Plan:

1. Discuss the bias-variance tradeoff
2. Introduce Cross Validation (CV)

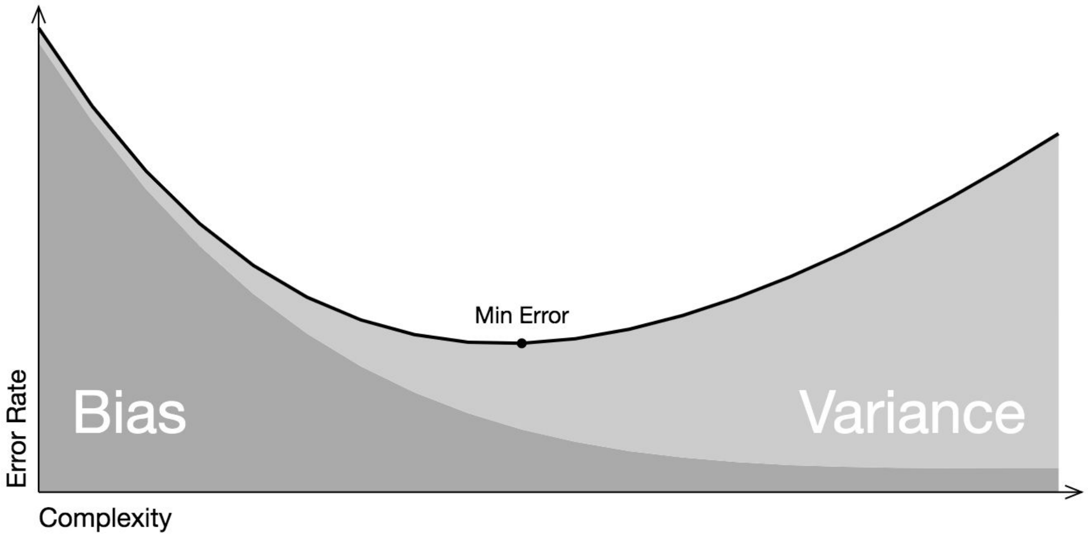
Machine Learning: Validation

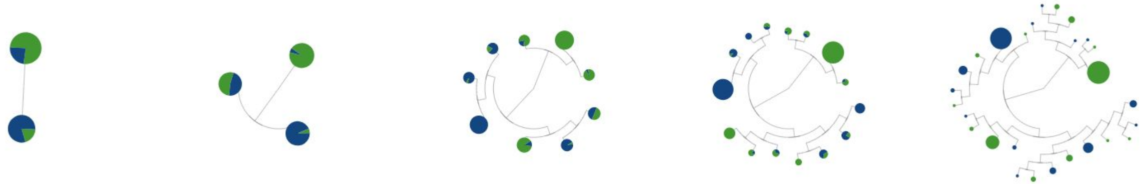
Shannon E. Ellis, Ph.D
UC San Diego

Department of Cognitive Science
sellis@ucsd.edu

So...what can we do about overfitting?

Bias-variance tradeoff



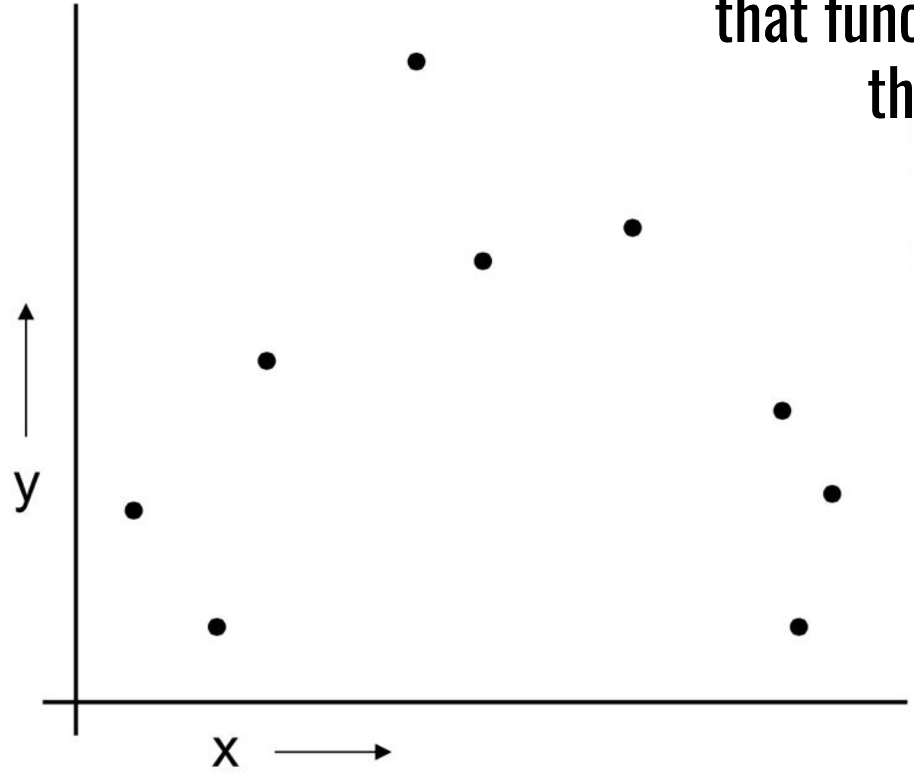


http://www.r2d3.us/visual-intro-to-machine-learning-part-1/?from=@

Bias-variance tradeoff

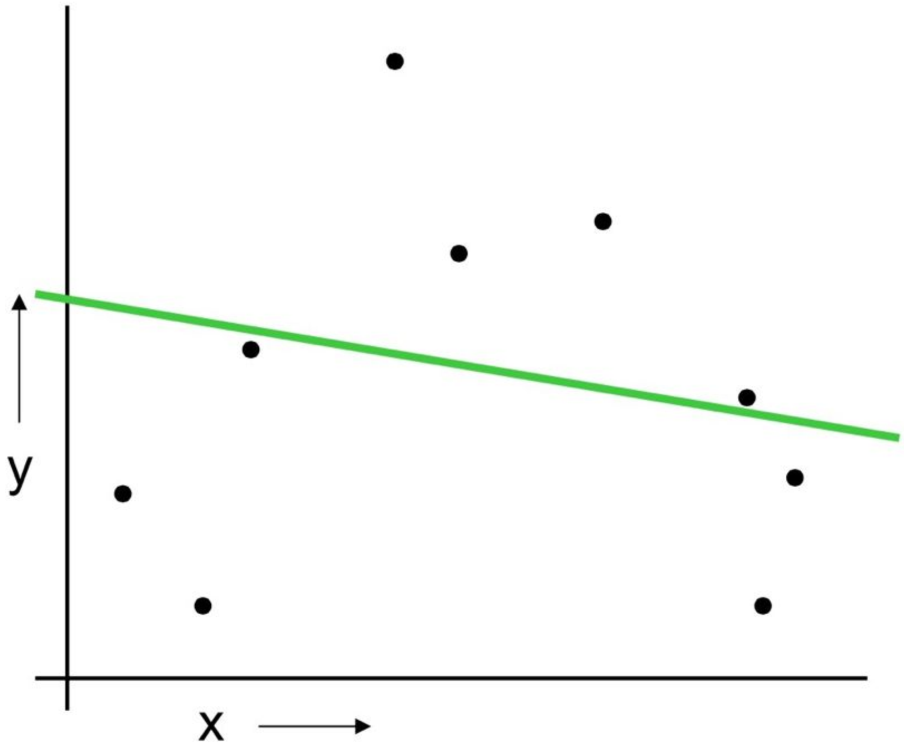
- **High variance** models make mistakes in *inconsistent* ways.
- **Biased models** tend to be overly simple and not reflect reality
- <u>What to do:</u>
  - Consider tuning parameters in the model
    - can avoid overfitting by setting minimum node size threshold (fewer splits; variance decreased)
  - Changing model approach
    - Bagging, boosting, & ensemble methods
  - Re-consider data splitting approach
    - Training + test?
    - LOOCV
    - K-fold CV

Can we determine what that function (`f`) *is* using these data?



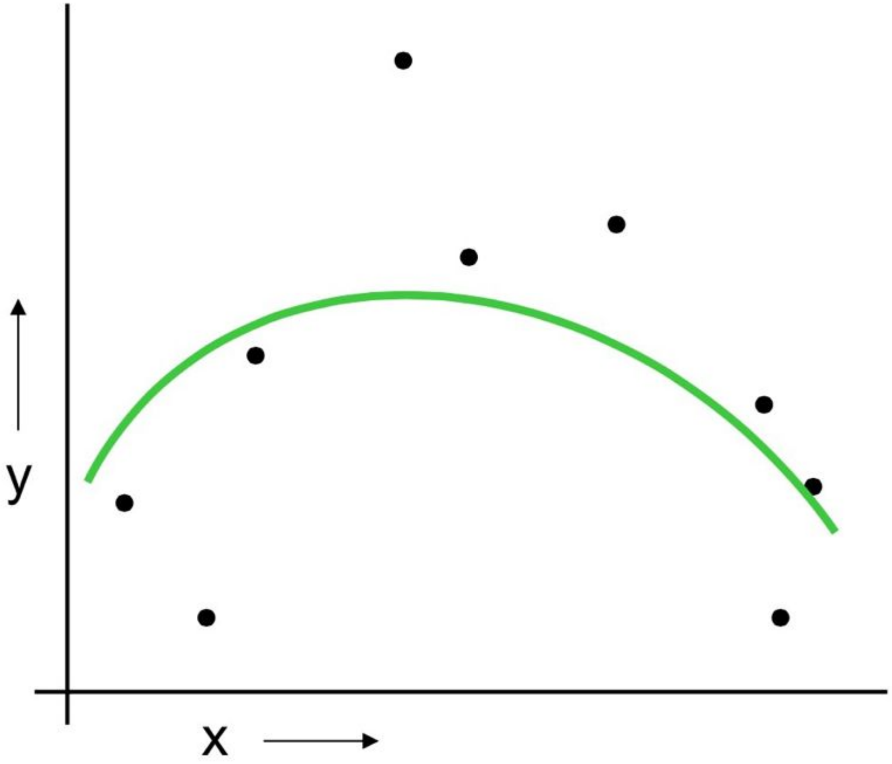
`y = f(x) + noise`

Linear regression

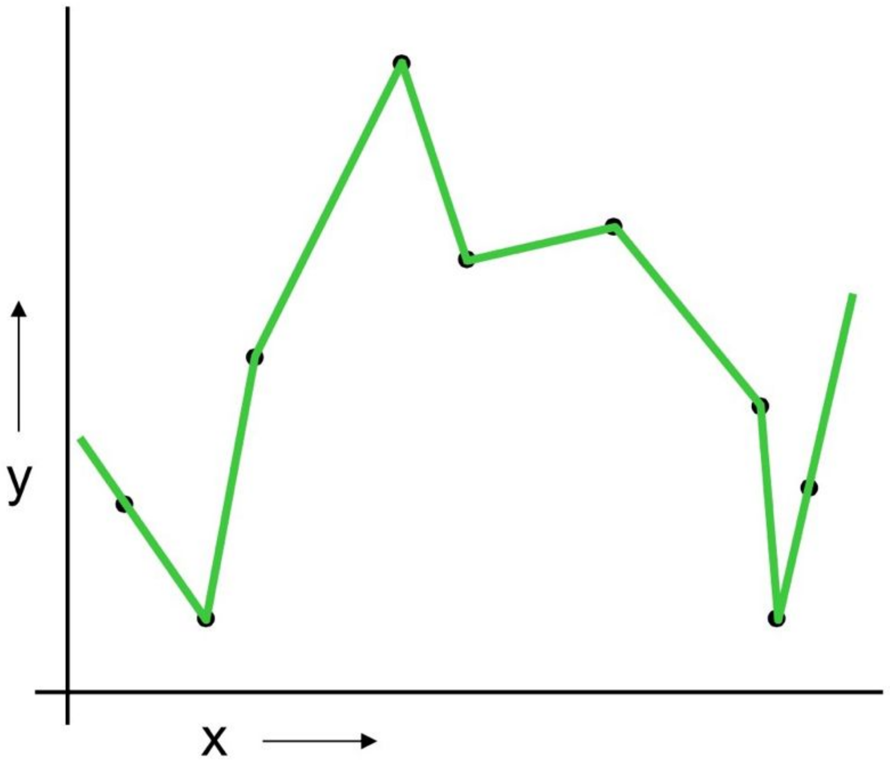


adapted from Brad Voytek

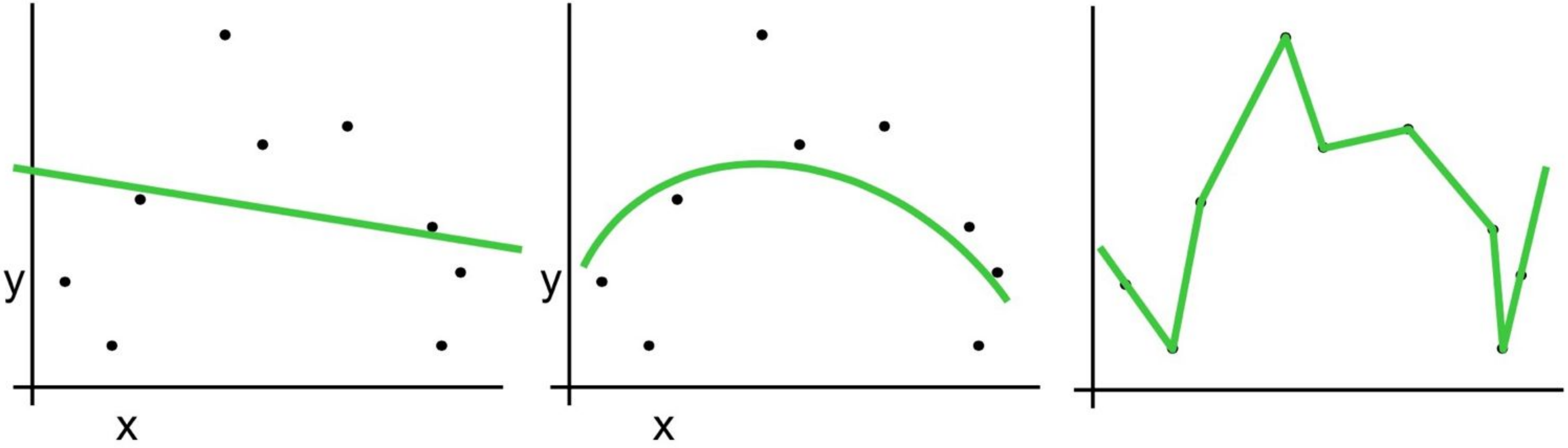
Quadratic regression



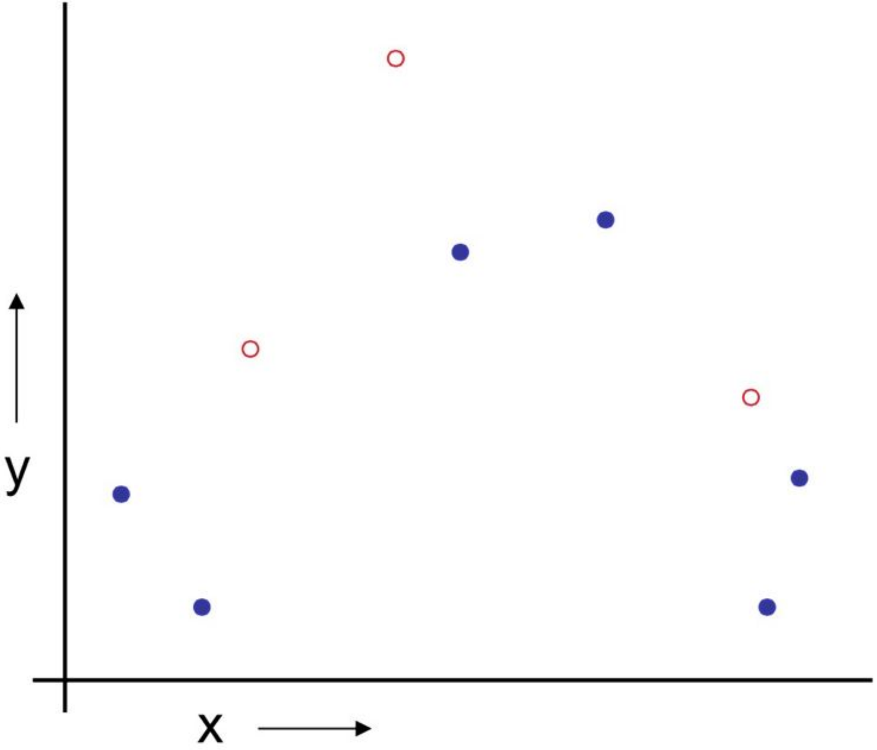
Piecewise linear nonparametric regression



Which to choose?



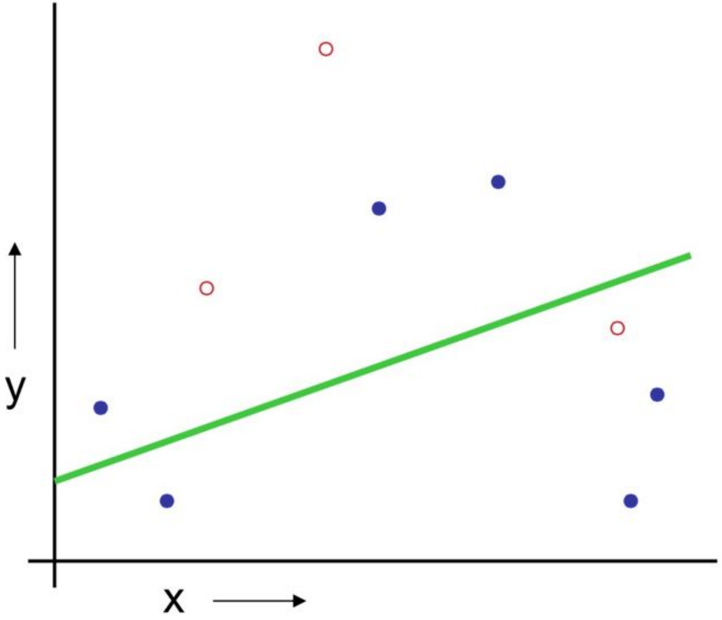
The data partition method



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

2. The remainder is a training set

Train the model on your training set



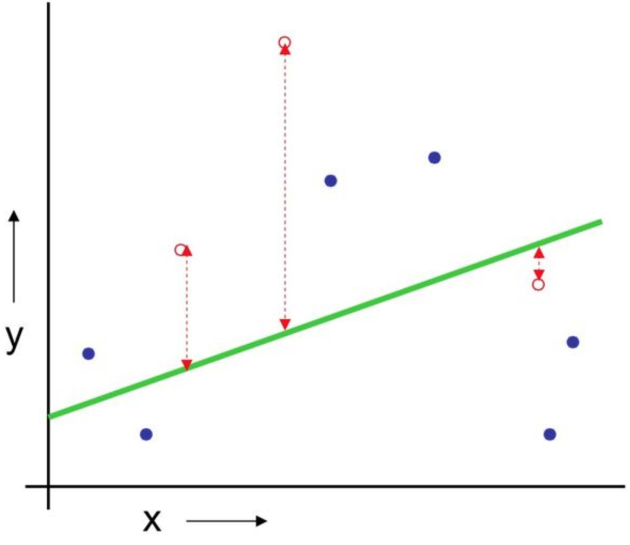
1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

3. Perform your regression on the training set

(Linear regression example)

Assess future performance using the test set



(Linear regression example)

Mean Squared Error = 2.4

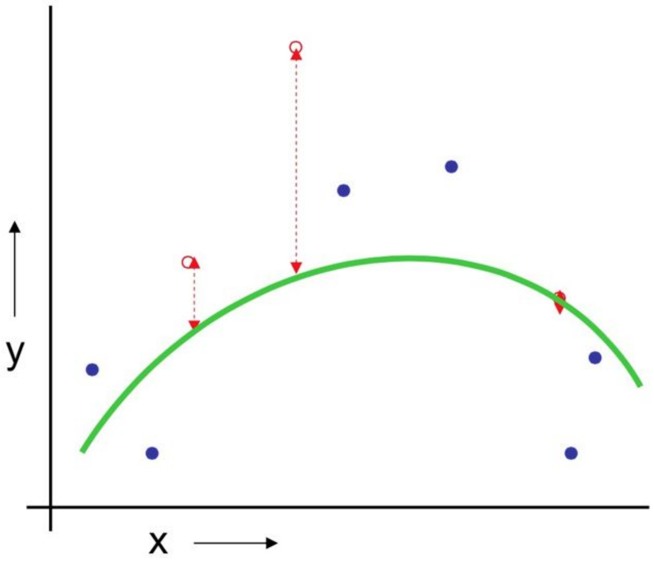
1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

3. Perform your regression on the training set

4. Estimate your future performance with the test set

Go through this process for each possible model



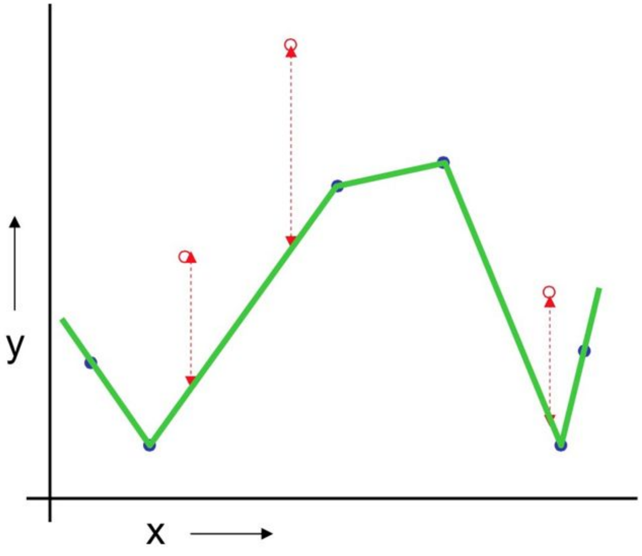
(Quadratic regression example)

Mean Squared Error = 0.9

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

adapted from Brad Voytek

Go through this process for each possible model



(Join the dots example)

Mean Squared Error = 2.2

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

Pros and cons of data partitioning

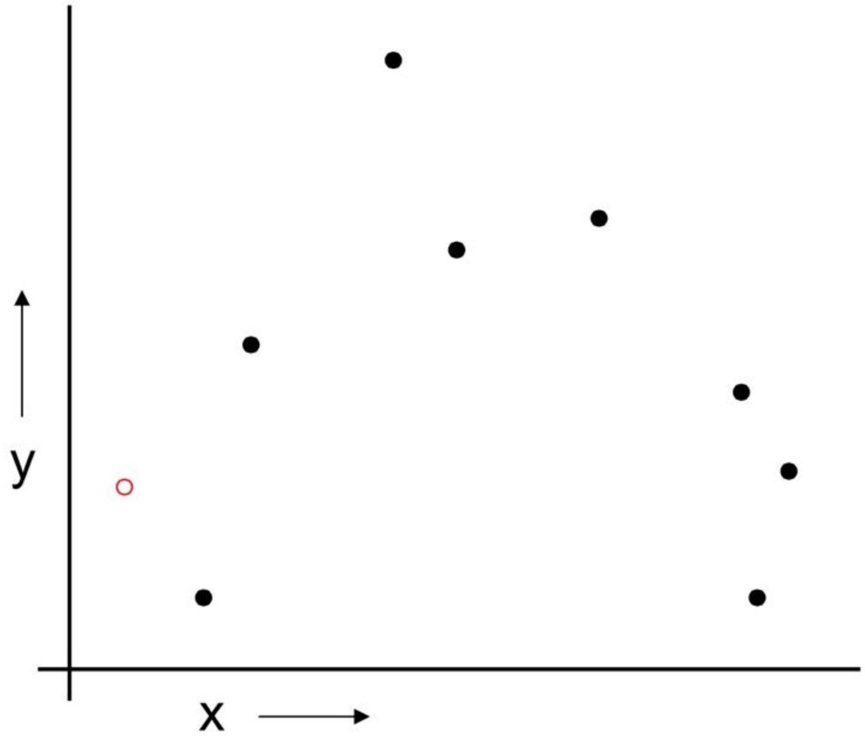
Pros:

- Simple approach
- Can choose model with best test-set score

Cons:

- Model fit on 30% less data than you have
- Without a large data set, removing 30% of the data could bias prediction

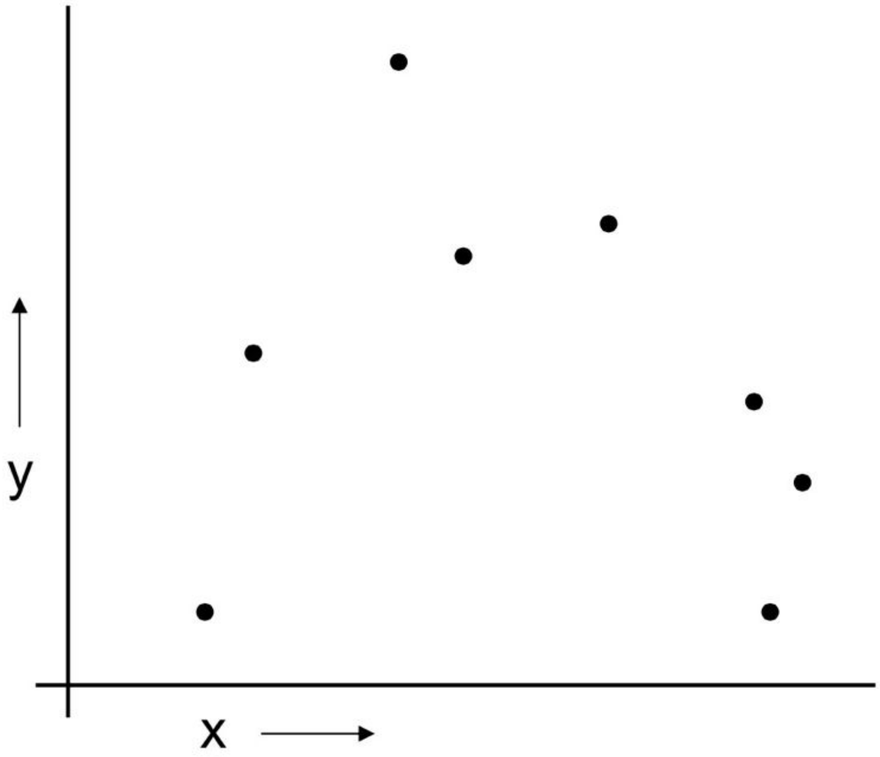
Leave one out cross validation (LOOCV)



For k=1 to R

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

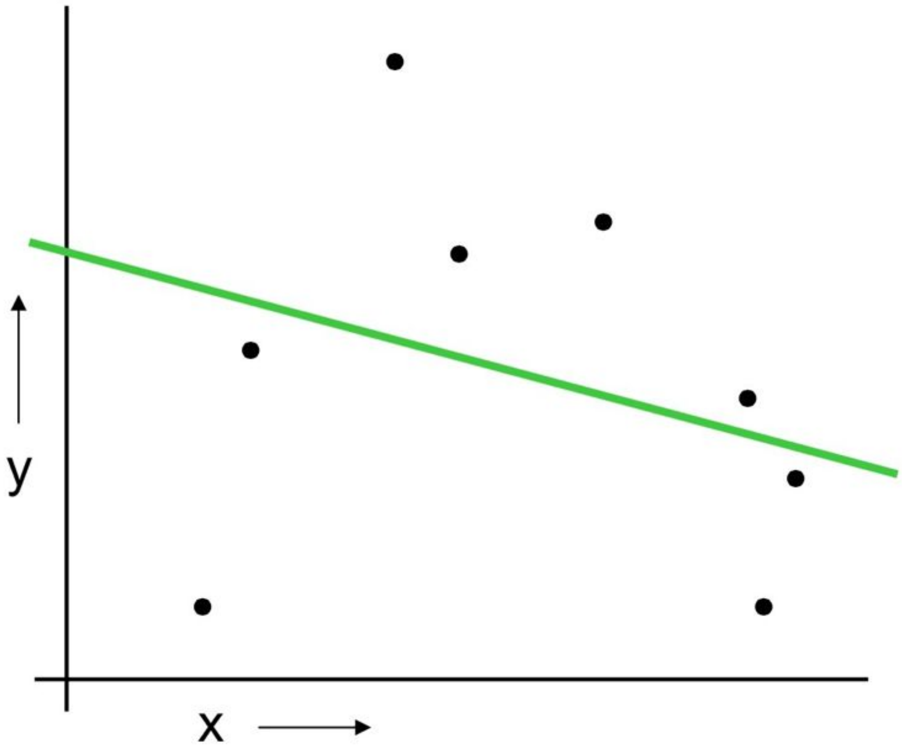
Leave one out cross validation (LOOCV)



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

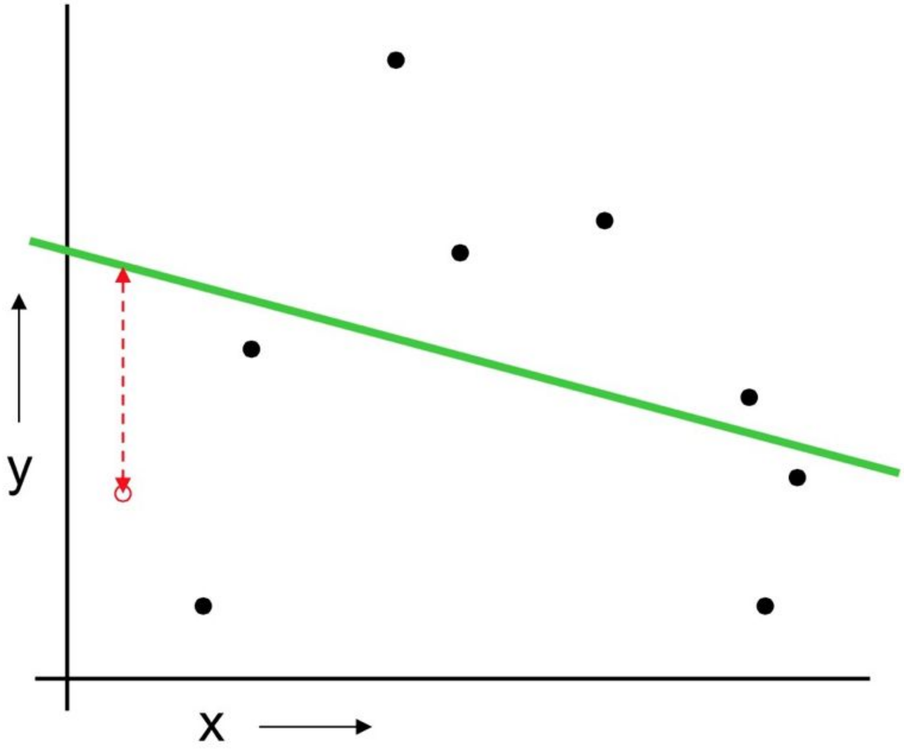
Leave one out cross validation (LOOCV)



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

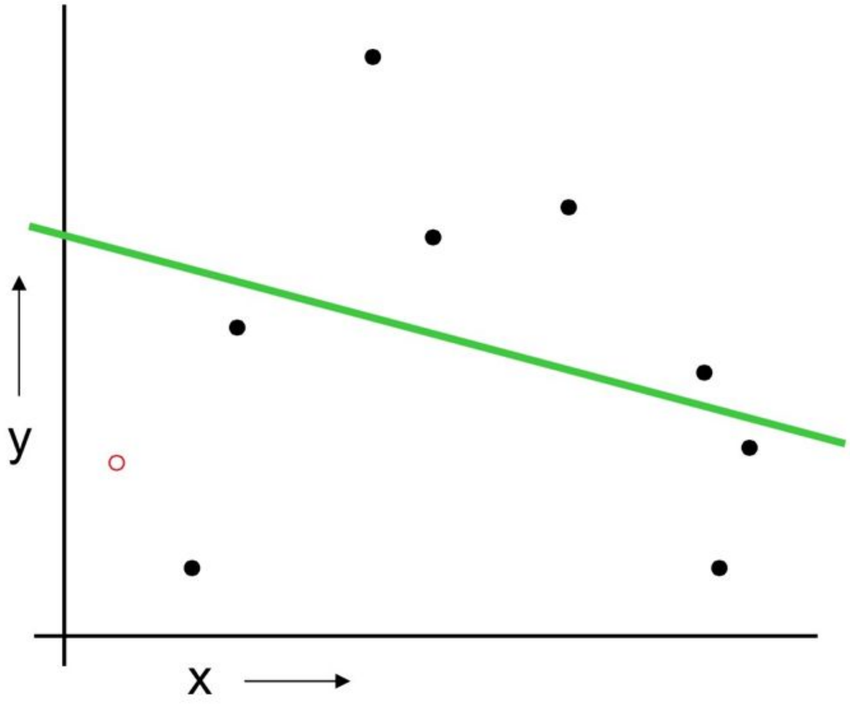
Leave one out cross validation (LOOCV)



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)

Leave one out cross validation (LOOCV)

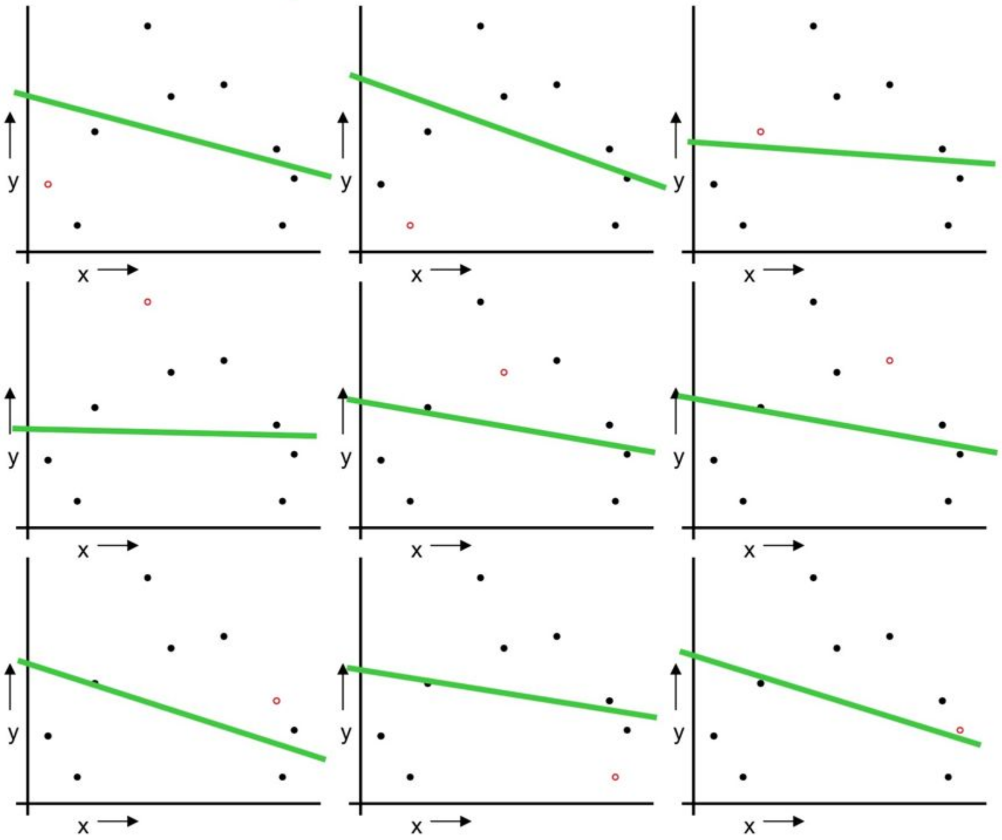


For k=1 to R

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.

Leave one out cross validation (LOOCV)



For k=1 to R

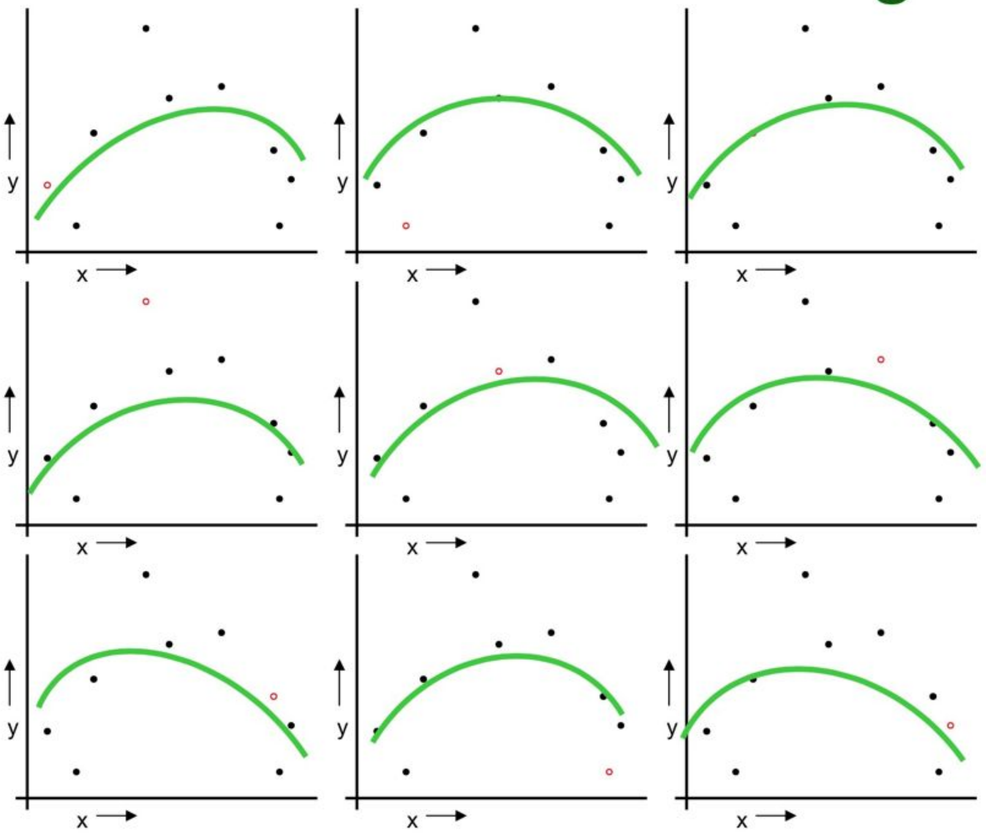
1. Let (x_k, y_k) be the k^{th} record
2. Temporarily remove (x_k, y_k) from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error (x_k, y_k)

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

$MSE_{LOOCV} = 2.12$

adapted from Brad Voytek

Leave one out cross validation (LOOCV)



For k=1 to R

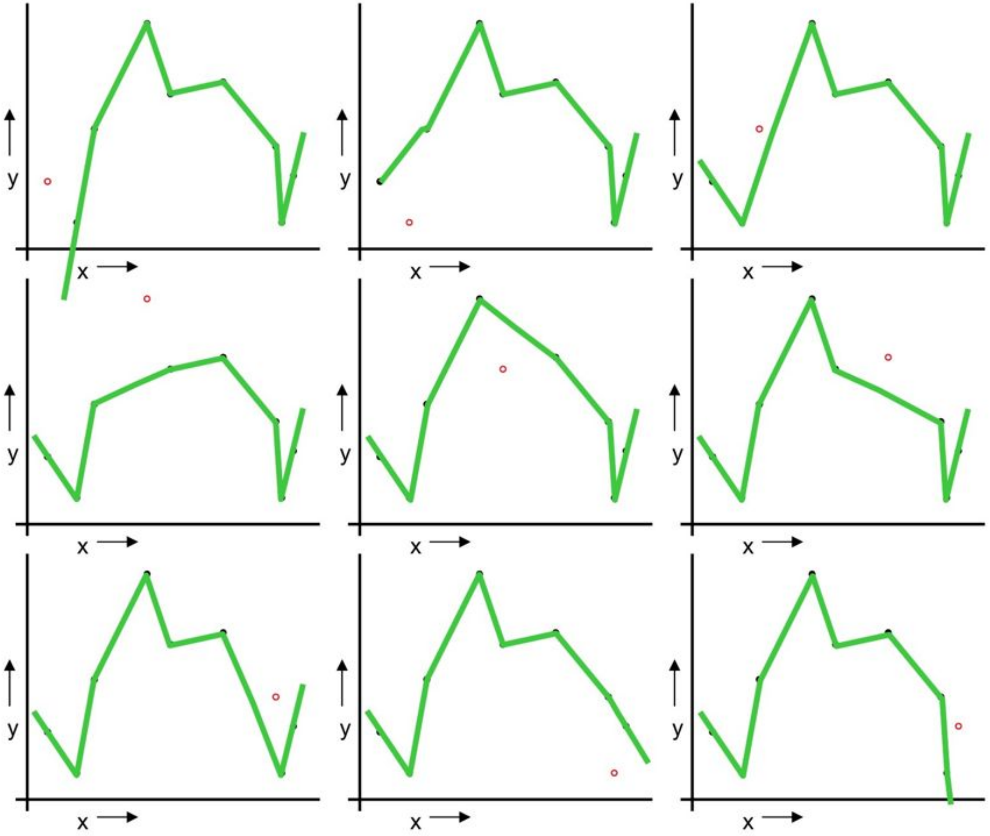
1. Let (x_k, y_k) be the k^{th} record
2. Temporarily remove (x_k, y_k) from the dataset
3. Train on the remaining R-1 datapoints
4. Note your error (x_k, y_k)

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

MSE_{LOOCV}
$=0.962$

adapted from Brad Voytek

Leave one out cross validation (LOOCV)



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)

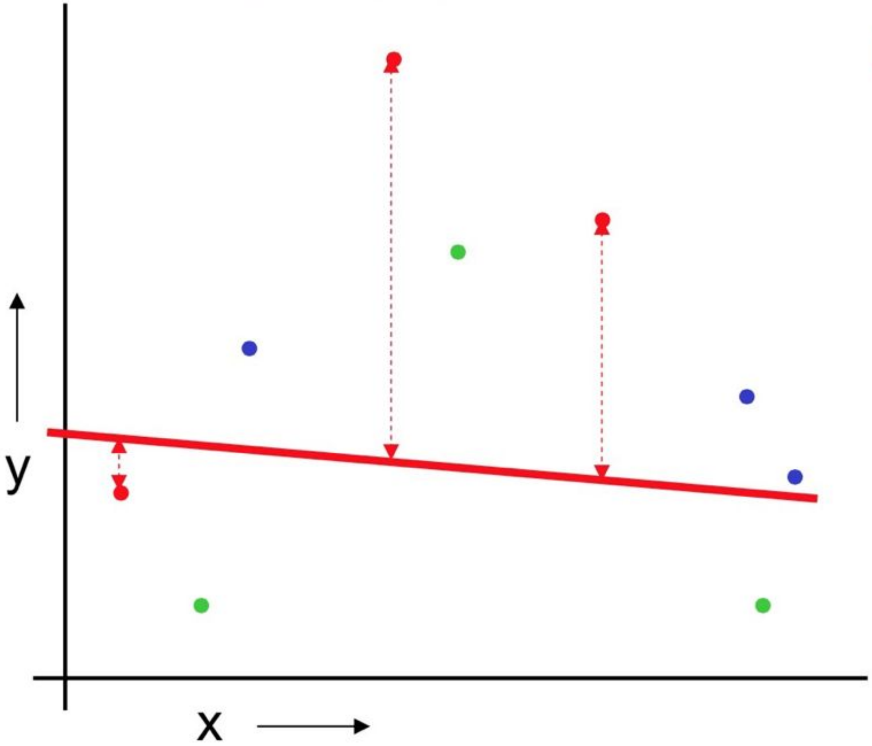
When you've done all points, report the mean error.

$MSE_{LOOCV} = 3.33$

Method Comparison

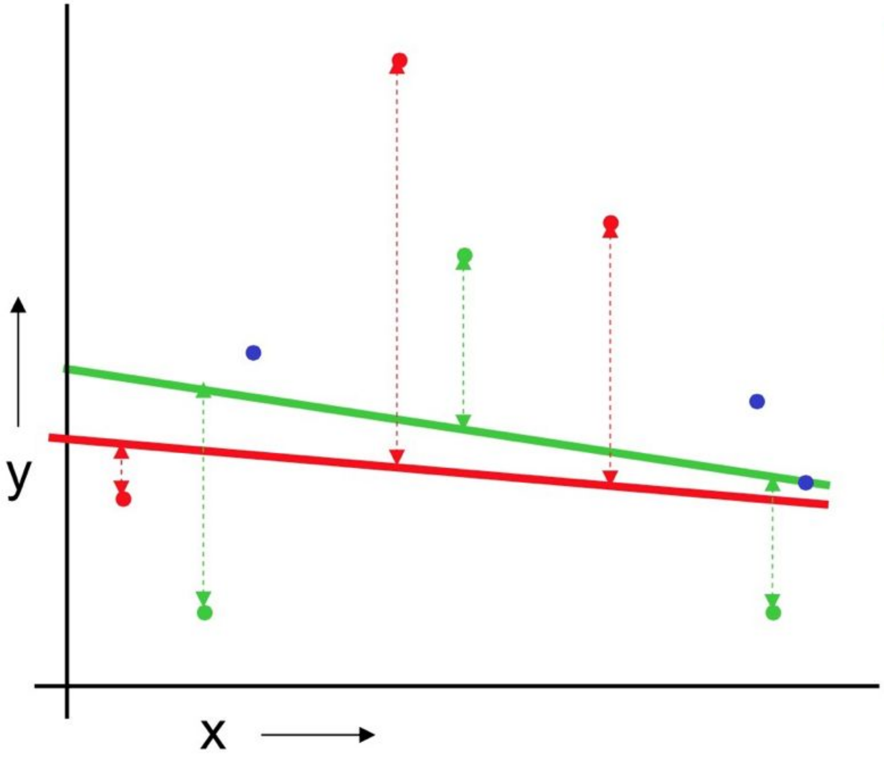
	Cons	Pros
Data partitioning	Variance: unreliable estimate of future performance	Cheap
LOOCV	Computationally expensive; has weird behavior	Uses all your data

k-fold cross validation



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.

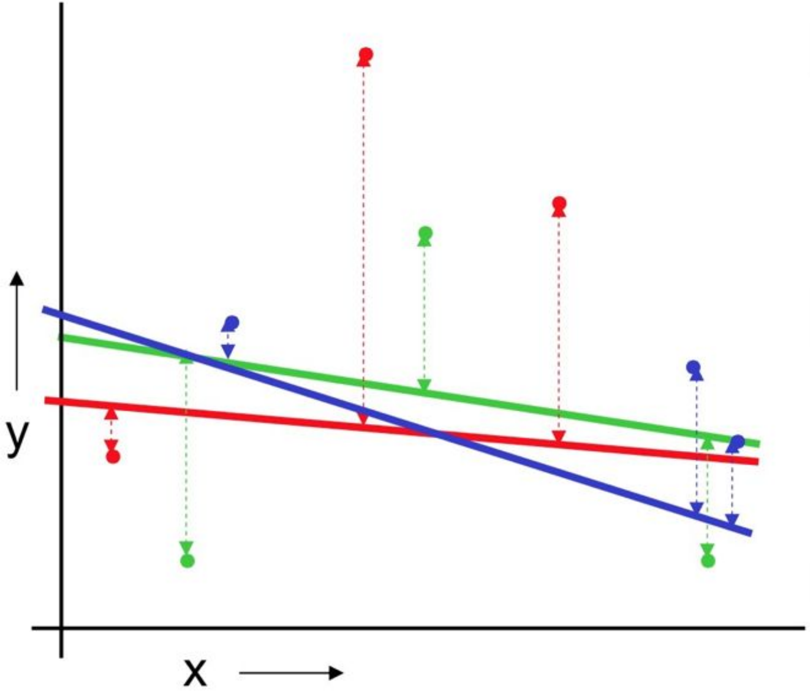
k-fold cross validation



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

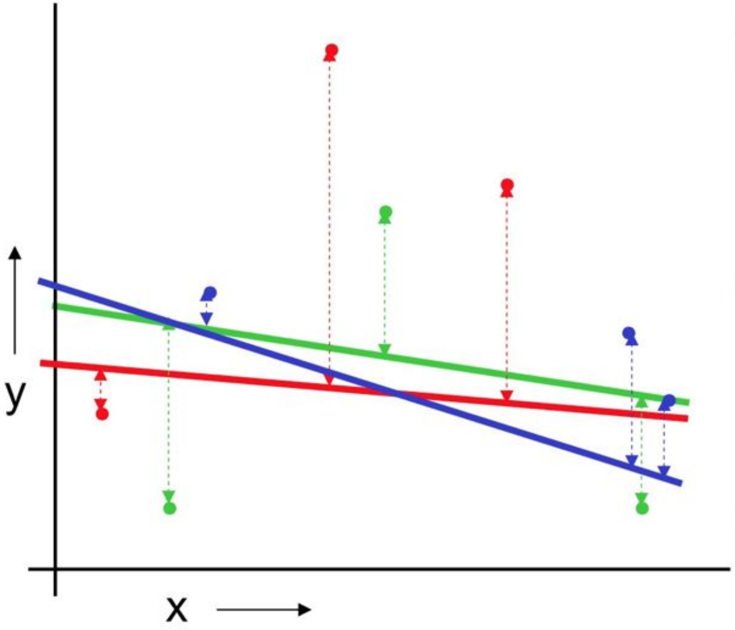


For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

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
$MSE_{3FOLD}=2.05$

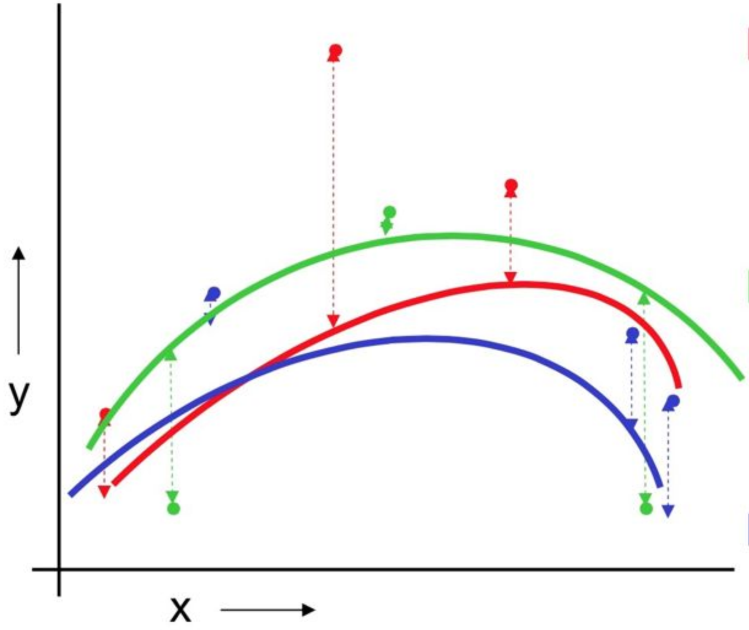
For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

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

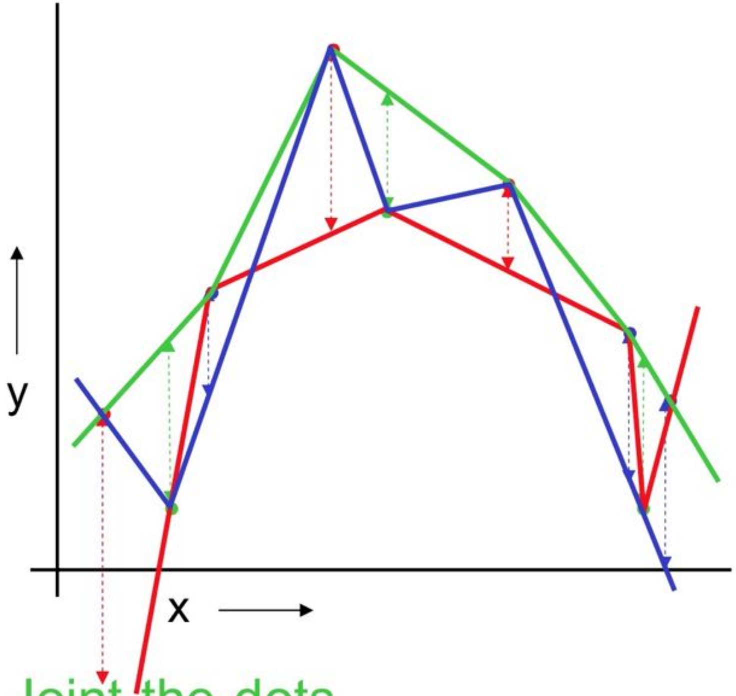
For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

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

$MSE_{3FOLD}=2.93$

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

Validator

Given the example we just worked, how would *you* model these data?

A
linear
regression

Linear Regression
MSE_{3FOLD}=2.05

B
quadratic
regression

Quadratic Regression
MSE_{3FOLD}=1.11

C
pairwise linear
nonparametric
regression

Joint-the-dots
MSE_{3FOLD}=2.93

Validator

Which approach would *you* use to limit overfitting?