*CS9840*
*Learning and Computer Vision*
*Prof. Olga Veksler*
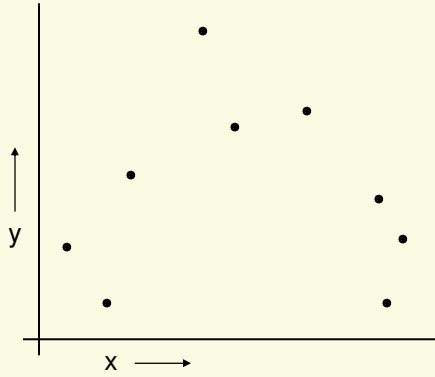
# Lecture 5

## Cross Validation

Cross Validation slides are from Andrew Moore (CMU)

Some slides are due to Robin Dhamankar

Vandi Verma & Sebastian Thrun

---

## *Today*

- New Machine Learning Topics:

  Performance evaluation method: cross-validation
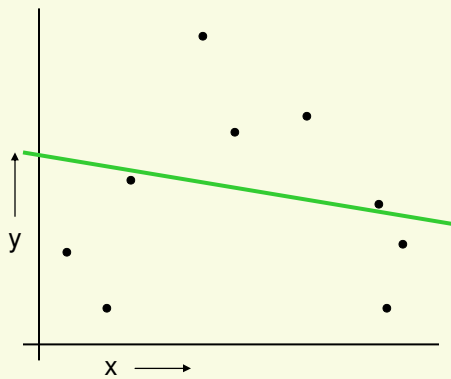
## A Regression Problem

$y = f(x) + noise$

Can we learn f from this data?
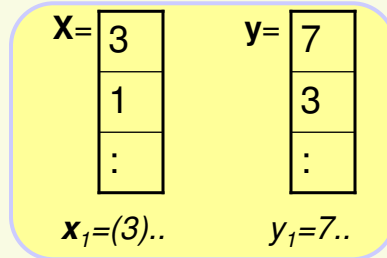
Let's consider three methods…

## Linear Regression

## Linear Regression

Univariate Linear regression with a constant term:

| X | Y |
|---|---|
| 3 | 7 |
| 1 | 3 |
| : | : |

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

$x_1=(3)..$     $y_1=7..$

---

## Linear Regression

Univariate Linear regression with a constant term:

| X | Y |
|---|---|
| 3 | 7 |
| 1 | 3 |
| : | . |

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

$y_1=7..$

$\mathbf{Z}=$
$\begin{array}{cc} 1 & 3 \\ 1 & 1 \\ : & \end{array}$
$\mathbf{y}=$
$\begin{array}{c} 7 \\ 3 \\ : \end{array}$

$z_1=(1,3)..$     $y_1=7..$

$z_k=(1,x_k)$

## Linear Regression

Univariate Linear regression with a constant term:

| X | Y |
|---|---|
| 3 | 7 |
| 1 | 3 |
| : | . |

$\mathbf{X}=\begin{array}{|c|}\hline 3 \\ \hline 1 \\ \hline : \\ \hline\end{array}$  $\mathbf{y}=\begin{array}{|c|}\hline 7 \\ \hline 3 \\ \hline : \\ \hline\end{array}$

$y_1=7..$

$\mathbf{Z}=\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 1 & 1 \\ \hline : & \\ \hline\end{array}$  $\mathbf{y}=\begin{array}{|c|}\hline 7 \\ \hline 3 \\ \hline : \\ \hline\end{array}$

$z_1=(1,3)..$  $y_1=7..$

$z_k=(1,x_k)$

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

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

## Quadratic Regression



$y$

$x \longrightarrow$
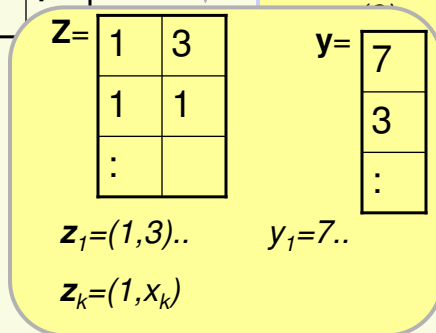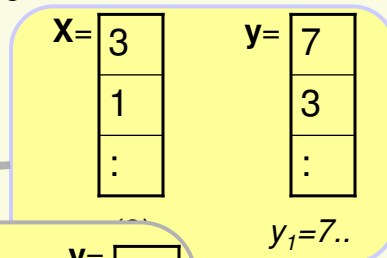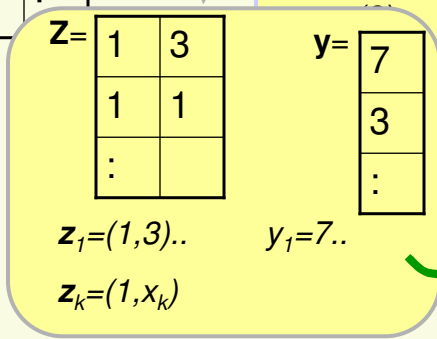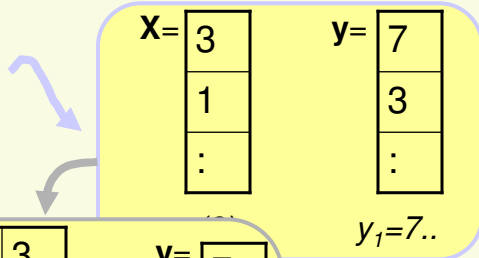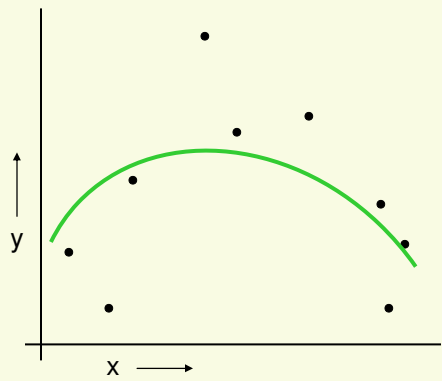
## Quadratic Regression

| X | Y |
|---|---|
| 3 | 7 |
| 1 | 3 |
| : | . |

$\mathbf{X}=$
| 3 |
|---|
| 1 |
| : |

$\mathbf{y}=$
| 7 |
|---|
| 3 |
| : |

$y_1=7..$

$\mathbf{Z}=$
| 1 | 3 | 9 |
|---|---|---|
| 1 | 1 | 1 |
| : | | |

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

$\mathbf{y}=$
| 7 |
|---|
| 3 |
| : |

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

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

**from Andrew Moore (CMU)**

---

## Join-the-dots
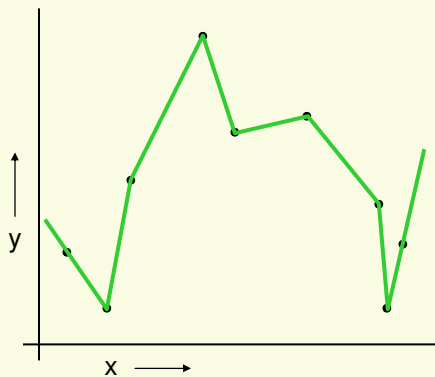


Also known as piecewise linear nonparametric regression if that makes you feel better
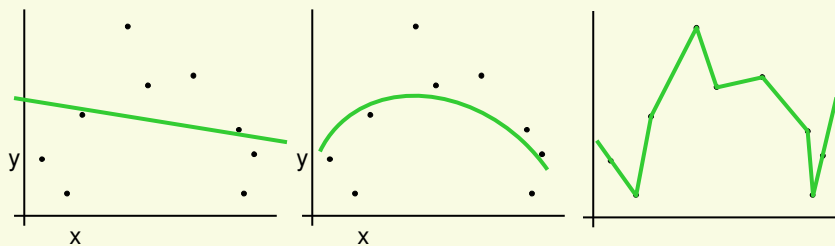
**from Andrew Moore (CMU)**

## Which is best?
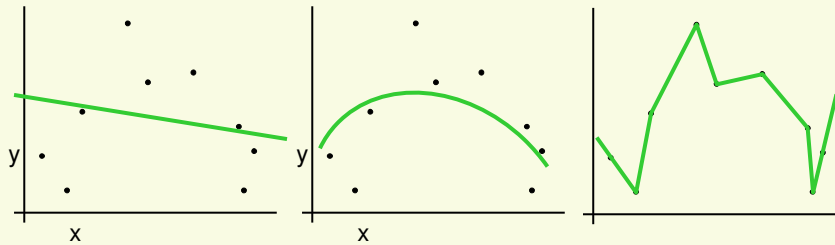


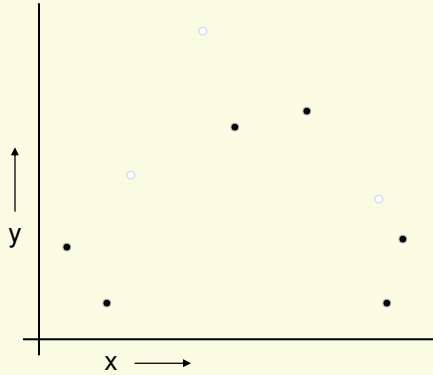Why not choose the method with the best fit to the data?

## 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?"
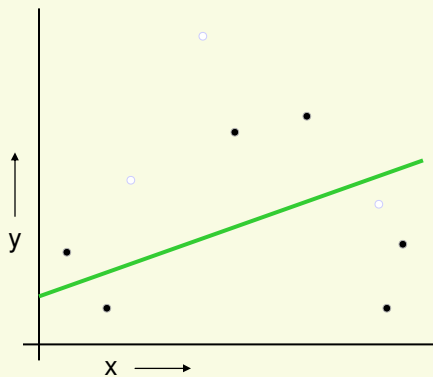
## The test set method



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

2. The remainder is a training set

## The test set method



(Linear regression example)
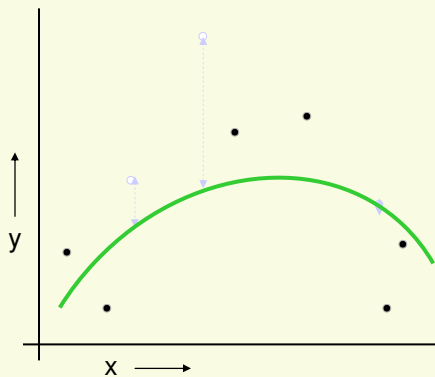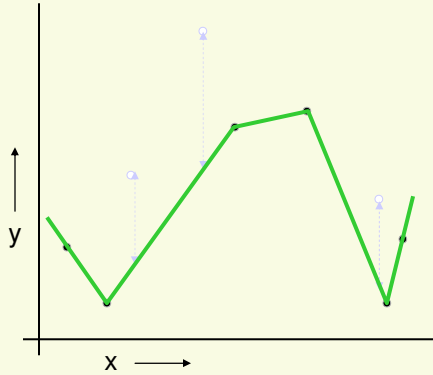
1. Randomly choose 30% of the data to be in a test set

2. The remainder is a training set

3. Perform your regression on the training set

## 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
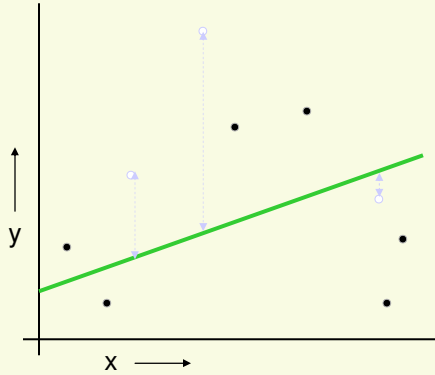
(Linear regression example)

Mean Squared Error = 2.4

from Andrew Moore (CMU)

## 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

from Andrew Moore (CMU)

## 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
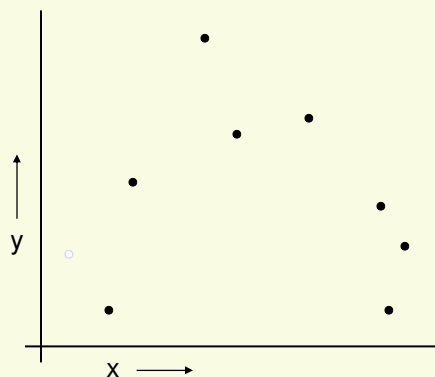
(Join the dots example)

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"
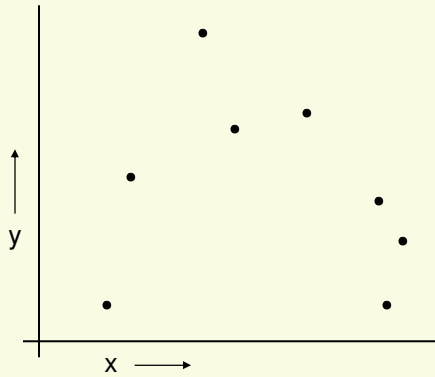
---

## *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
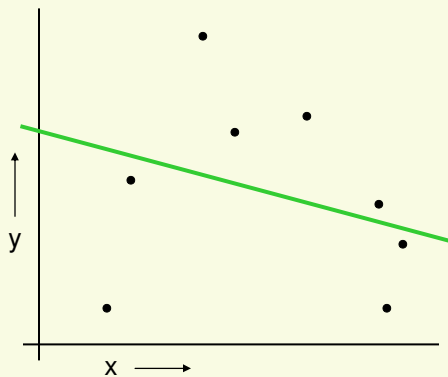
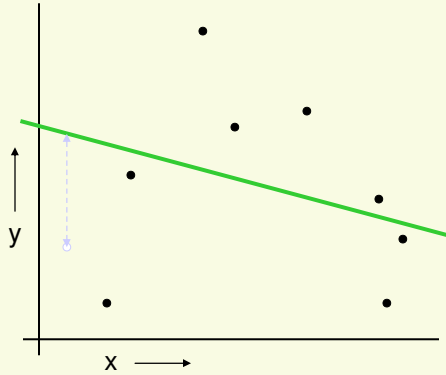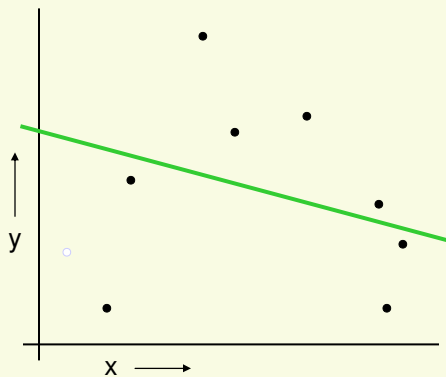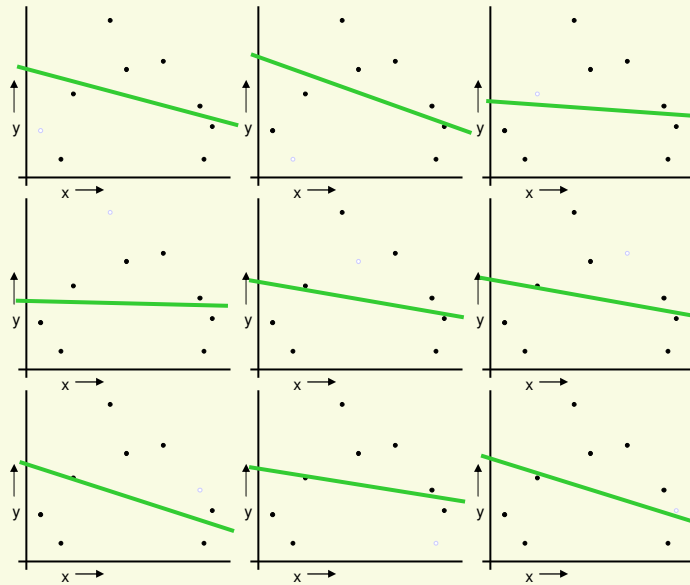## LOOCV *(Leave-one-out Cross Validation)*

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

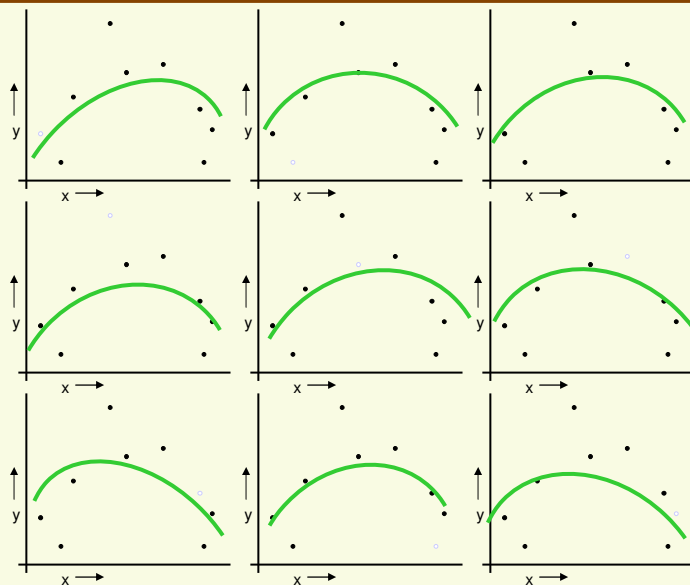## LOOCV *(Leave-one-out Cross Validation)*



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 $(x_k, y_k)$

**from Andrew Moore (CMU)**

## LOOCV *(Leave-one-out Cross Validation)*
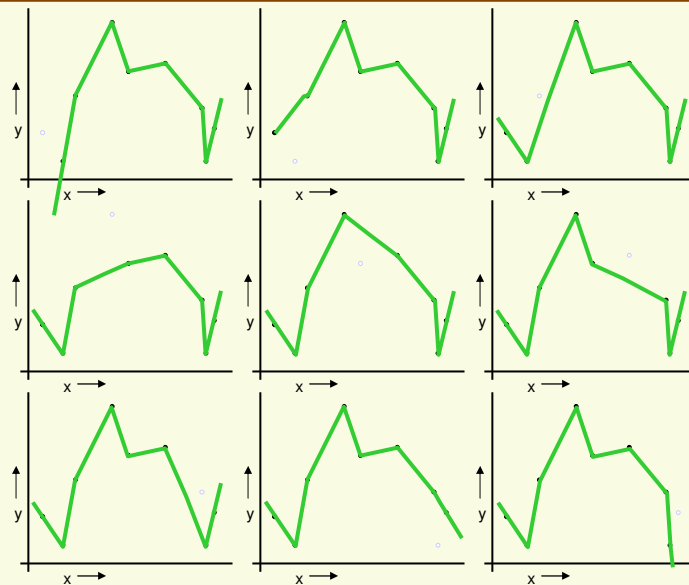


For k=1 to R

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.

**from Andrew Moore (CMU)**

# 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.

$MSE_{LOOCV} = 2.12$

**from Andrew Moore (CMU)**

---

# 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.

$MSE_{LOOCV} = 0.962$

**from Andrew Moore (CMU)**

## 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.

$MSE_{LOOCV} = 3.33$

**from Andrew Moore (CMU)**
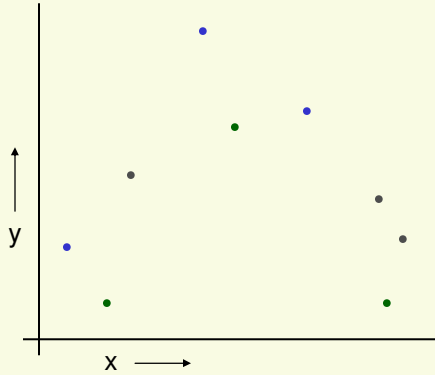
## Which kind of Cross Validation?

|  | Downside | Upside |
|---|---|---|
| **Test-set** | Variance: unreliable estimate of future performance | Cheap |
| **Leave-one-out** | Expensive | Doesn't waste data |

..can we get the best of both worlds?

**from Andrew Moore (CMU)**

## k-fold Cross Validation

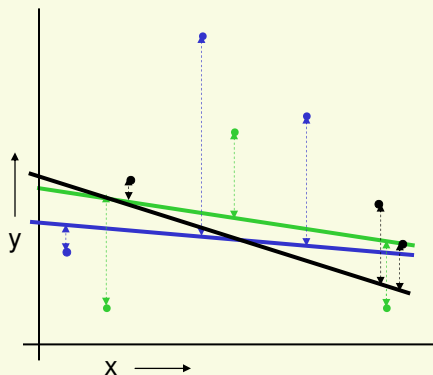Randomly break the dataset into k partitions (in our example we'll have k=3 partitions colored Red Green and Blue)

## 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 blue partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue points.
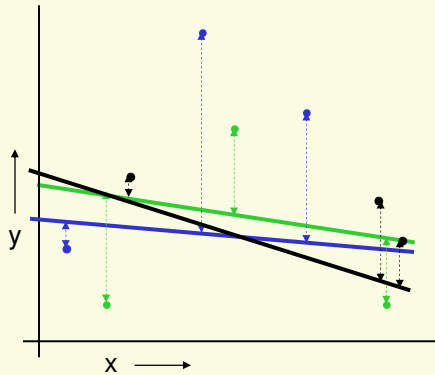
*15*

## 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 blue partition: Train on all the points not in the blue partition. Find the test-set sum of errors on the blue 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.
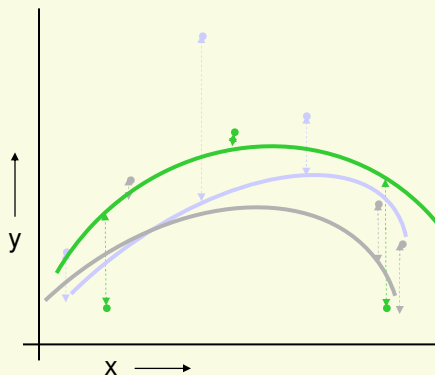
## k-fold Cross Validation

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

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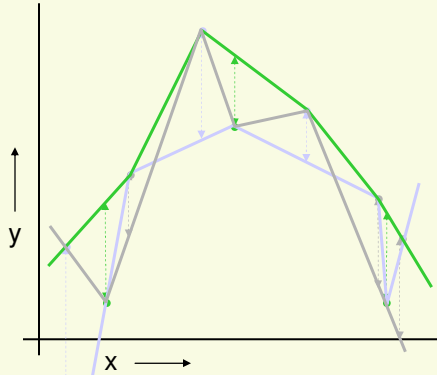
For the gray partition: Train on all the points not in the gray partition. Find the test-set sum of errors on the gray points.

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

Then report the mean error

Linear Regression
$MSE_{3FOLD}=2.05$

---

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

Then report the mean error

Quadratic Regression
$MSE_{3FOLD}=1.11$

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

Then report the mean error

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

## Which kind of Cross Validation?

| | Downside | Upside |
|---|---|---|
| **Test-set** | Variance: unreliable estimate of future performance | Cheap |
| **Leave-one-out** | Expensive | 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 |
| **N-fold** | Identical to Leave-one-out | |

# 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 | $f_6$ | | | |

# CV-based Model Selection

- Example: Choosing number of hidden units in a one-hidden-layer neural net.
- Step 1: Compute 10-fold CV error for six different model

| Algorithm | TRAINERR | 10-FOLD-CV-ERR | Choice |
|-----------|----------|----------------|--------|
| 0 hidden units | | | |
| 1 hidden units | | | |
| 2 hidden units | | | √ |
| 3 hidden units | | | |
| 4 hidden units | | | |
| 5 hidden units | | | |

- Step 2: Whichever model class gave best CV score: train it with all the data, and that's the predictive model you'll use.

## CV-based Model Selection

- Example: Choosing "k" for a k-nearest-neighbor regression.
- Step 1: Compute LOOCV error for six different model classes:

| Algorithm | TRAINERR | 10-fold-CV-ERR | Choice |
|-----------|----------|----------------|--------|
| K=1 | | | |
| K=2 | | | |
| K=3 | | | |
| K=4 | | | √ |
| K=5 | | | |
| K=6 | | | |

- Step 2: Whichever model class gave best CV score: train it with all the data, and that's the predictive model you'll use.

---

## CV-based Model Selection

- Example: Choosing "k" for a k-nearest-neighbor regression.
- Step 1: Com... classes:

> Why did we use 10-fold-CV for neural nets and LOOCV for k-nearest neighbor?
>
> And why stop at K=6
>
> Are we guaranteed that a local optimum of K vs LOOCV will be the global optimum?
>
> What should we do if we are depressed at the expense of doing LOOCV for K= 1 through 1000?

> The reason is Computational. For k-NN (and all other nonparametric methods) LOOCV happens to be as cheap as regular predictions.

> No good reason, except it looked like things were getting worse as K was increasing

> Sadly, no. And in fact, the relationship can be very bumpy.

> Idea One: K=1, K=2, K=4, K=8, K=16, K=32, K=64 … K=1024
>
> Idea Two: Hillclimbing from an initial guess at K

| Algorithm | T |
|-----------|---|
| K=1 | |
| K=2 | |
| K=3 | |
| K=4 | √ |
| K=5 | |
| K=6 | |

- Step 2: Whichever model class gave best CV score: train it with all the data, and that's the predictive model you'll use.

## CV-based Model Selection

- Can you think of other decisions we can ask Cross Validation to make for us, based on other machine learning algorithms in the class so far?

## CV-based Algorithm Choice

- Example: Choosing which regression algorithm to use
- Step 1: Compute 10-fold-CV error for six different model classes:

| Algorithm | TRAINERR | 10-fold-CV-ERR | Choice |
|-----------|----------|----------------|--------|
| 1-NN | | | |
| 10-NN | | | |
| Linear Reg'n | | | |
| Quad reg'n | | | √ |
| LWR, KW=0.1 | | | |
| LWR, KW=0.5 | | | |

- Step 2: Whichever algorithm gave best CV score: train it with all the data, and that's the predictive model you'll use.

## *Cross-validation for classification*

- Instead of computing the sum squared errors on a test set, you should compute…

## *Cross-validation for classification*

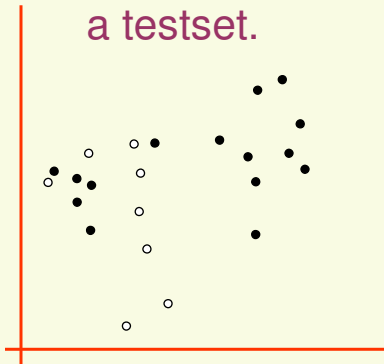- Instead of computing the sum squared errors on a test set, you should compute…

  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.

- What's LOOCV of 1-NN?
- What's LOOCV of 3-NN?
- What's LOOCV of 22-NN?

## *Cross-Validation for classification*

- Choosing k for k-nearest neighbors
- Choosing Kernel parameters for SVM
- Any other "free" parameter of a classifier
- Choosing which classifier to use
- Choosing Features to use