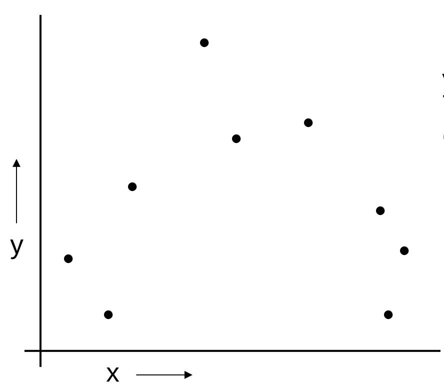
Cross-validation for detecting and preventing overfitting

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.

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

 Oct 15th, 2001

A Regression Problem

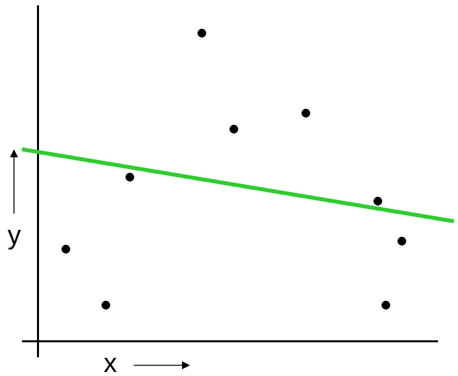


y = f(x) + noise

Can we learn f from this data?

Let's consider three methods…

Linear Regression



Linear Regression

Univariate Linear regression with a constant term:

X	*Y*
3	7
1	3
:	:

$$\mathbf{X} = \begin{bmatrix} 3 \\ 1 \\ \vdots \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 7 \\ 3 \\ \vdots \end{bmatrix}$$

$\mathbf{x}_1=(3)$.. $\qquad y_1=7$..

Originally discussed in the previous Andrew Lecture: “Neural Nets”

Linear Regression

Univariate Linear regression with a constant term:

X	Y
3	7
1	3
:	:

$\mathbf{X}=\begin{bmatrix}3\\1\\:\end{bmatrix}$ $\mathbf{y}=\begin{bmatrix}7\\3\\:\end{bmatrix}$

$y_1=7..$

$\mathbf{Z}=\begin{bmatrix}1 & 3\\1 & 1\\: & \end{bmatrix}$ $\mathbf{y}=\begin{bmatrix}7\\3\\:\end{bmatrix}$

$\mathbf{z}_1=(1,3)..$ $y_1=7..$

$\mathbf{z}_k=(1,x_k)$

Linear Regression

Univariate Linear regression with a constant term:

X	Y
3	7
1	3
:	:

$\mathbf{X}=\begin{bmatrix}3\\1\\:\end{bmatrix}$ $\mathbf{y}=\begin{bmatrix}7\\3\\:\end{bmatrix}$

$y_1=7..$

$\mathbf{Z}=\begin{bmatrix}1 & 3\\1 & 1\\: & \end{bmatrix}$ $\mathbf{y}=\begin{bmatrix}7\\3\\:\end{bmatrix}$

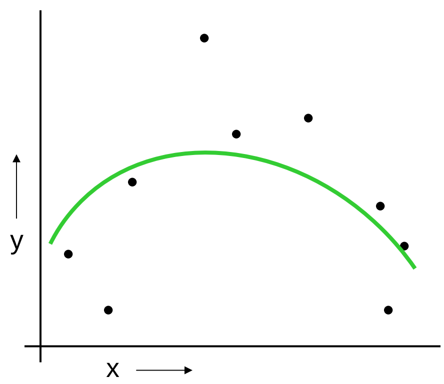
$\mathbf{z}_1=(1,3)..$ $y_1=7..$

$\mathbf{z}_k=(1,x_k)$

$\beta=(\mathbf{Z}^T\mathbf{Z})^{-1}(\mathbf{Z}^T\mathbf{y})$

$y^{est} = \beta_0 + \beta_1 x$

Quadratic Regression



Quadratic Regression

X	Y
3	7
1	3
:	:

$$\mathbf{X}=\begin{bmatrix}3\\1\\:\end{bmatrix} \qquad \mathbf{y}=\begin{bmatrix}7\\3\\:\end{bmatrix}$$

$y_1=7..$

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

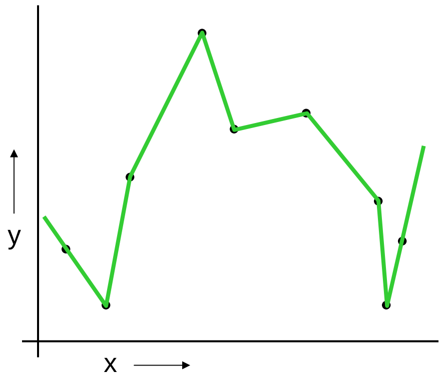
$$\mathbf{Z}=\begin{bmatrix}1 & 3 & 9\\1 & 1 & 1\\: & & \end{bmatrix} \qquad \mathbf{y}=\begin{bmatrix}7\\3\\:\end{bmatrix}$$

$\mathbf{z}=(1,\ x, x^2)$

$$\beta=(\mathbf{Z}^T\mathbf{Z})^{-1}(\mathbf{Z}^T\mathbf{y})$$

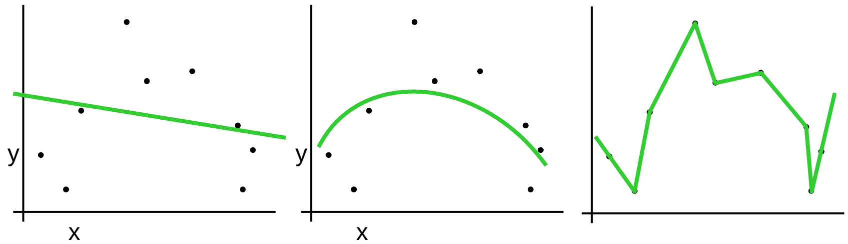
$$y^{est} = \beta_0+ \beta_1 x+ \beta_2 x^2$$

Join-the-dots



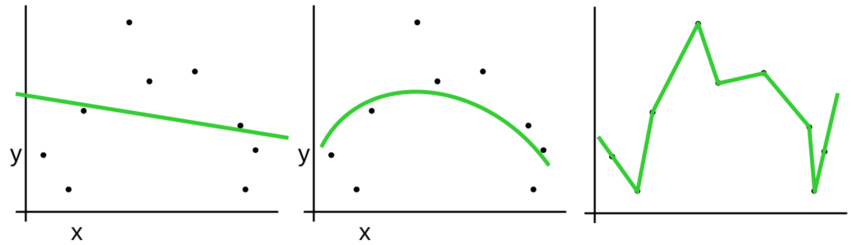
Also known as piecewise linear nonparametric regression if that makes you feel better

Which is best?



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

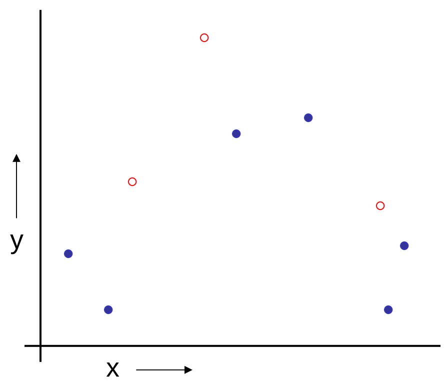
What do we really want?



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?"

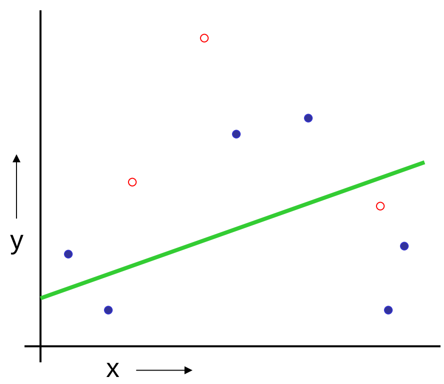
The test set method



1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

The test set method



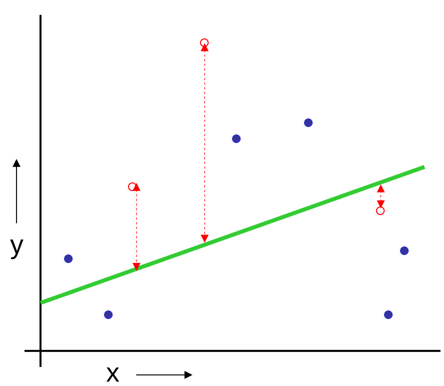
1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

3. Perform your regression on the training set

(Linear regression 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

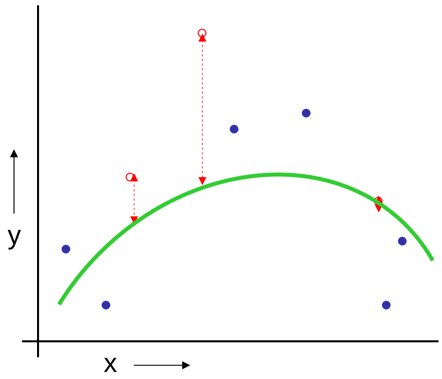
3. Perform your regression on the training set

4. Estimate your future performance with the test set

(Linear regression example)

Mean Squared Error = 2.4

The test set method

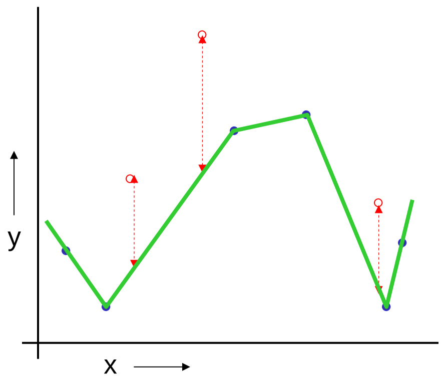


1. Randomly choose 30% of the data to be in a test set
2. The remainder is a training set
3. Perform your regression on the training set
4. Estimate your future performance with the test set

(Quadratic regression example)

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 your regression on the training set
4. Estimate your future performance with the test set

(Join the dots example)

Mean Squared Error = 2.2

The test set method

Good news:

•Very very simple

•Can then 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 then simply choose the method with the best test-set score

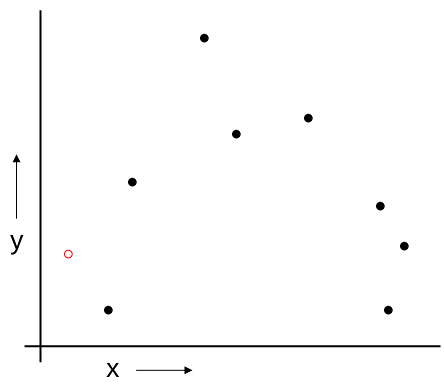
Bad news:

•Wastes data: we get an estimate of the best method to apply to 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"

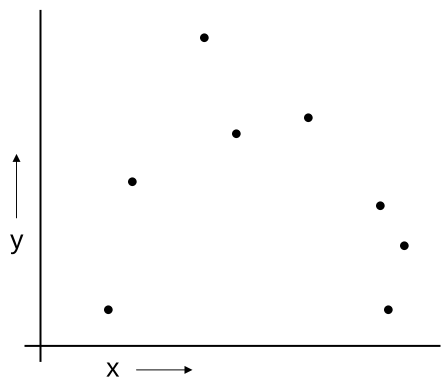
LOOCV (Leave-one-out Cross Validation)



For k=1 to R

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

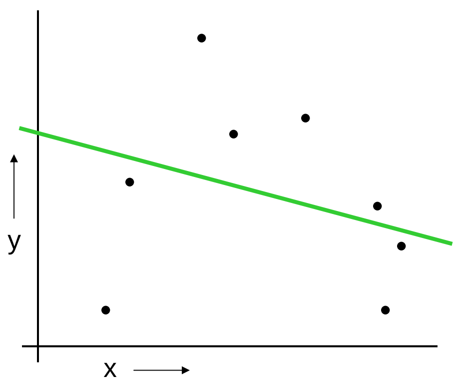
LOOCV (Leave-one-out Cross Validation)



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

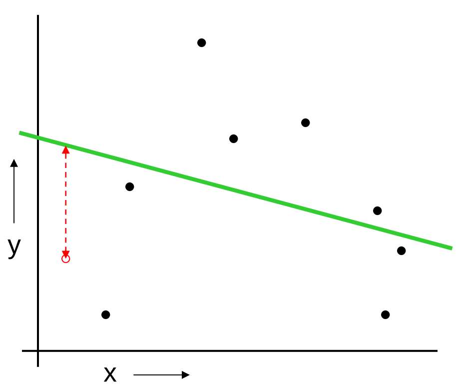
LOOCV (Leave-one-out Cross Validation)



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

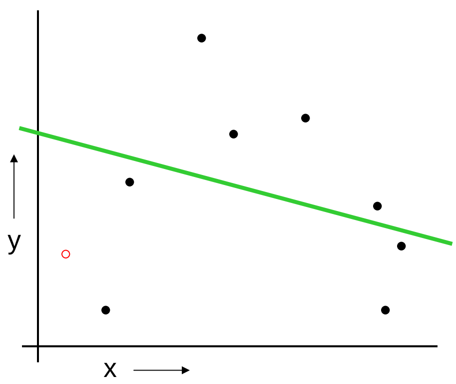
LOOCV (Leave-one-out Cross Validation)



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)

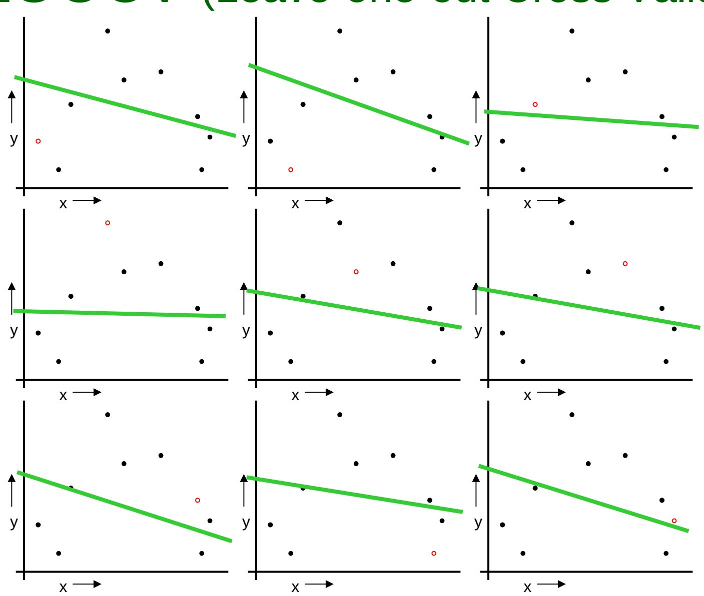


For k=1 to R

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 (Leave-one-out Cross Validation)



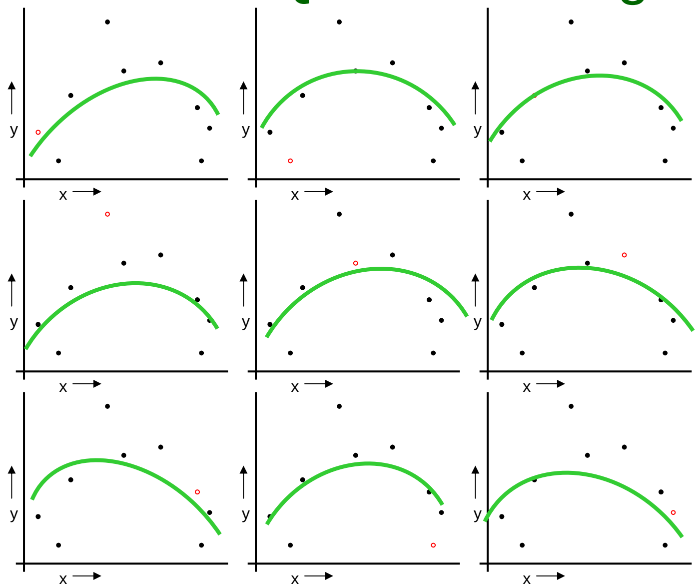
For k=1 to R

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.

$MSE_{LOOCV} = 2.12$

LOOCV for Quadratic Regression



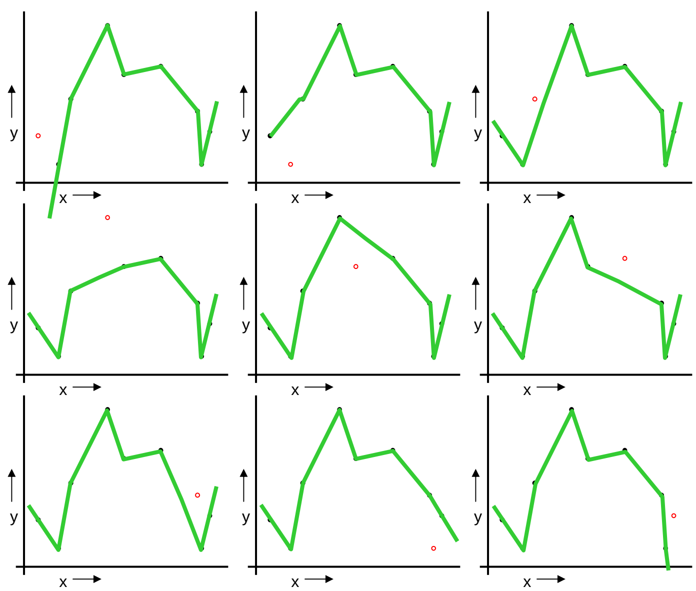
For k=1 to R

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.

MSE$_{LOOCV}$ *=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.

MSE$_{LOOCV}$ *=3.33*

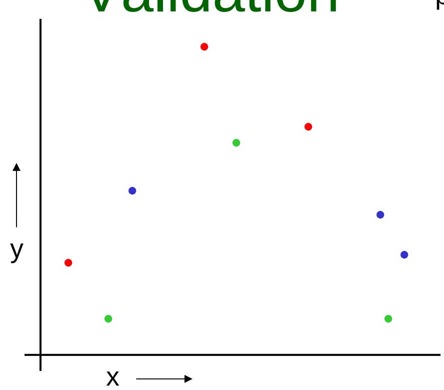
Which kind of Cross Validation?

	Downside	**Upside**
Test-set	Variance: unreliable estimate of future performance	Cheap
Leave-one-out	Expensive. Has some weird behavior	Doesn't waste data

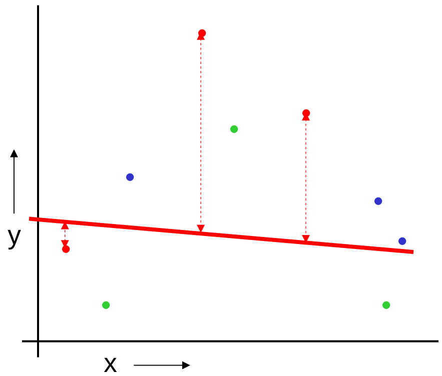
..can we get the best of both worlds?

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)



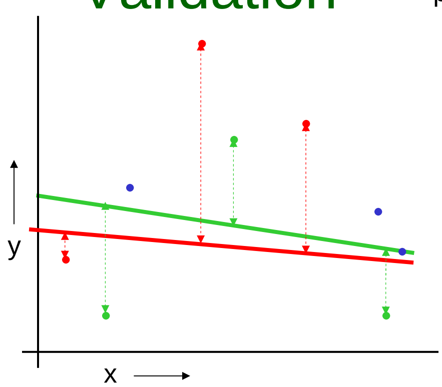
k-fold Cross Validation



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.

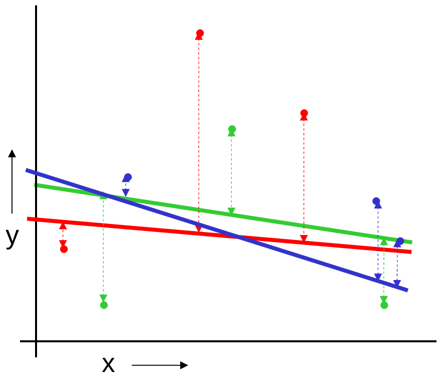
k-fold Cross Validation



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



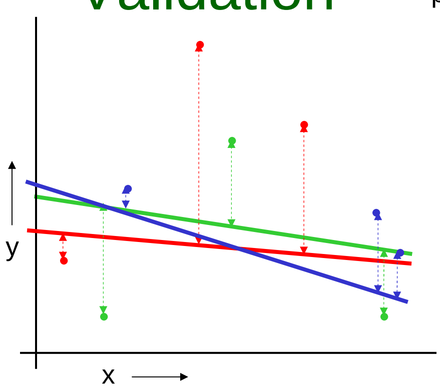
Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

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.

k-fold Cross Validation



Linear Regression
MSE_{3FOLD}=2.05

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

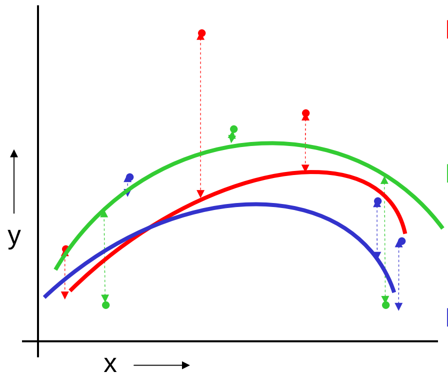
For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

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

k-fold Cross Validation



Quadratic Regression

MSE_{3FOLD}=1.11

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

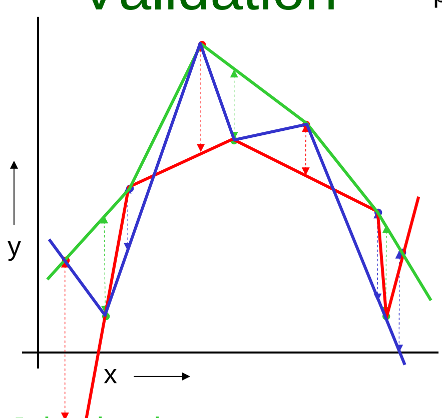
For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

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

k-fold Cross Validation



Joint-the-dots

MSE_{3FOLD}=2.93

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

Which kind of Cross Validation?

	Downside	Upside
Test-set	Variance: unreliable estimate of future performance	Cheap
Leave-one-out	Expensive. Has some weird behavior	Doesn't waste data
10-fold	Wastes 10% of the data. 10 times more expensive than test set	Only wastes 10%. Only 10 times more expensive instead of R times.
3-fold	Wastier than 10-fold. Expensivier than test set	Slightly better than test-set
R-fold	Identical to Leave-one-out	

Which kind of Cross Validation?

	Downside	Upside
Test-set	Variance: unreliable estimate of future performance	Cheap
Leave-one-out	Expensive. Has some weird behavio	
10-fold	Wastes 10% of the data 10 times more expensive than testset	ly sive instead of R times.
3-fold	Wastier than 10-fold. Expensivier than testset	Slightly better than test-set
R-fold	Identical to Leave-one-out	

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…

i	f_i	TRAINERR	10-FOLD-CV-ERR	Choice
1	f_1			
2	f_2			
3	f_3			⌧
4	f_4			
5	f_5			
6	f_6			

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)

Only directly applicable to choosing classifiers

Described in a future Andrew Lecture

Which model selection method is best?

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

- AIC, BIC and SRMVC advantage: you only need the training error.
- CV error might have more variance
- SRMVC is wildly 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 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.

Other Cross-validation issues

- Can do "leave all pairs out" or "leave-all-ntuples-out" if feeling resourceful.
- Some folks do k-folds in which each fold is an independently-chosen subset of the data
- Do you know what AIC and BIC are?

  If so…

  - LOOCV behaves like AIC asymptotically.
  - k-fold behaves like BIC if you choose k carefully

  If not…

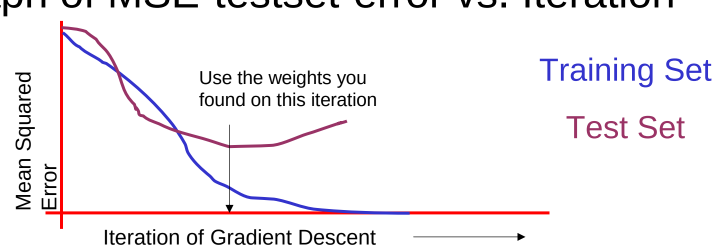
  - Nyardely nyardely nyoo nyoo

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

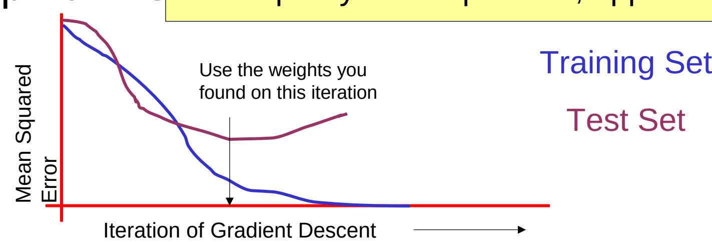


Supervising Gradient Descent

- This is a weird but common use of Test-set validation
- Suppose you hav   ral net with too many hidde
- As gradient
graph of MS

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…

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 testset.

- But there's a more sensitive alternative:

  Compute

  log P(all test outputs|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 .. 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?
- 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 regression 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 regression models, each one using one of the attributes.
  - The best of those 1000 looks good!
- 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 regression 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 testset before doing any model selection. Check the best model performs well even on the additional testset.

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