Cross-validation for detecting and preventing overfitting

Andrew W. Moore
Associate Professor
School of Computer Science
Carnegie Mellon University
www.cs.cmu.edu/~awm
awm@cs.cmu.edu
412-268-7599

Note to other teachers and users of these slides. Andrew would be delighted if you found this source material useful in giving your own lectures. Feel free to use these slides verbatim, or to modify them to fit your own needs. PowerPoint originals are available. If you make use of a significant portion of these slides in your own lecture, please include this message, or the following link to the source repository of Andrew's tutorials:

http://www.cs.cmu.edu/~awm/tutorials.

Comments and corrections gratefully received.

A Regression Problem

\[ y = f(x) + \text{noise} \]
Can we learn \( f \) from this data?

Let's consider three methods...

Linear Regression

Univariate Linear regression with a constant term:

\[ Y = X \beta + \epsilon \]

\[ X = \begin{pmatrix} 1 & 3 \\ 1 & 3 \\ 1 & 3 \\ \vdots & \vdots \end{pmatrix}, \quad Y = \begin{pmatrix} 7 \\ 3 \\ 3 \\ \vdots \end{pmatrix} \]

\[ \beta = (X^T X)^{-1} X^T Y \]

\[ y_{\text{est}} = \beta_0 + \beta_1 x \]

Originally discussed in the previous Andrew Lecture: "Neural Nets"
Quadratic Regression

\[ y = \beta_0 + \beta_1 x + \beta_2 x^2 \]

\[ z = (1, x, x^2) \]

\[ \beta = (Z^T Z)^{-1} (Z^T y) \]

Much more about this in the future
Andrew Lecture: “Favorite Regression Algorithms”

Join-the-dots

Also known as piecewise linear nonparametric regression if that makes you feel better

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?"

Which is best?

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

The test set method

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

(Linear regression example)
Mean Squared Error = 2.4

(Quadratic regression example)
Mean Squared Error = 0.9

(Join the dots example)
Mean Squared Error = 2.2

Good news:
• Very very simple
• Can then simply choose the method with the best test-set score
Bad news:
• What’s the downside?

We say the "test-set estimator of performance has high variance"
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.

\[ \text{MSE}_{\text{LOOCV}} = 2.12 \]
LOOCV for Quadratic Regression

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

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.
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

Linear Regression

MSE_{3FOLD}=2.05

Quadratic Regression

MSE_{3FOLD}=1.11

Join-the-dots

MSE_{3FOLD}=2.93

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</td>
<td>Cheap</td>
</tr>
<tr>
<td></td>
<td>estimate of future</td>
<td></td>
</tr>
<tr>
<td></td>
<td>performance</td>
<td></td>
</tr>
<tr>
<td>Leave-one-out</td>
<td>Expensive. Has some weird behavior</td>
<td>Doesn’t waste data</td>
</tr>
<tr>
<td>10-fold</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>3-fold</td>
<td>Wastier than 10-fold.</td>
<td>Slightly better than test set</td>
</tr>
<tr>
<td>R-fold</td>
<td>Identical to Leave-one-out</td>
<td></td>
</tr>
</tbody>
</table>

But note: One of Andrew’s joys in life is algorithmic tricks for making these cheap
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>f1</th>
<th>f2</th>
<th>f3</th>
<th>f4</th>
<th>f5</th>
<th>f6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>f1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>f2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>f1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Alternatives to CV-based model selection

- Model selection methods:
  1. Cross-validation
  2. AIC (Akaike Information Criterion)
  3. BIC (Bayesian Information Criterion)
  4. VC-dimension (Vapnik-Chervonenkis Dimension)

Which model selection method is best?

1. (CV) Cross-validation
2. AIC (Akaike Information Criterion)
3. BIC (Bayesian Information Criterion)
4. (SRMVC) Structural Risk Minimization with VC-dimension

All of these methods have advantages:
- AIC, BIC and SRMVC advantage: you only need the training error.
- CV error might have more variance
- SRMVC is widely conservative
- Asymptotically AIC and Leave-one-out CV should be the same
- Asymptotically BIC and carefully chosen k-fold should be same
- You want BIC if you want the best structure instead of the best predictor (e.g. for clustering or Bayes Net structure finding)
- Many alternatives—including proper Bayesian approaches.
- It's an emotional issue.

Cross-Validation for regression

- Choosing the number of hidden units in a neural net
- Feature selection (see later)
- Choosing a polynomial degree
- Choosing which regressor to use

Supervising Gradient Descent

- This is a weird but common use of Test-set validation
- Suppose you have a neural net with too many hidden units. It will overfit.
- As gradient descent progresses, maintain a graph of MSE-testset-error vs. Iteration
Supervising Gradient Descent

- This is a weird but common use of Test-set validation
- Suppose you have a neural net with too many hidden units. It will overfit.
- As gradient descent progresses, maintain a graph of MSE-testset-error vs. Iteration.

Use the weights you found on this iteration.

Relies on an intuition that a not-fully-minimized set of weights is somewhat like having fewer parameters.

Works pretty well in practice, apparently.

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.

- 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

Cross-Validation for density estimation

- Compute the sum of log-likelihoods of test points

Example uses:
- Choosing what kind of Gaussian assumption to use
- Choose the density estimator
- NOT Feature selection (testset density will almost always look better with fewer features)
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)

Another fun area in which Andrew has spent a lot of his wild youth

Very serious warning

- Intensive use of cross validation can overfit.
- How?
  - Imagine a dataset with 50 records and 1000 attributes.
  - You try 1000 linear regression models, each one using one of the attributes.
  - The best of those 1000 looks good!

- What can be done about it?

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
- CV for classification, regression & densities