-out” if feeling resourceful.

- k-folds where each fold is an independently-chosen subset of the data

Cross-Validation is useful

- Preventing overfitting
- Comparing different learning algorithms
- Choosing the number of hidden units in a neural net
- Feature selection (see later)
- Choosing a polynomial degree
Cross-validation for classification

• Instead of computing the sum squared errors on a test set, you should compute...

The total number of misclassifications on a testset.
Cross-validation for classification

- Instead of computing the sum squared errors on a test set, you should compute...
  
  The total number of misclassifications on a test set.

- But there’s a more sensitive alternative:
  
  Compute
  
  \[ \log P(\text{all test outputs}|\text{all test inputs, your model}) \]

Cross-Validation for classification

- Choosing the pruning parameter for decision trees
- Feature selection (see later)
- What kind of Gaussian to use in a Gaussian-based Bayes Classifier
- Choosing which classifier to use
Feature Selection

• Suppose you have a learning algorithm LA and a set of input attributes \( \{ X_1, X_2, \ldots, X_m \} \)
• You expect that LA will only find some subset of the attributes useful.
• Question: How can we use cross-validation to find a useful subset?
• Four ideas:
  • Forward selection
  • Backward elimination
  • Hill Climbing
  • Stochastic search (Simulated Annealing or GAs)

Very serious warning

• Intensive use of cross validation can overfit.
• How?

• What can be done about it?
Very serious warning

• Intensive use of cross validation can overfit.

• How?
  • Imagine a dataset with 50 records and 1000 attributes.
  • You try 1000 linear models, each one using one of the attributes.

• What can be done about it?
Very serious warning

- Intensive use of cross validation can overfit.
- How?
  - Imagine a dataset with 50 records and 1000 attributes.
  - You try 1000 linear models, each one using one of the attributes.
  - The best of those 1000 looks good!
  - But you realize it would have looked good even if the output had been purely random!
- What can be done about it?
  - Hold out an additional test set before doing any model selection. The best model should perform well even on the additional test set.

What you should know

- Why you can’t use “training-set-error” to estimate the quality of your learning algorithm on your data.
- Why you can’t use “training set error” to choose the learning algorithm
- Test-set cross-validation
- Leave-one-out cross-validation
- k-fold cross-validation
- Feature selection methods