Overfitting, Validation

Machine Learning I, Week 6

Sibylle Mueller

The overfitting and validation examples are taken from:

Andrew W. Moore

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

Overview

What we had so far:

- Given data
- Decide for a type of function parametrized with weights
- Optimize weights w.r.t. best fit to data

Today:

- How do we choose the function? Should it be linear, quadratic, exponential, ... ? Is there a systematic in the choice of the function?

Today‘s Topics

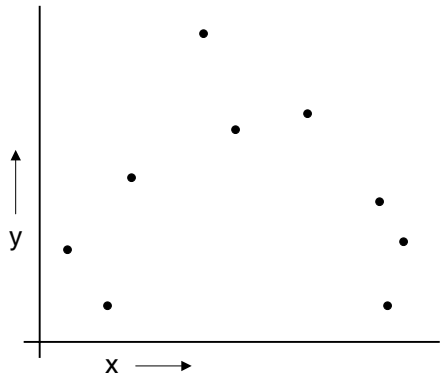
Overfitting:

- Example: Fit three different functions to data
- Measure the quality of a fit: Estimation of true risk

Validation / Resampling:

- Cross-Validation
- Analytic validation techniques

Example: A Regression Problem



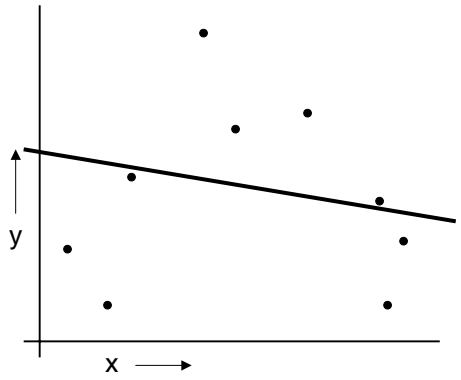
y = f(x) + noise

Can we learn f from this data?

Let’s consider three methods:

- Linear regression
- Quadratic regression
- Join-the-dots

Linear Regression



Linear Regression

with a constant term:

X	Y
3	7
1	3
:	:

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

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

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

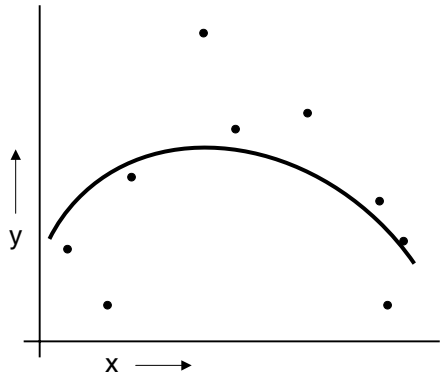
$\boldsymbol{z}_1=(1,3)..$ $y_1=7..$

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

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

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

Quadratic Regression



Quadratic Regression

X	Y
3	7
1	3
:	:

X=

3
1
:

y=

7
3
:

y_1=7..

Z=

1	3	9
1	1	1
:		

y=

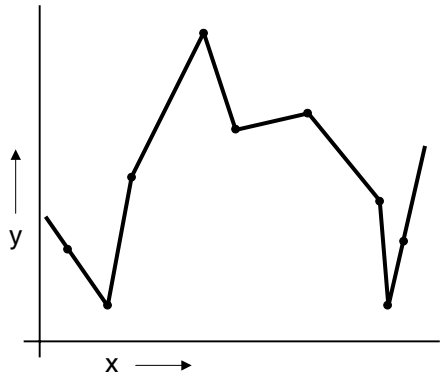
7
3
:

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

$z=(1,\ x,\ x^2)$

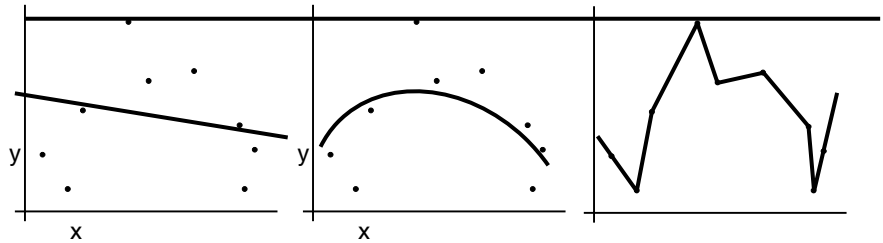
$y^{est} = \beta_0 + \beta_1 x + \beta_2 x^2$

Join-the-dots



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

Which is best?



Underfitting ? "Right" fitting ? Overfitting ?

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

→ overfitting

The key question is: How well are you going to predict future data drawn from the same distribution?

Empirical Risk vs. True Risk

Risk = expected value of the loss L

Empirical risk = average value of L for given set of data. Minimizing L over training data = empirical risk minimization (ERM)

True risk = expected value of L over true density of data:

$$R(w) = \int L(y, f(\mathbf{x}, w))\, p(\mathbf{x}, y)\, d\mathbf{x}\, dy$$

The problem here: We do not know the probability density function p(**x**,y) that created our data. So we cannot compute the true risk. → Estimate the true risk

Maximum likelihood leads to specific loss functions for density estimation, regression, and classification.

Estimation of the True Risk

In **Validation**, the true risk is estimated.

Two different approaches for validation:

- Resampling (today):

  uses given data

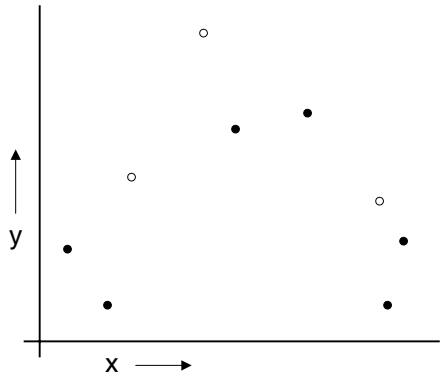
  e.g., Test set method, Cross-Validation

- Analytic (next week):

  estimates analytically the true risk (realm of statisticians), e.g., AIC, BIC

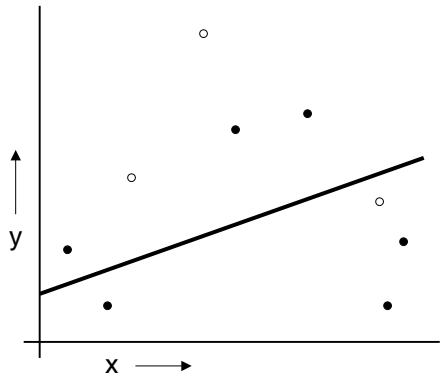
  depends on model complexity (VC-dimension)

The test set method



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

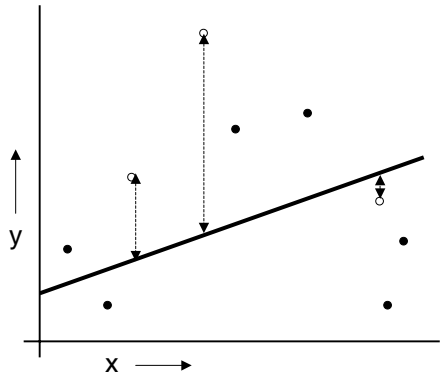
The test set method



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

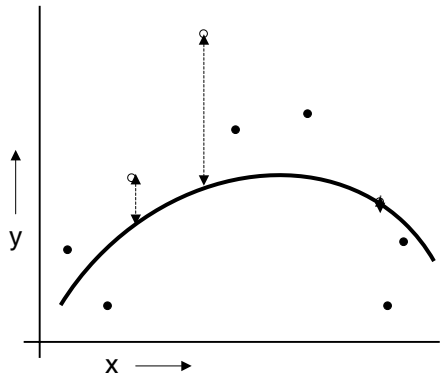


(Linear regression example)

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

The test set method

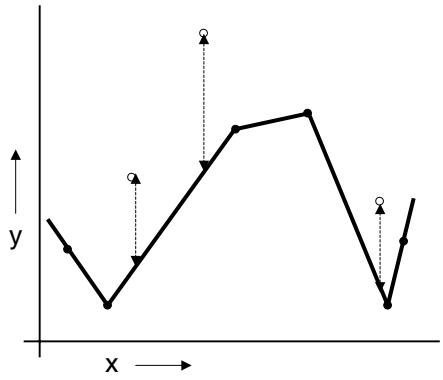


(Quadratic regression example)

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

The test set method



(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

The test set method

Good news:

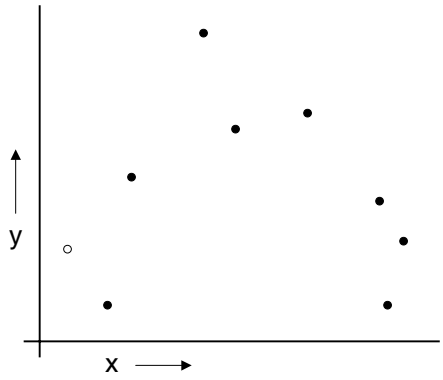
- Very very simple
- Can then simply choose the method with the best test-set score

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"

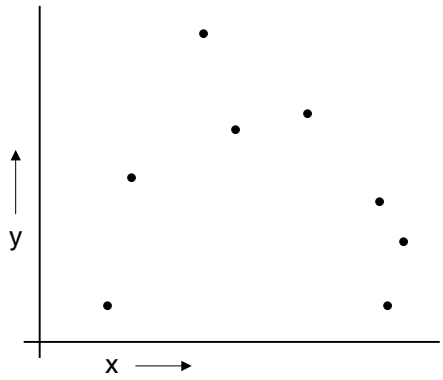
LOOCV (Leave-one-out Cross Validation)



For k=1 to R

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

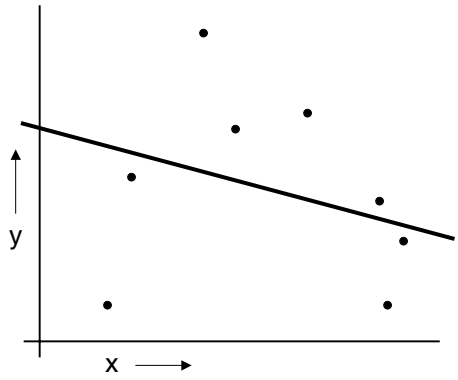
LOOCV (Leave-one-out Cross Validation)



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

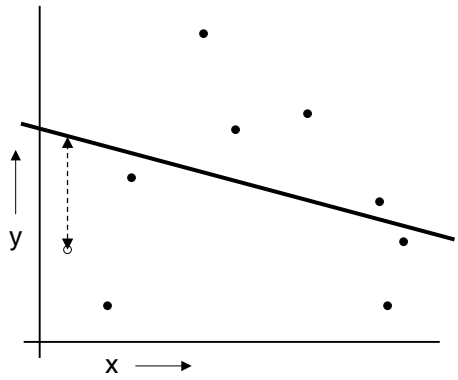
LOOCV (Leave-one-out Cross Validation)



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

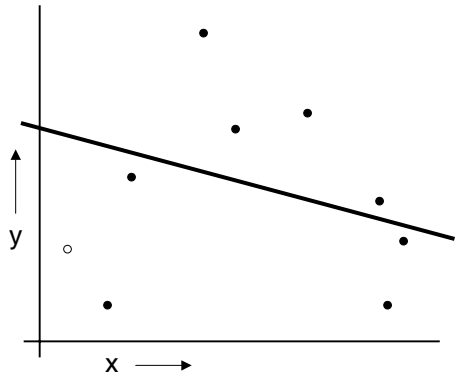
LOOCV (Leave-one-out Cross Validation)



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)

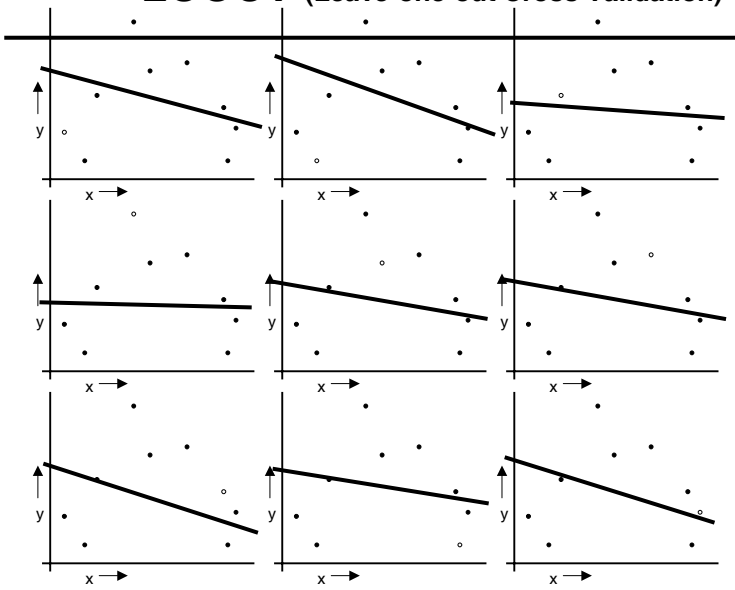


For k=1 to R

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)



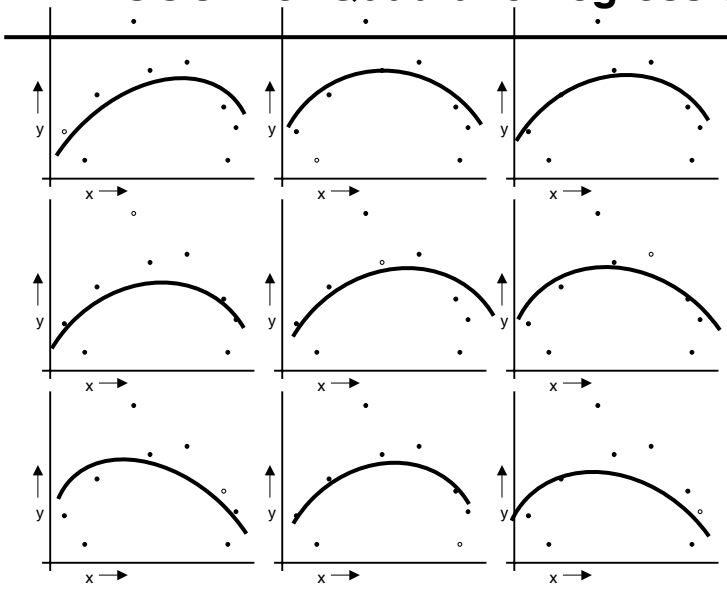
For k=1 to R

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



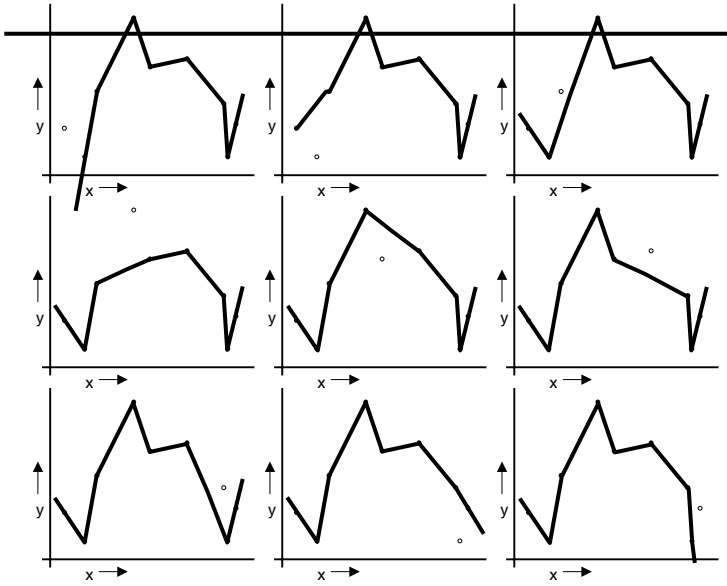
For k=1 to R

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.

$MSE_{LOOCV} = 3.33$

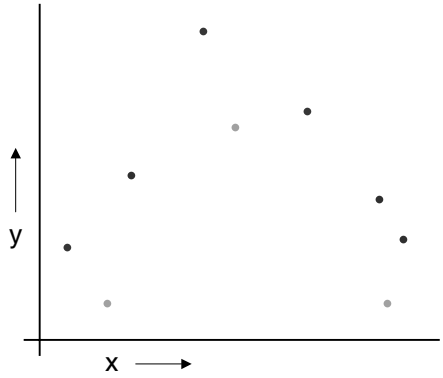
Which kind of Cross Validation?

	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

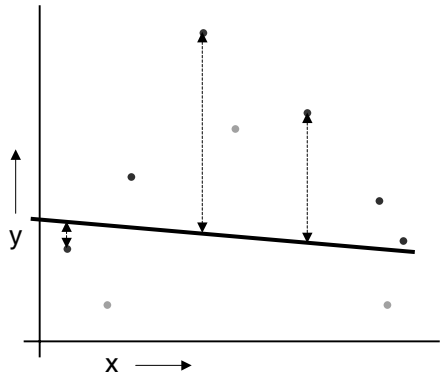
Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)



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.

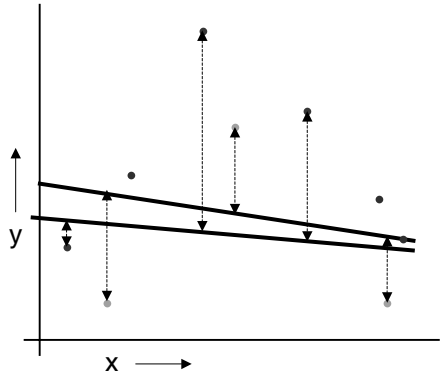


k-fold Cross Validation

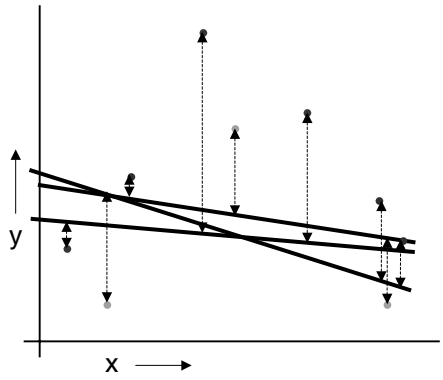
Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

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



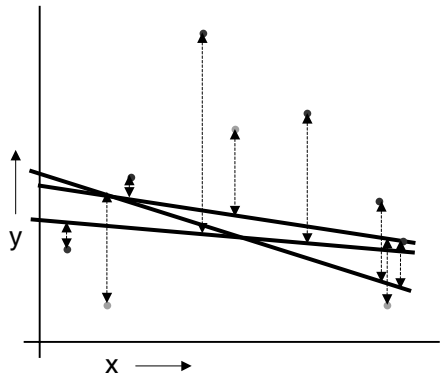
Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

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)

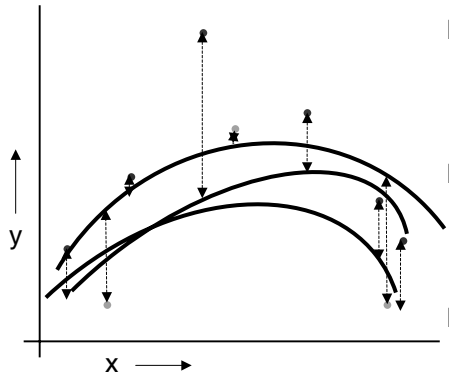
For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

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)

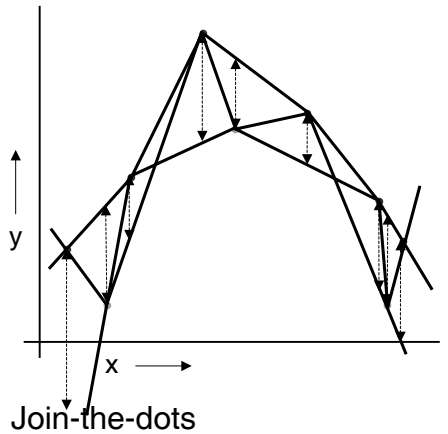
For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points.

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



Join-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.	Wastes 1/R data
7-fold	Wastes 1/7 of the data. 7 times more expensive than test set	Only wastes 1/7. Only 7 times more expensive instead of R times.
3-fold	Wastes 1/3 of the data.	Slightly better than test-set
R-fold	Identical to Leave-one-out	

CV-based Model Selection

Decide for a CV technique, e.g., 3-fold CV:

Linear Regression $MSE_{3FOLD}=2.05$

Quadratic Regression $MSE_{3FOLD}=1.11$

Join-the-dots $MSE_{3FOLD}=2.93$

→ Quadratic Regression is the best method.

Note that for all CV methods, the quadratic regression showed the lowest MSE compared with the other two methods.