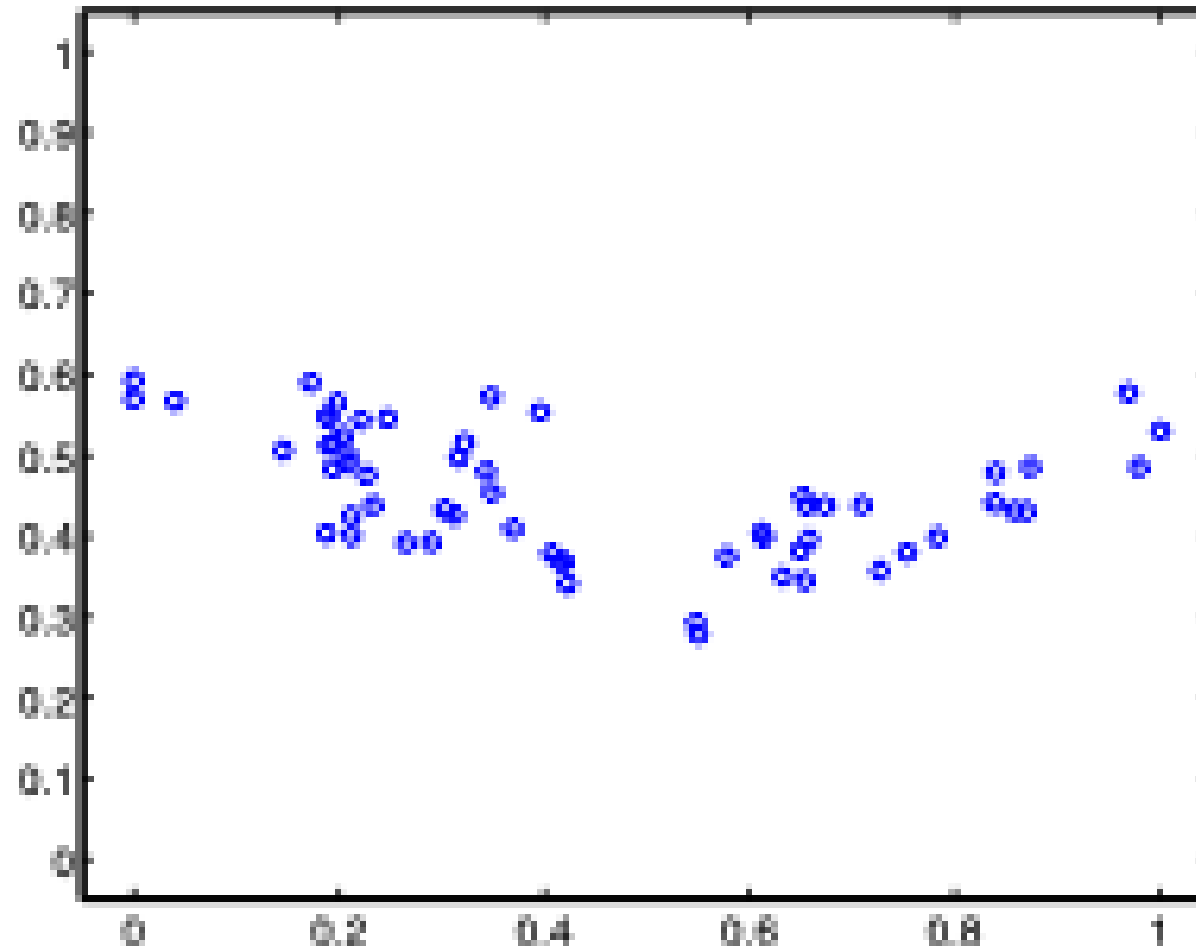


# **Bias-Variance Tradeoff and Model Selection**

**Naresh Manwani**  
**Machine Learning Lab**  
**IIIT-H**

**Example1: We want to fit a curve for the following data**  
**!**



## Example1: continue

- Here we want to fit a polynomial of degree  $p$  as follows.

$$y = w_0 + w_1x + w_2x^2 + \dots + w_px^p$$

- Training data =  $\{(x_1, y_1), \dots, (x_N, y_N)\}$

- Test data =  $\{(x_{N+1}, y_{N+1}), \dots, (x_M, y_M)\}$

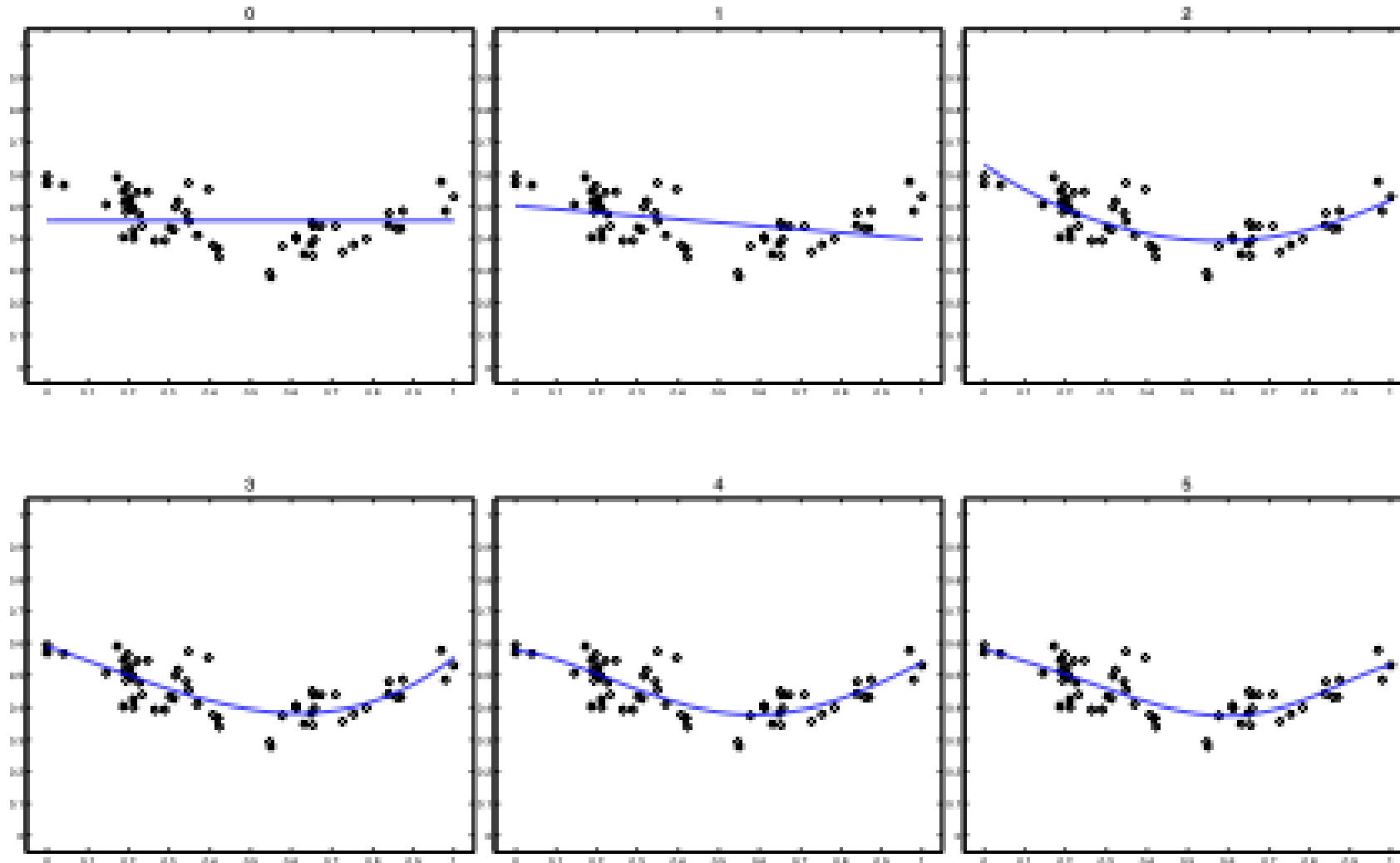
- Objective function:

$$\text{Training Error} = \frac{1}{2} \sum_{i=1}^N (w_0 + w_1x_i + w_2x_i^2 + \dots + w_px_i^p - y_i)^2$$

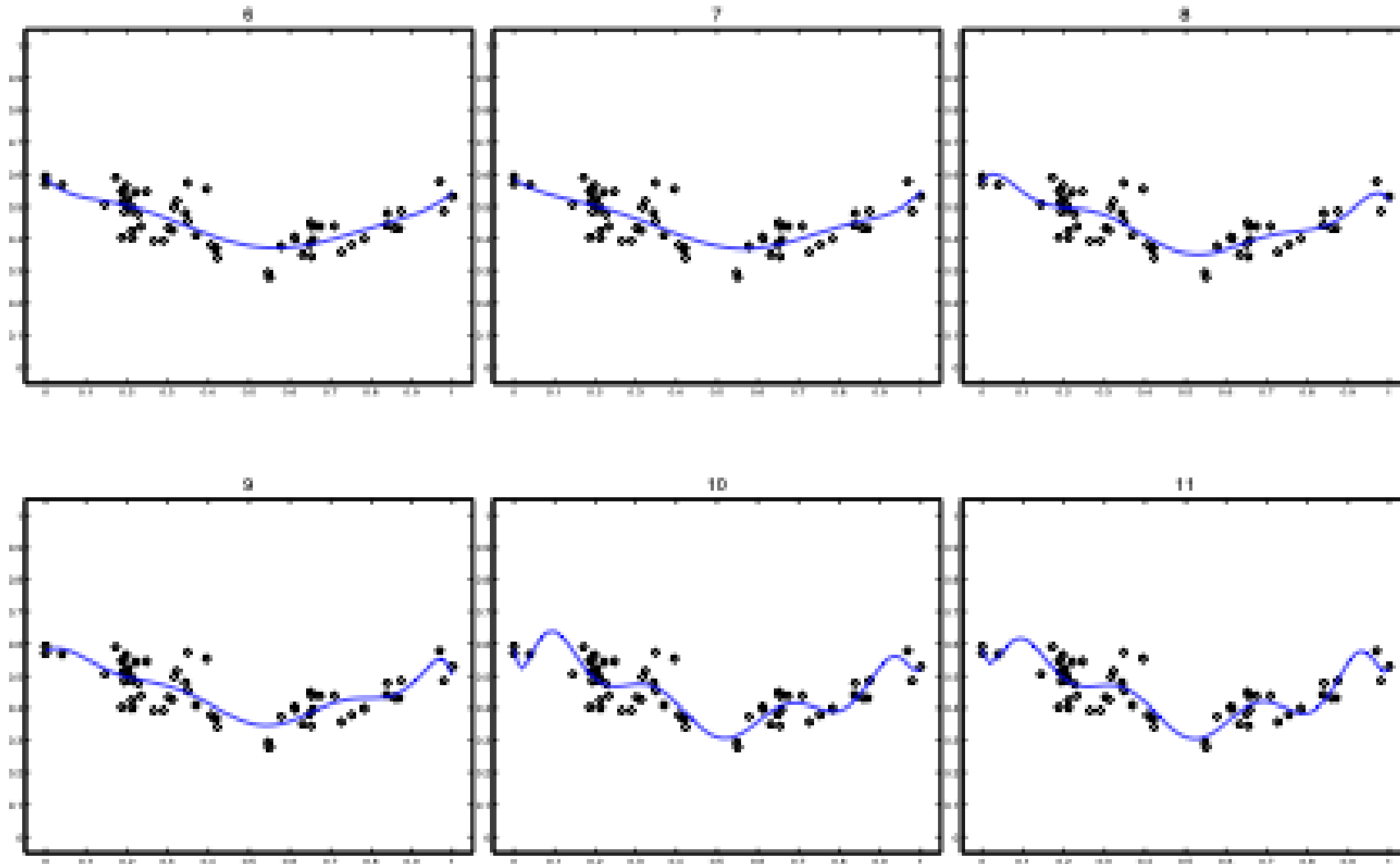
## Performance on unseen data

$$\text{Test Error} = \frac{1}{2} \sum_{i=N+1}^M (w_0 + w_1 x_i + w_2 x_i^2 + \dots + w_p x_i^p - y_i)^2$$

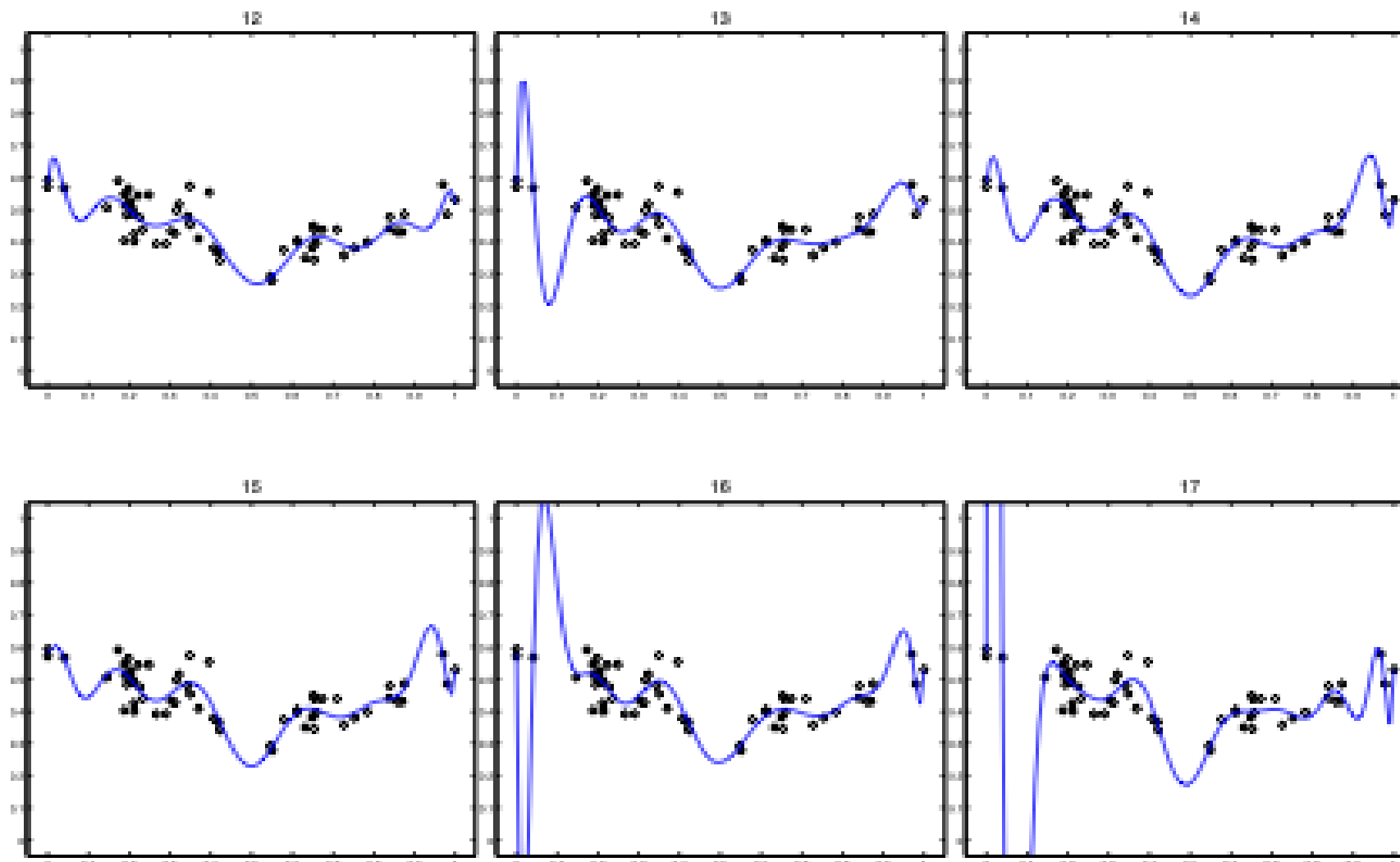
## Example1: Fitted curve for $p=0,1,2,3,4,5$



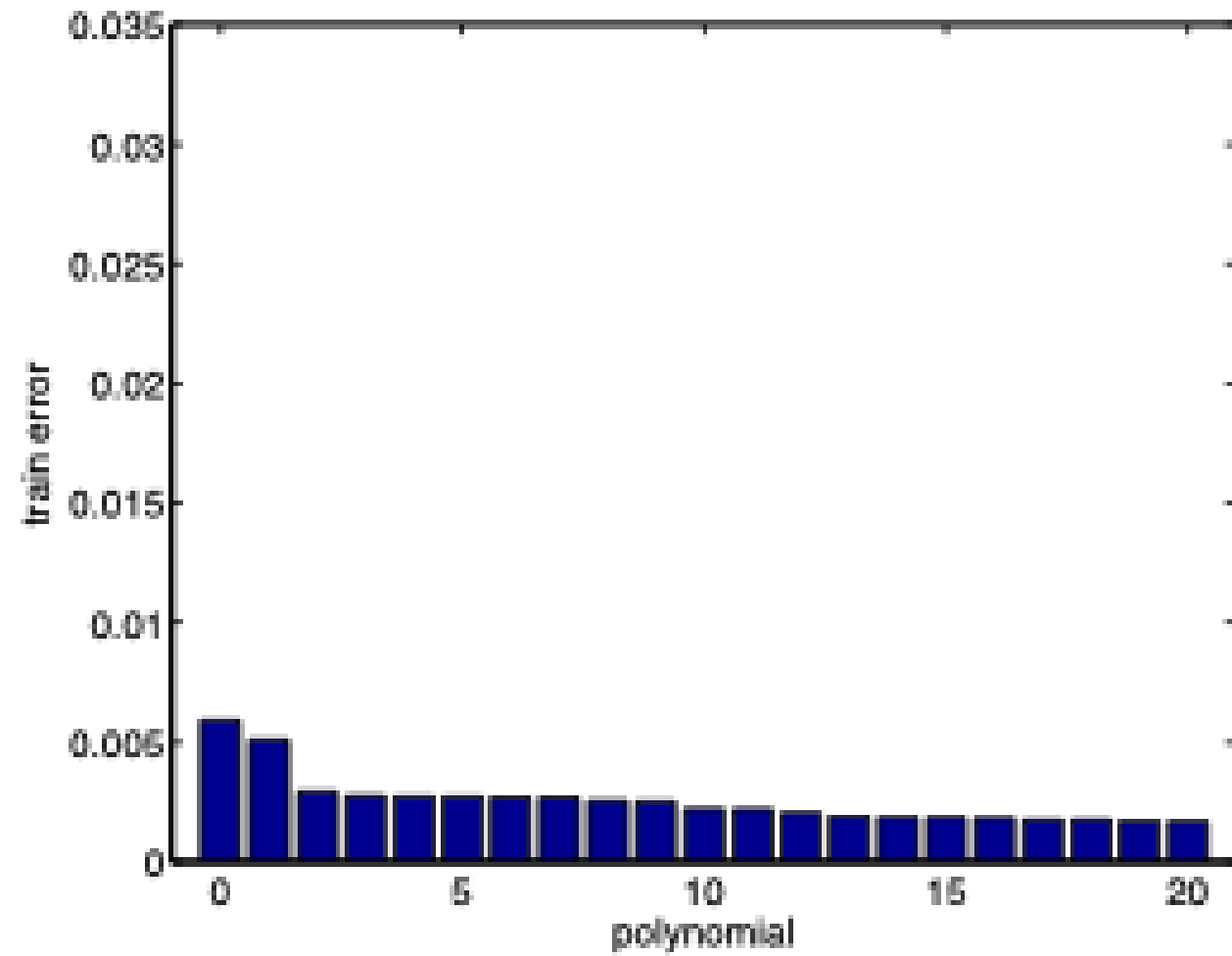
## Example1: Fitted curve for $p=6,7,8,9,10,11$



## Example1: Fitted curve for $p=12,13,14,15,16,17$

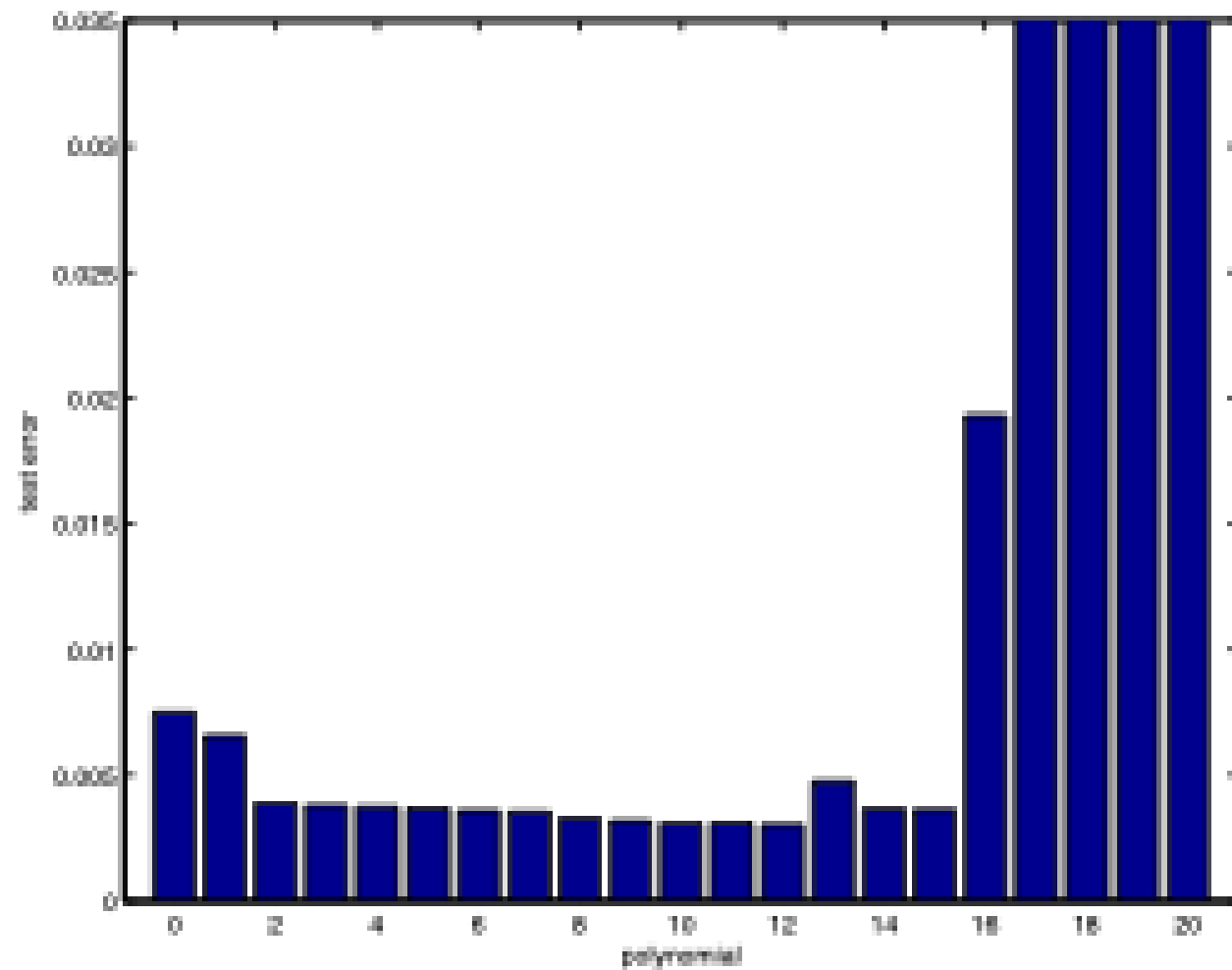


## Example1: Training Error





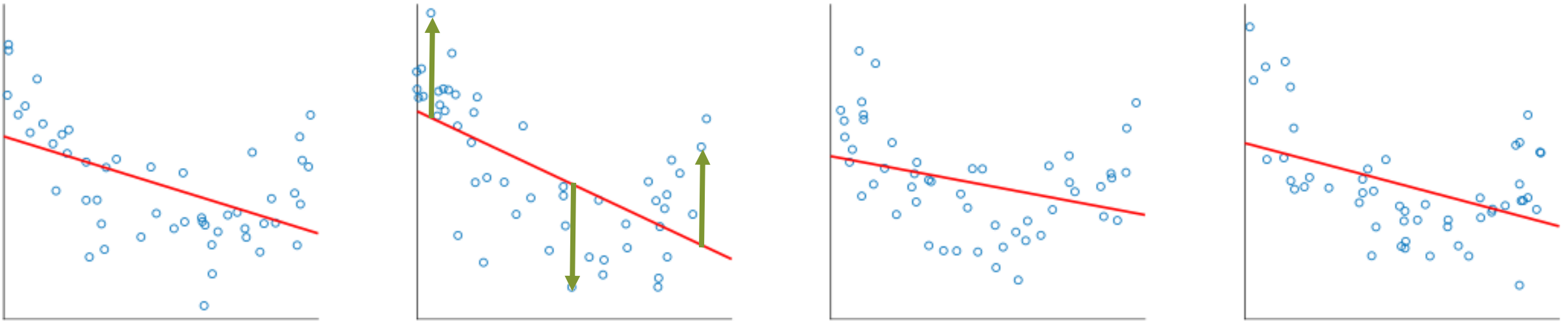
# Example1: Test Error



# Bias Variance Tradeoff

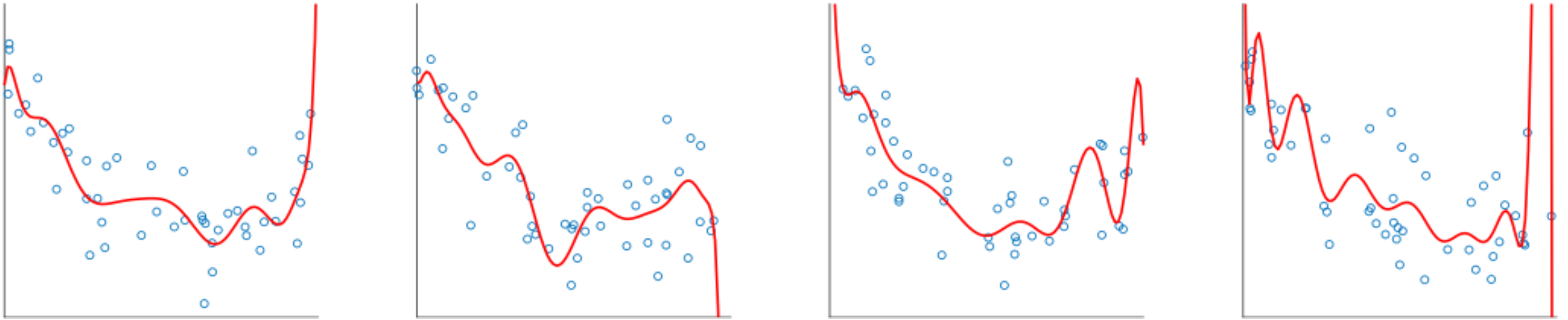
- For very low  $p$ , the model is very simple, and so can't capture the full complexities of the data. It “underfits” the data. This is called **bias**.
- For very high  $p$ , the model is complex, and so tends to “overfit” to spurious properties of the data. This is called **variance**.

## Example2: Bias



**Linear model learnt on different training samples. Regardless of training sample, or size of training sample, model will produce consistent errors**

## Example2: Variance



Keeping the degree  $p$  very high. Different samples of training data yield different model fits

# Formalizing Bias and Variance

Given data set

- $\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\}$

And model built from data set,

- $f(x; \mathcal{D})$

We can evaluate the effectiveness of the model using mean squared error:

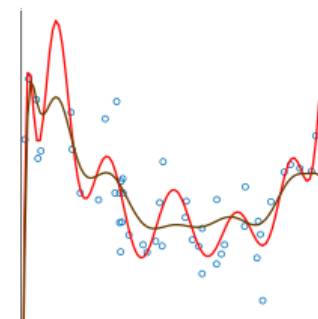
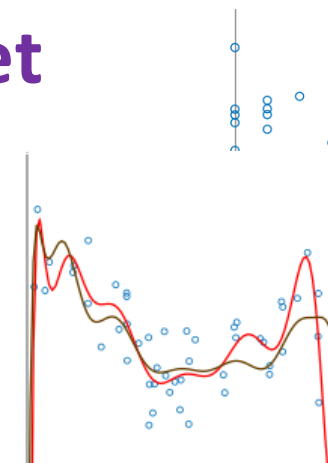
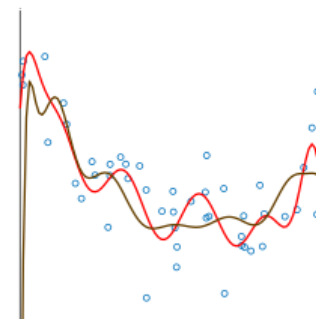
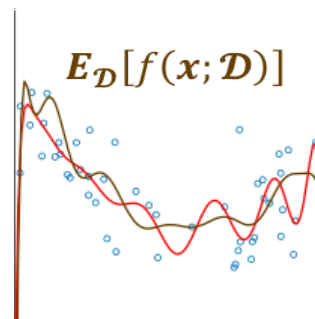
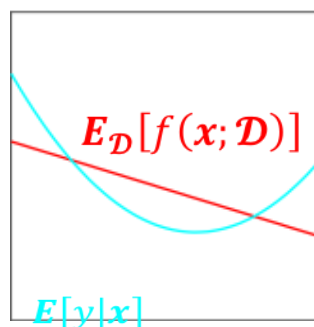
- $\text{MSE} = E_{p(x,y,\mathcal{D})} \left[ (y - f(x; \mathcal{D}))^2 \right]$

- with constant  $|\mathcal{D}| = N$

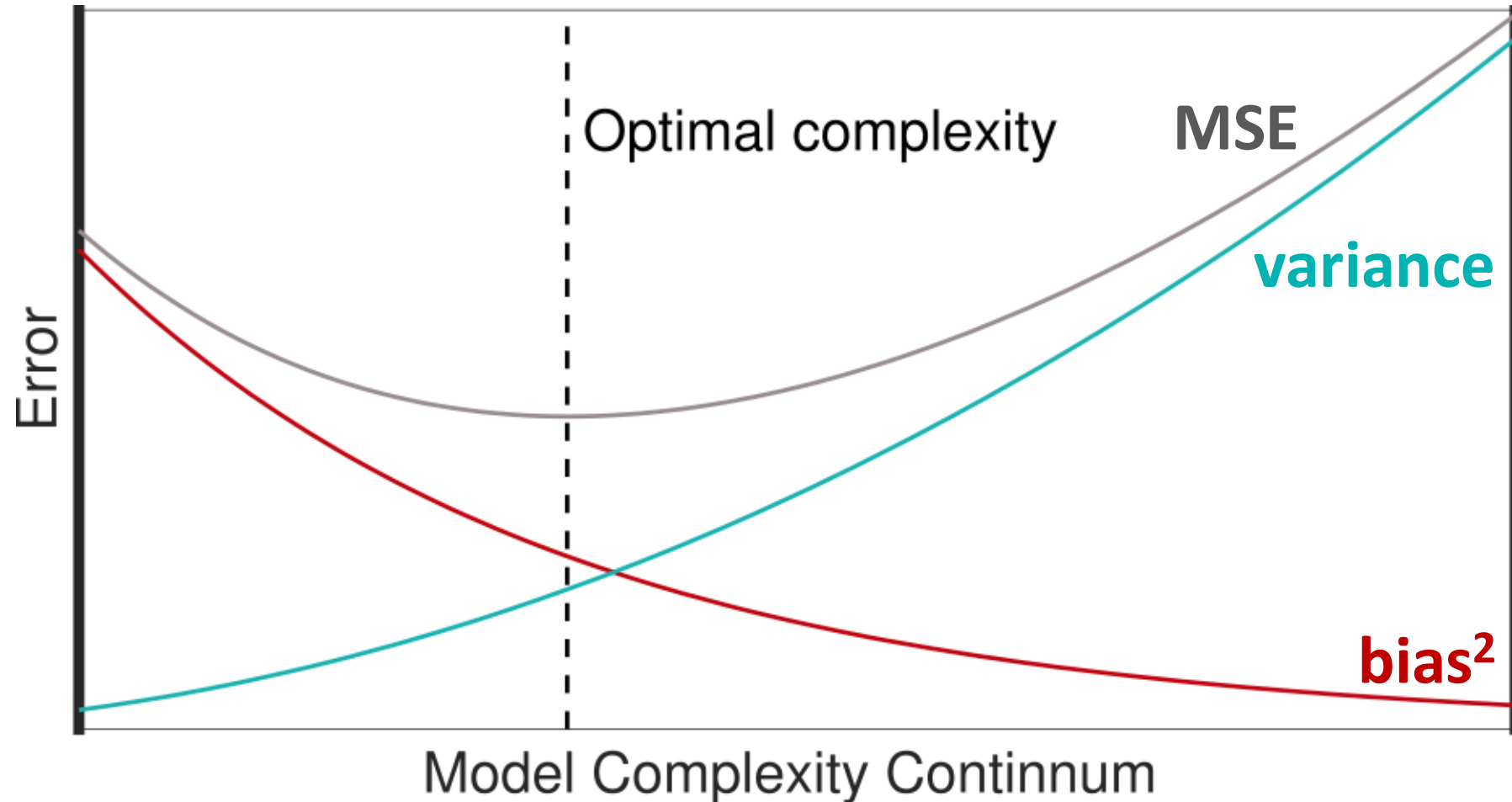
$$\text{MSE}_x = E_{\mathcal{D}|x} \left[ (y - f(x; \mathcal{D}))^2 \right]$$

**bias:**  
**variance of models**  
**(across datasets)**  
**for given points**  
**data sets) and the**  
**target**

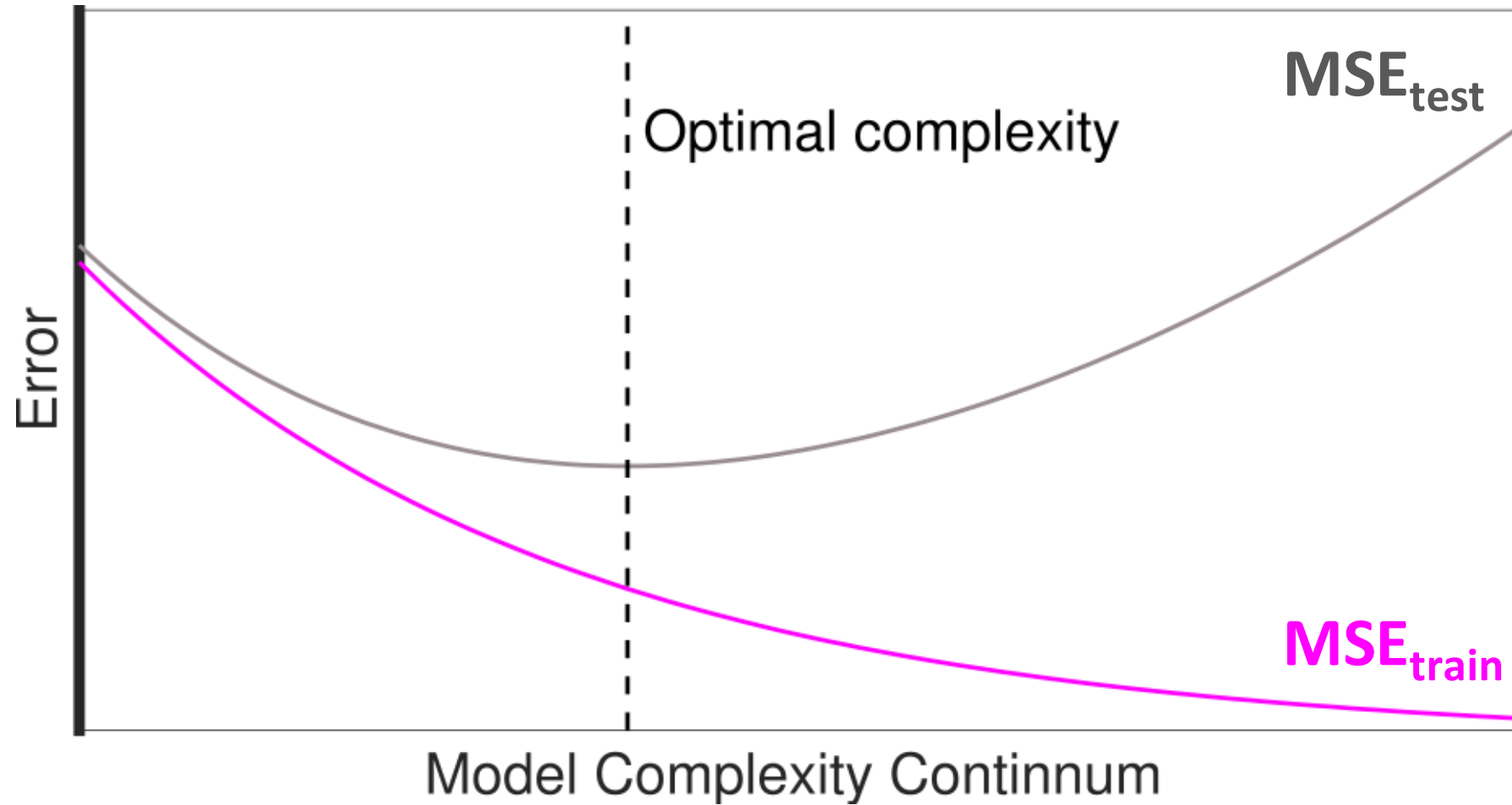
intrinsic noise in data set



# Bias-Variance Trade Off

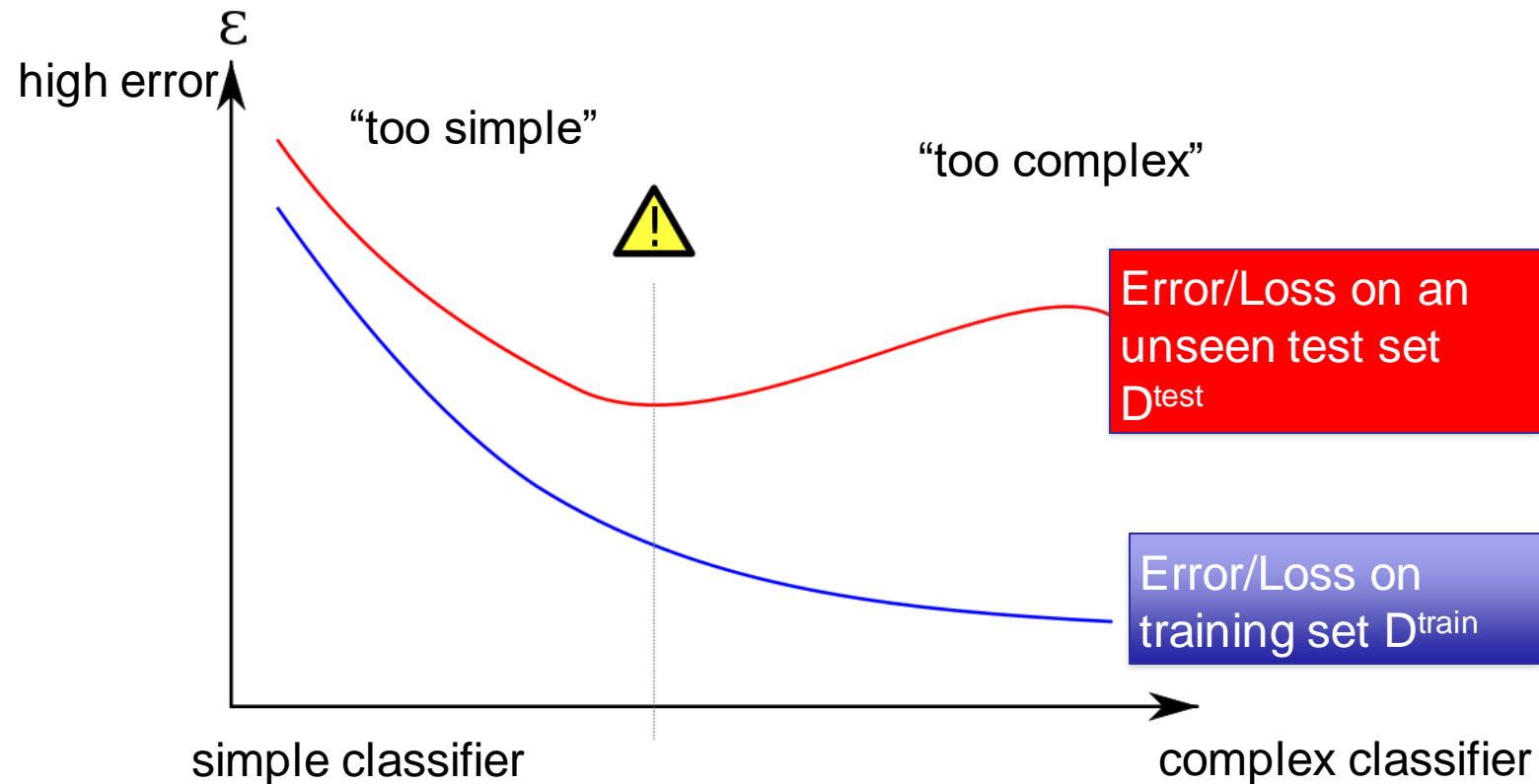


# Bias-Variance Trade Off Is Revealed Via Test Set Not Training Set





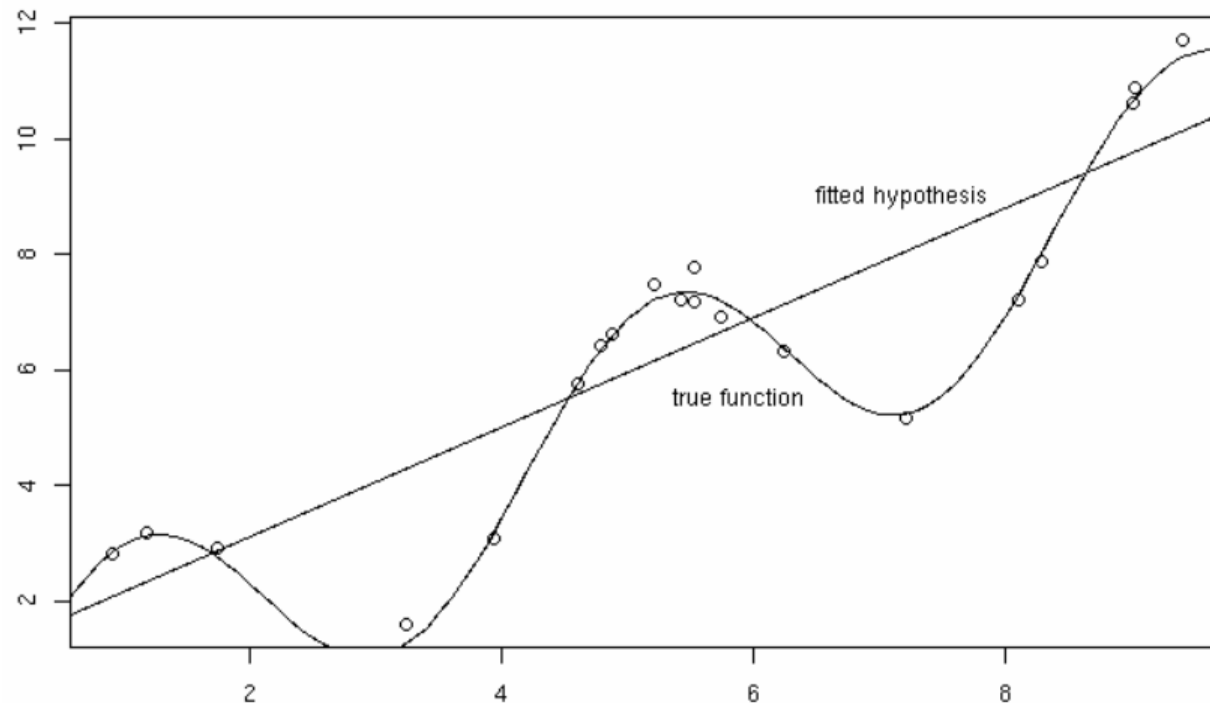
# Bias/Variance is a Way to Understand Overfitting and Underfitting



# Bias-Variance: Another Example

# Example

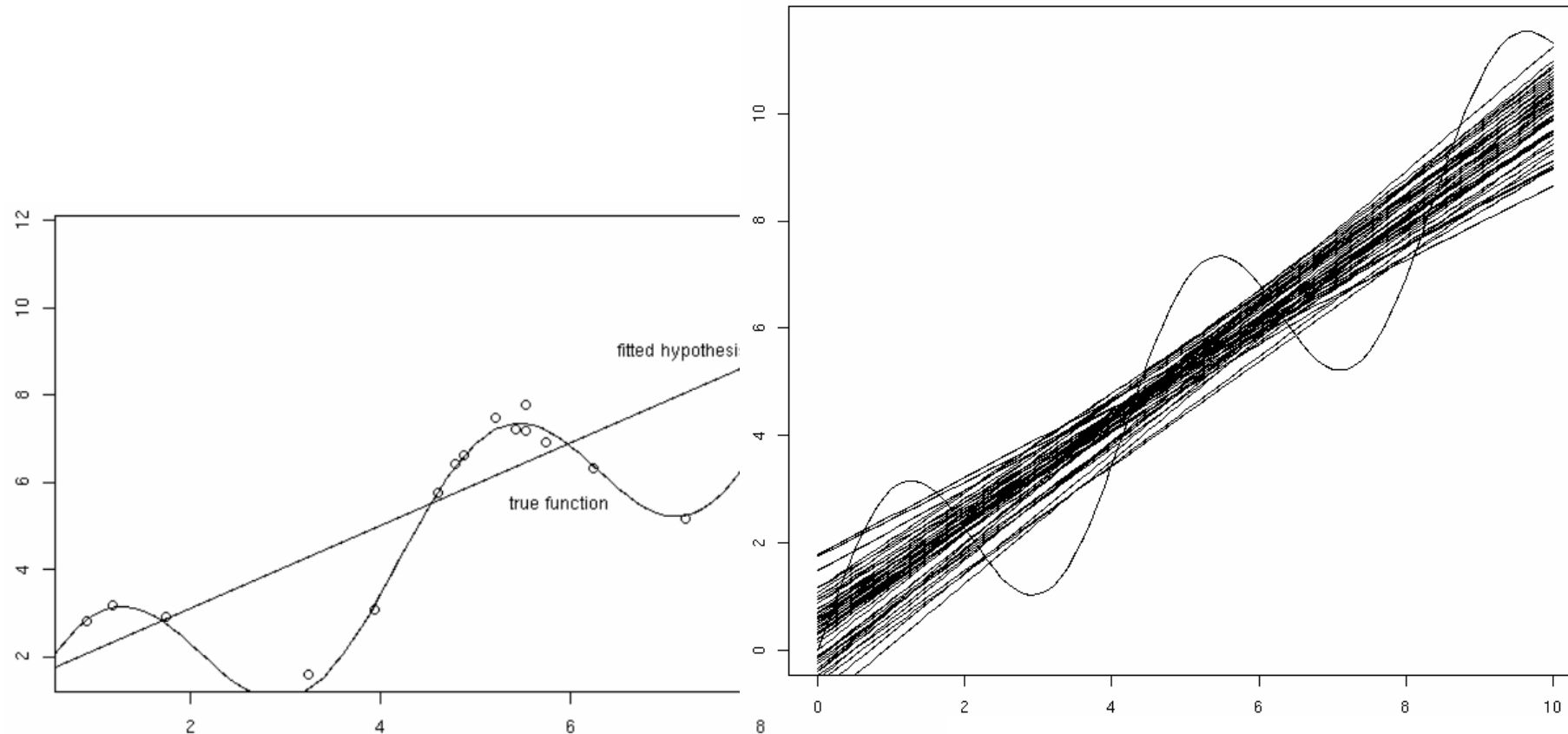
Tom Dietterich, Oregon St



$$y = x + 2 \sin(1.5x) + N(0,0.2)$$

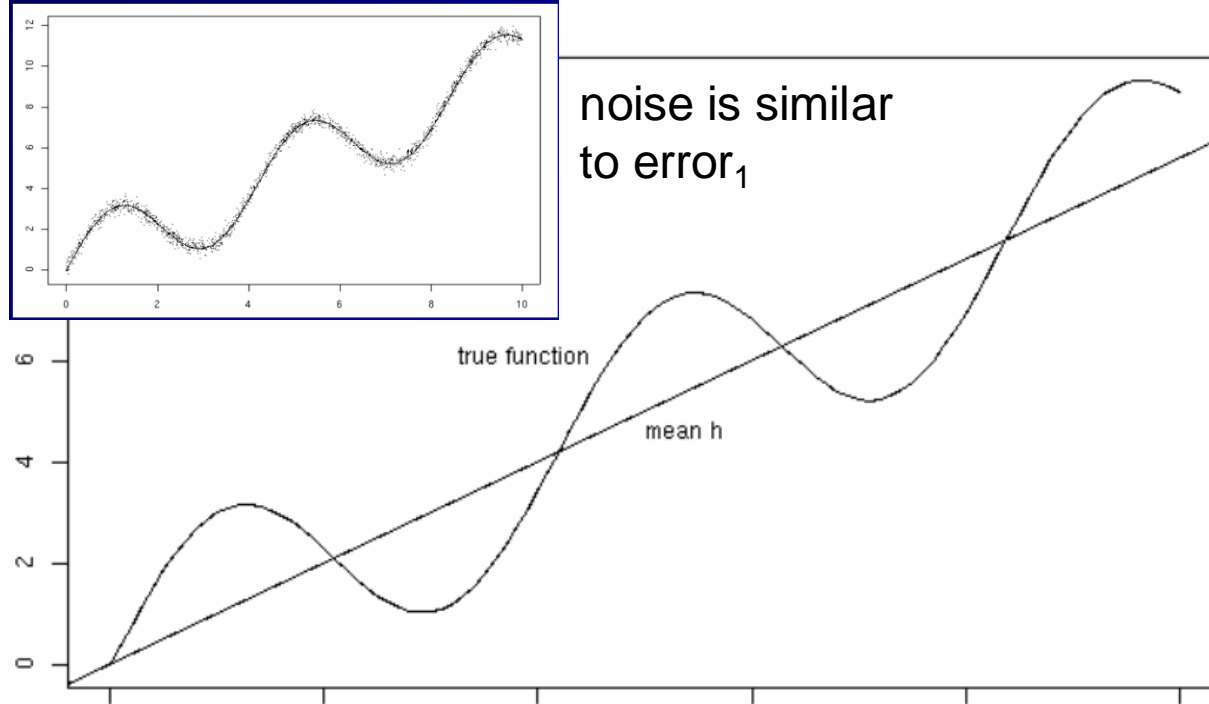
# Example

Tom Dietterich, Oregon St



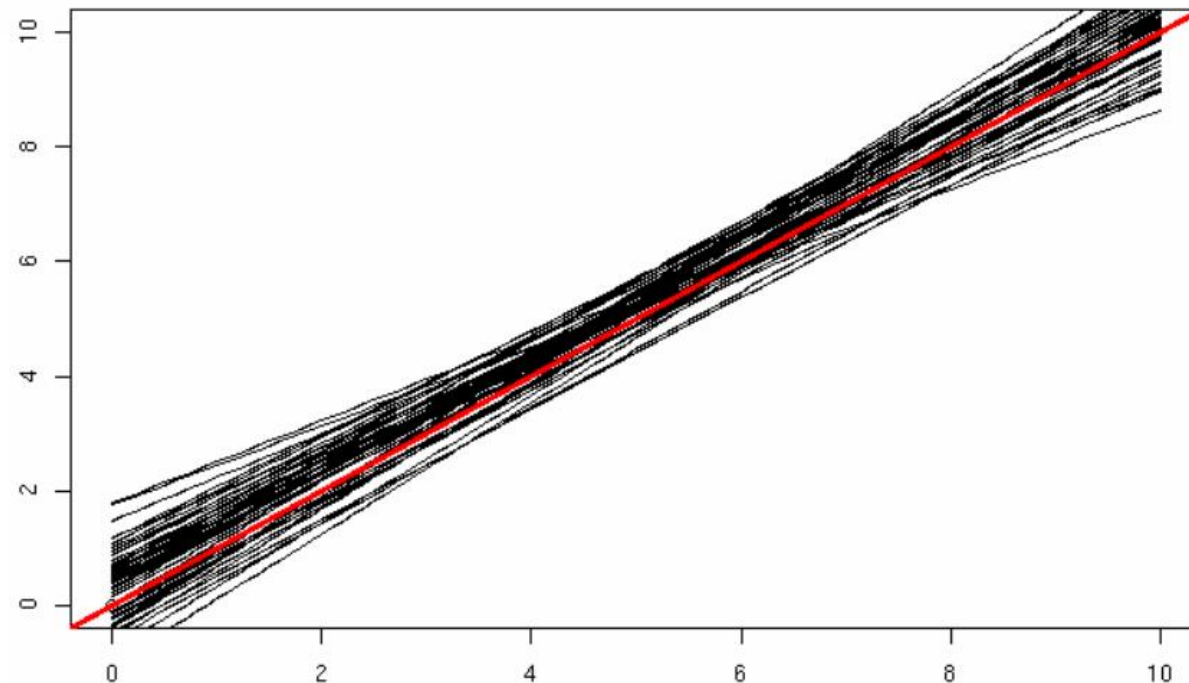
$$y = x + 2 \sin(1.5x) + N(0,0.2)$$

Same experiment, repeated:  
with 50 samples of 20 points each



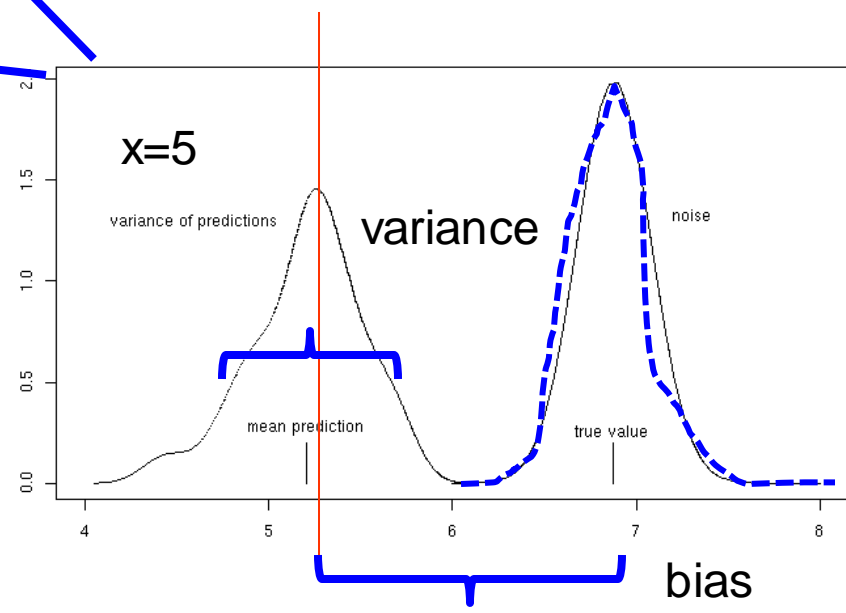
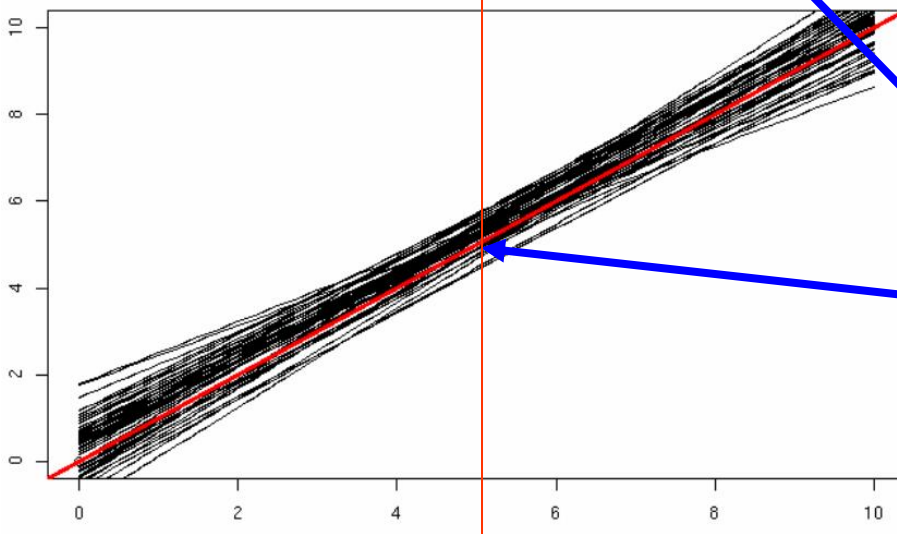
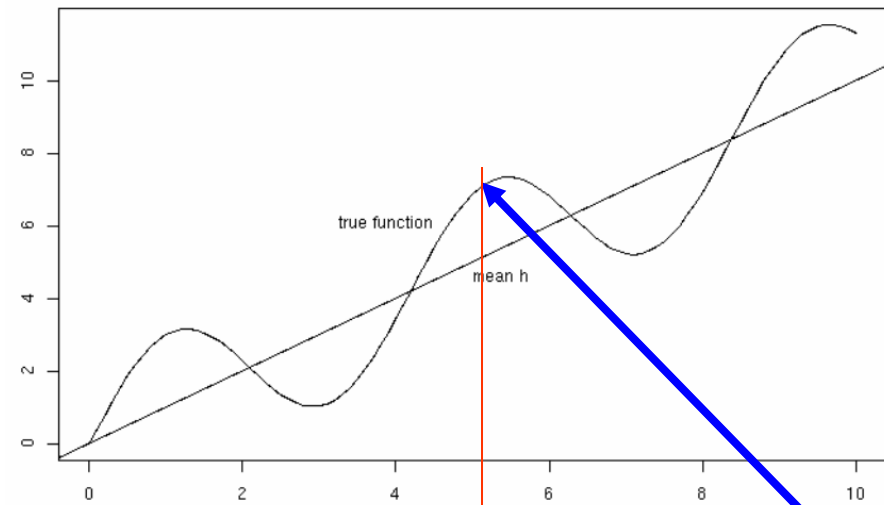
The true function  $f$  can't be fit perfectly with hypotheses from our class  $H$  (lines) ⑨  $\text{Error}_1$

Fix: *more* expressive set of hypotheses  $H$



We don't get the best hypothesis from  $H$  because of noise/small sample size ⑨  $\text{Error}_2$

Fix: *less* expressive set of hypotheses  $H$



# CROSS-VALIDATION AND MODEL SELECTION

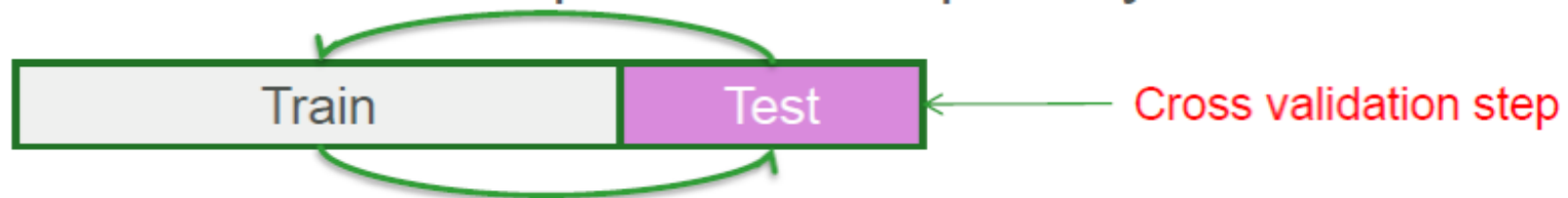
Many Slides are from: Dr. Thomas Jensen -Expedia.com and  
*Prof. Olga Veksler - CS9840 - Learning and Computer Vision*

# CROSS VALIDATION – THE IDEAL PROCEDURE

1. Divide data into three sets, training, validation and test sets



2. Find the optimal model on the training set, and use the test set to check its predictive capability



3. See how well the model can predict the test set



4. The validation error gives an unbiased estimate of the predictive power of a model



# TRAINING/TEST DATA SPLIT

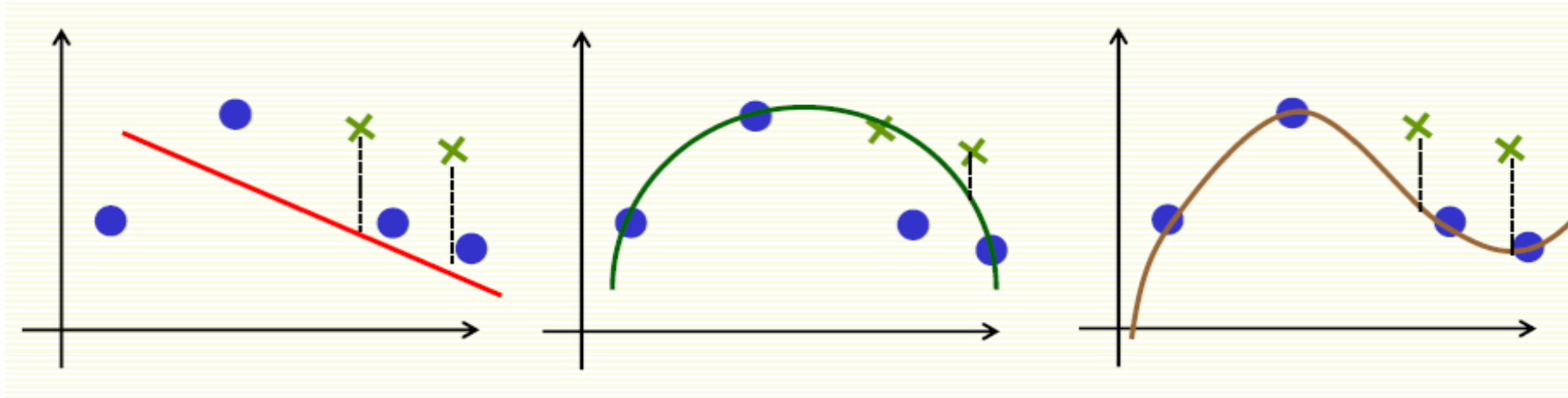
Talked about splitting data in training/test sets

- training data is used to fit parameters
- test data is used to assess how classifier generalizes to new data

What if classifier has “non-tunable” parameters?

- a parameter is “non-tunable” if tuning (or training) it on the training data leads to overfitting

# TRAINING/TEST DATA SPLIT



What about test error? Seems appropriate

- degree 2 is the best model according to the test error

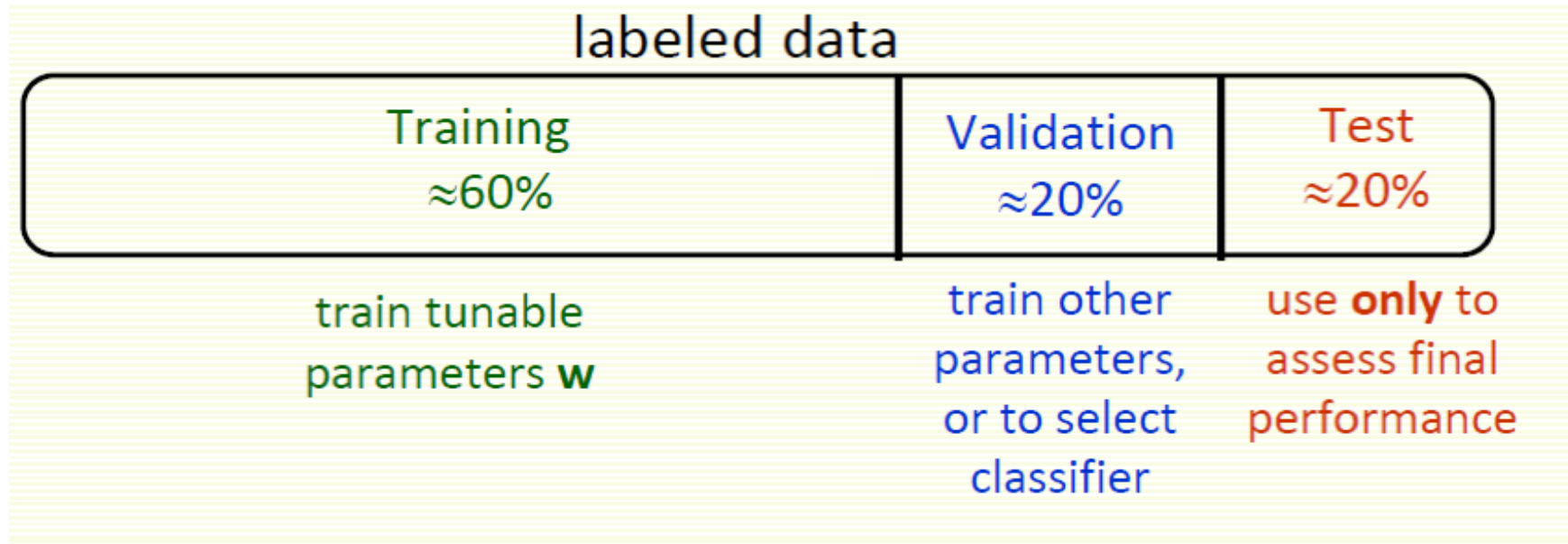
Except what do we report as the test error now?

- Test error should be computed on data that was **not used for training at all**
- Here used “test” data for training, i.e. choosing model

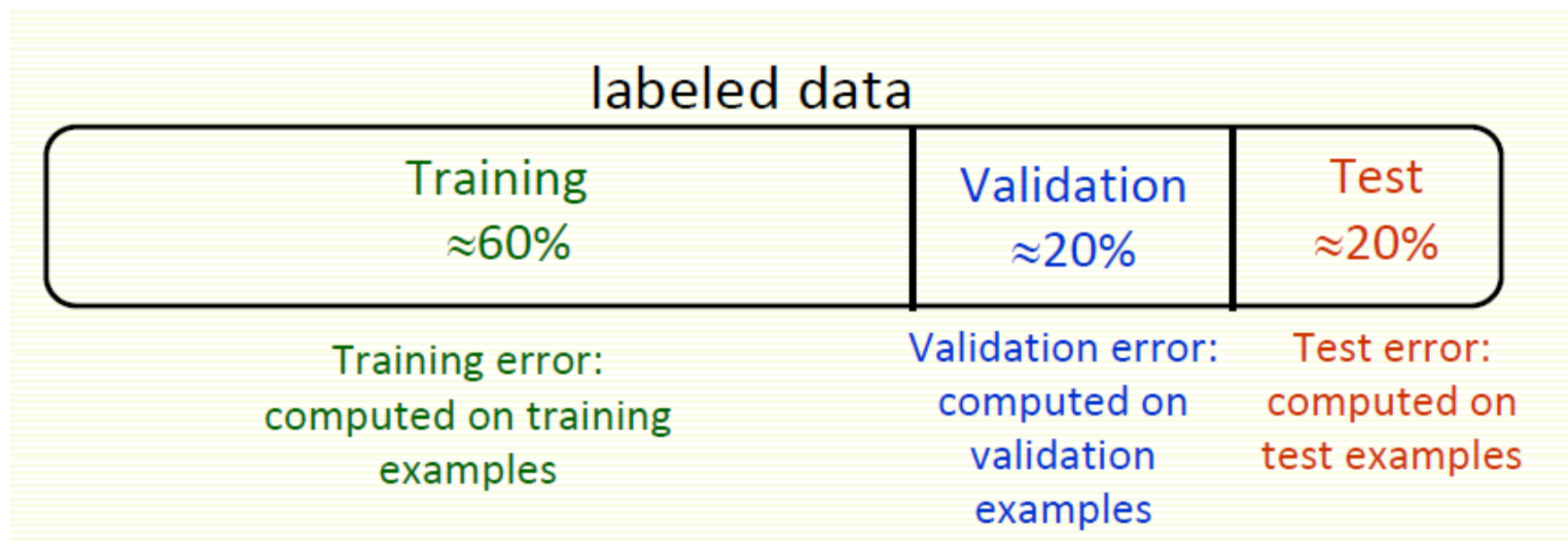
# VALIDATION DATA

Same question when choosing among several classifiers

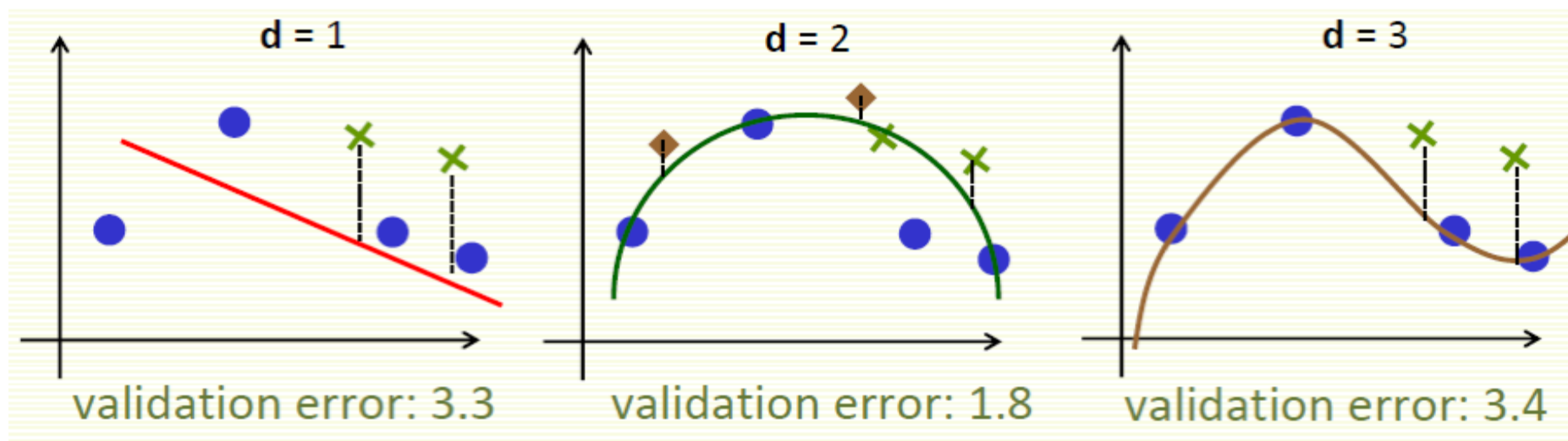
- our polynomial degree example can be looked at as choosing among 3 classifiers (degree 1, 2, or 3)
- Solution: split the labeled data into three parts



# TRAINING/ VALIDATION



# Training/Validation/Test Data



- Training Data

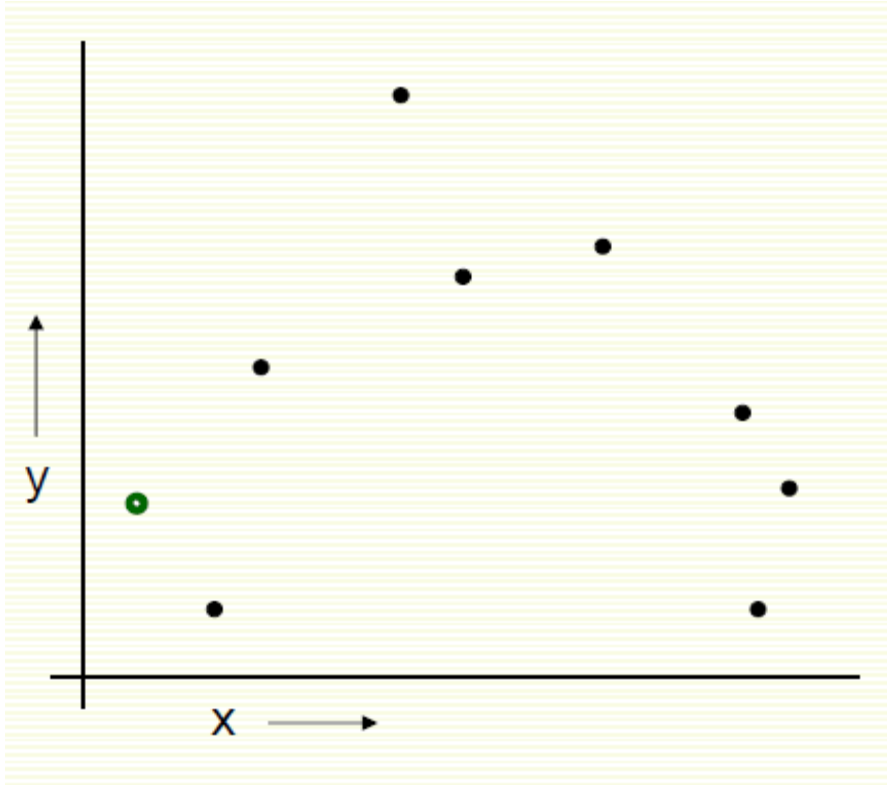
- Validation Data

$d = 2$  is chosen

- Test Data

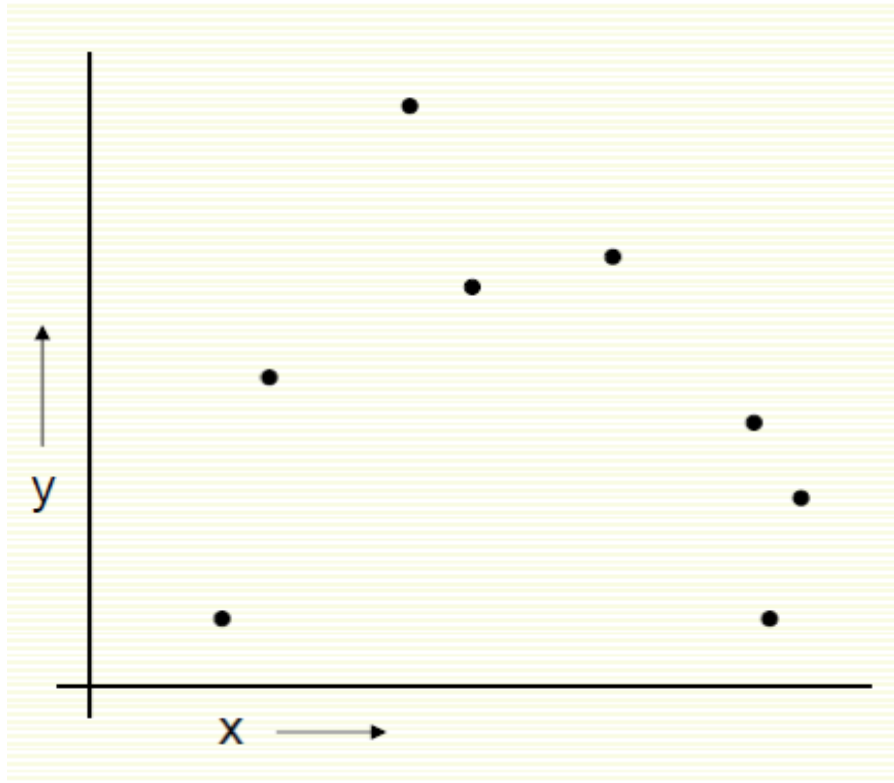
1.3 test error computed for  $d = 2$

# LOOCV (Leave-one-out Cross Validation)



- For  $k=1$  to  $R$ 
  1. Let  $(\mathbf{x}^k, \mathbf{y}^k)$  be the  $k$  example

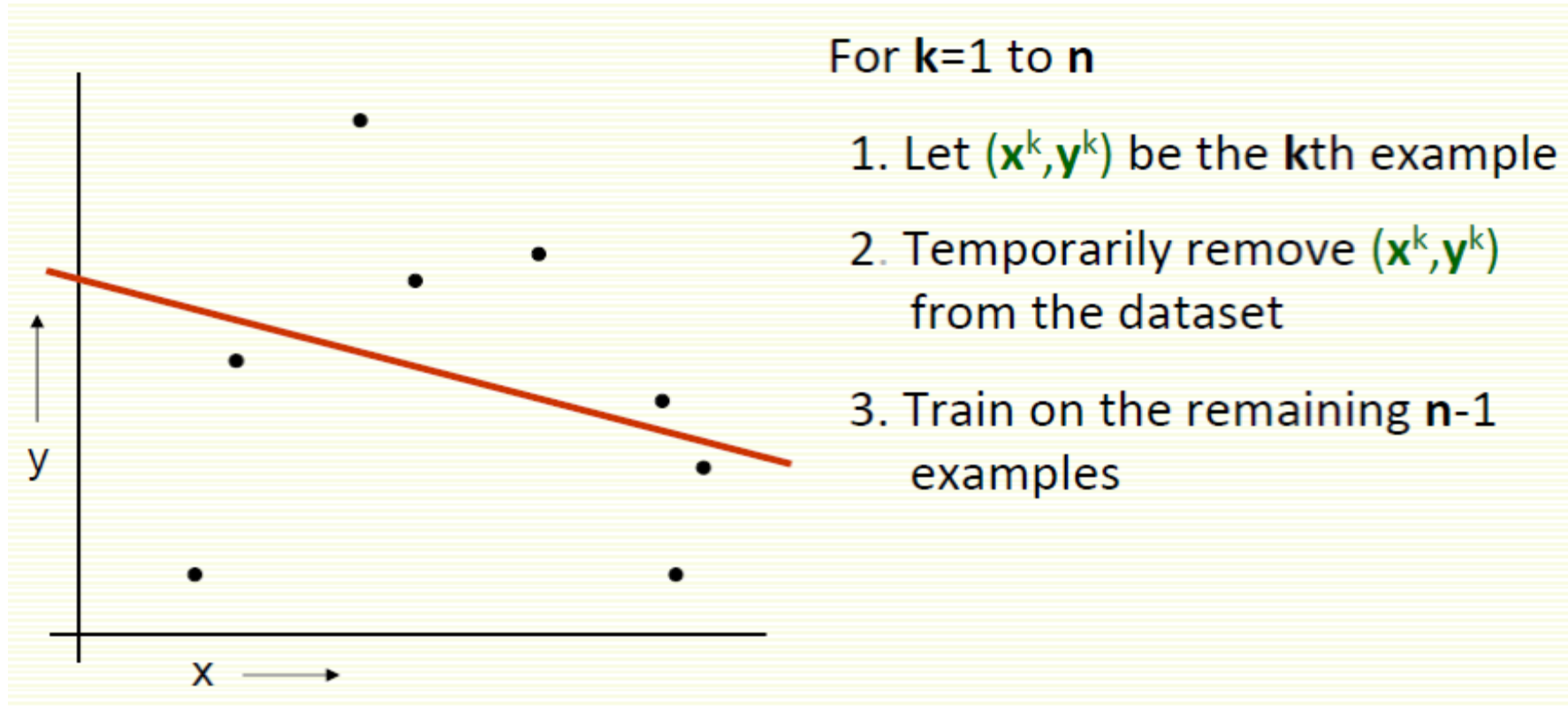
# LOOCV (Leave-one-out Cross Validation)



For  $k=1$  to  $n$

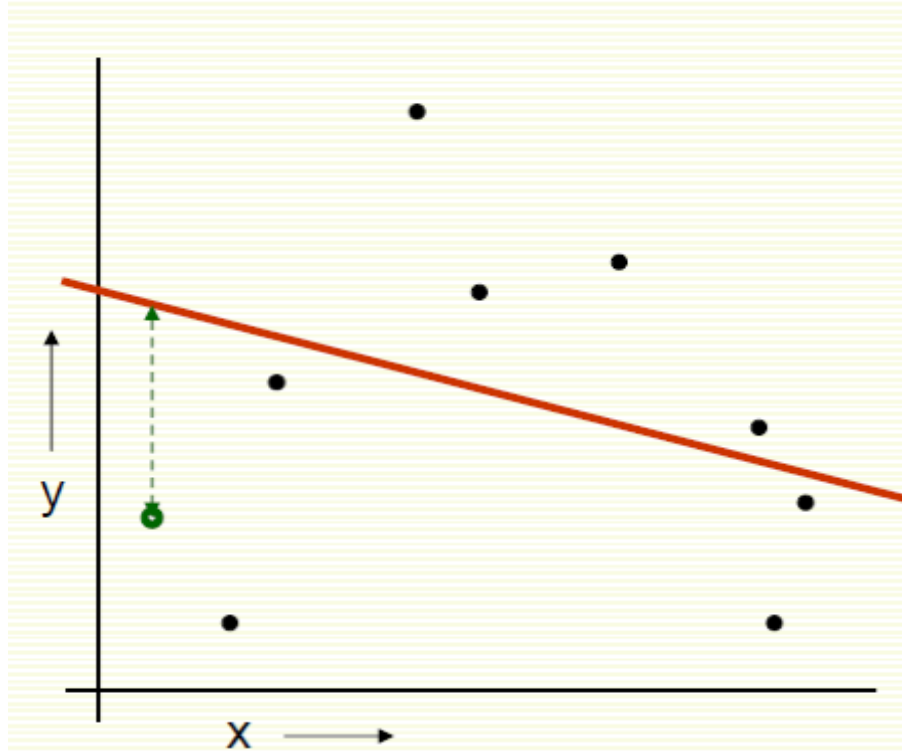
1. Let  $(\mathbf{x}^k, \mathbf{y}^k)$  be the  $k$ th example
2. Temporarily remove  $(\mathbf{x}^k, \mathbf{y}^k)$  from the dataset

# LOOCV (Leave-one-out Cross Validation)





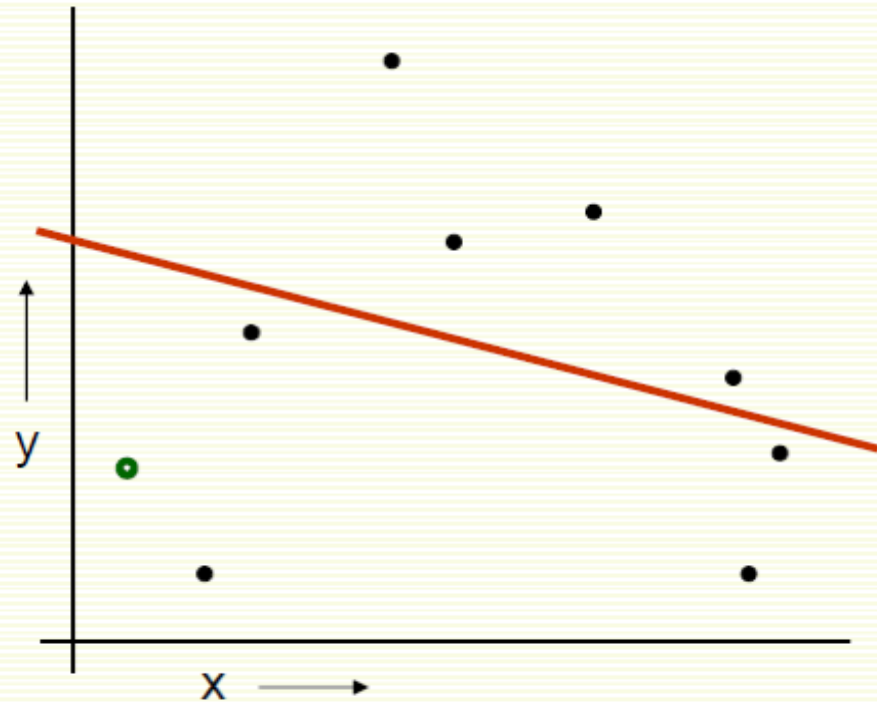
# LOOCV (Leave-one-out Cross Validation)



For  $k=1$  to  $n$

1. Let  $(x^k, y^k)$  be the  $k$ th example
2. Temporarily remove  $(x^k, y^k)$  from the dataset
3. Train on the remaining  $n-1$  examples
4. Note your error on  $(x^k, y^k)$

# LOOCV (Leave-one-out Cross Validation)

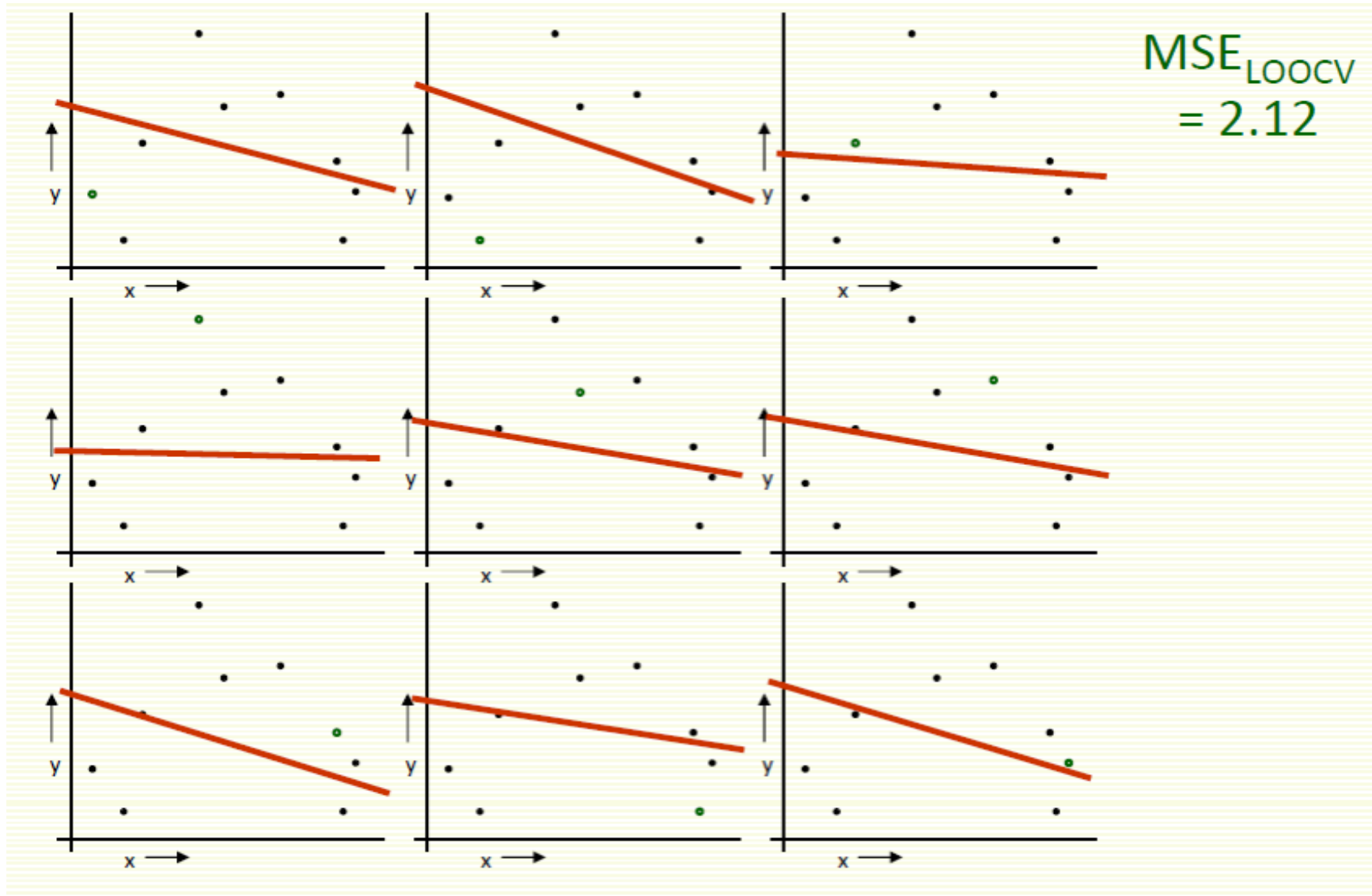


For  $k=1$  to  $n$

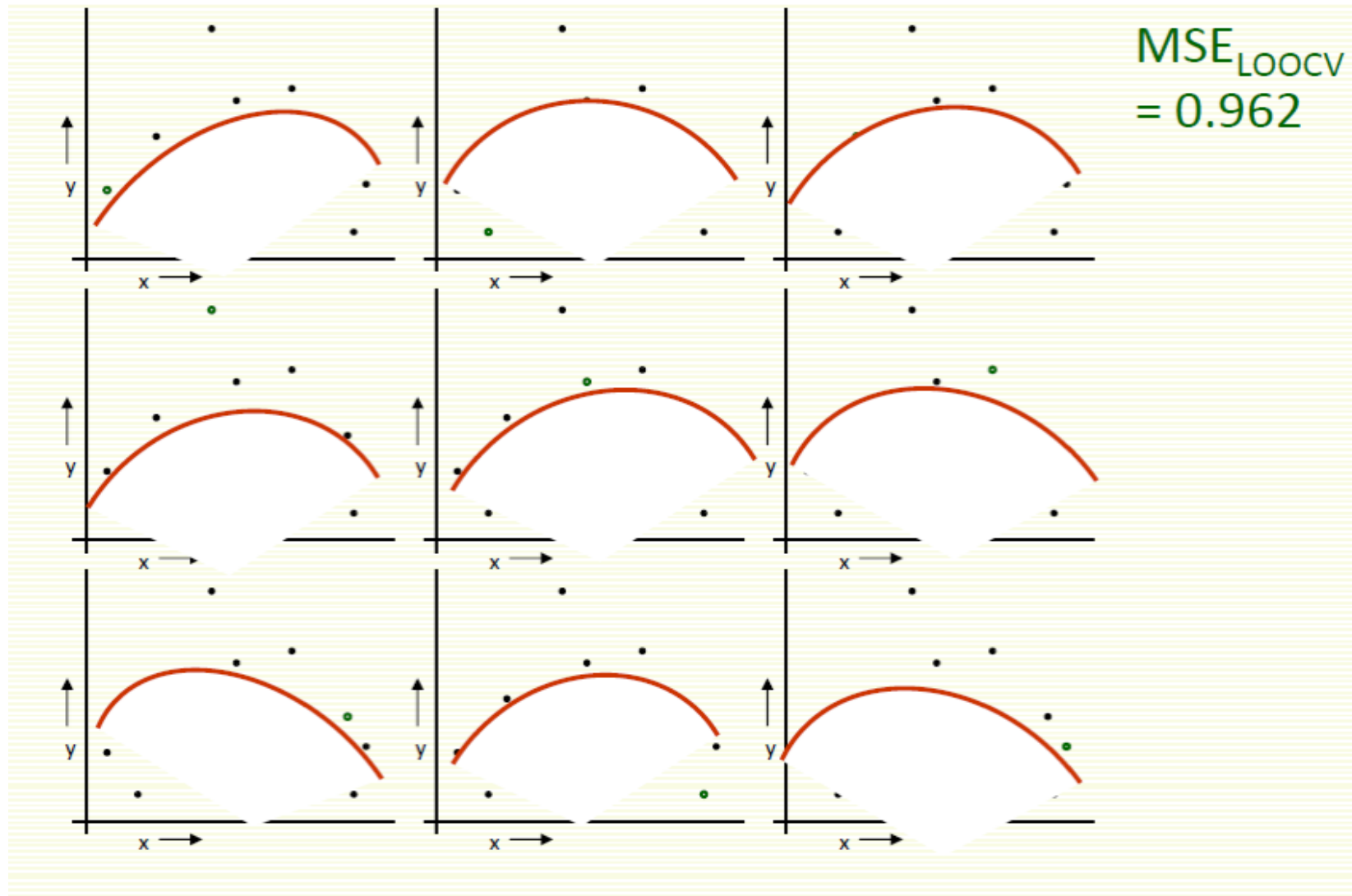
1. Let  $(\mathbf{x}^k, \mathbf{y}^k)$  be the  $k$ th example
2. Temporarily remove  $(\mathbf{x}^k, \mathbf{y}^k)$  from the dataset
3. Train on the remaining  $n-1$  examples
4. Note your error on  $(\mathbf{x}^k, \mathbf{y}^k)$

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

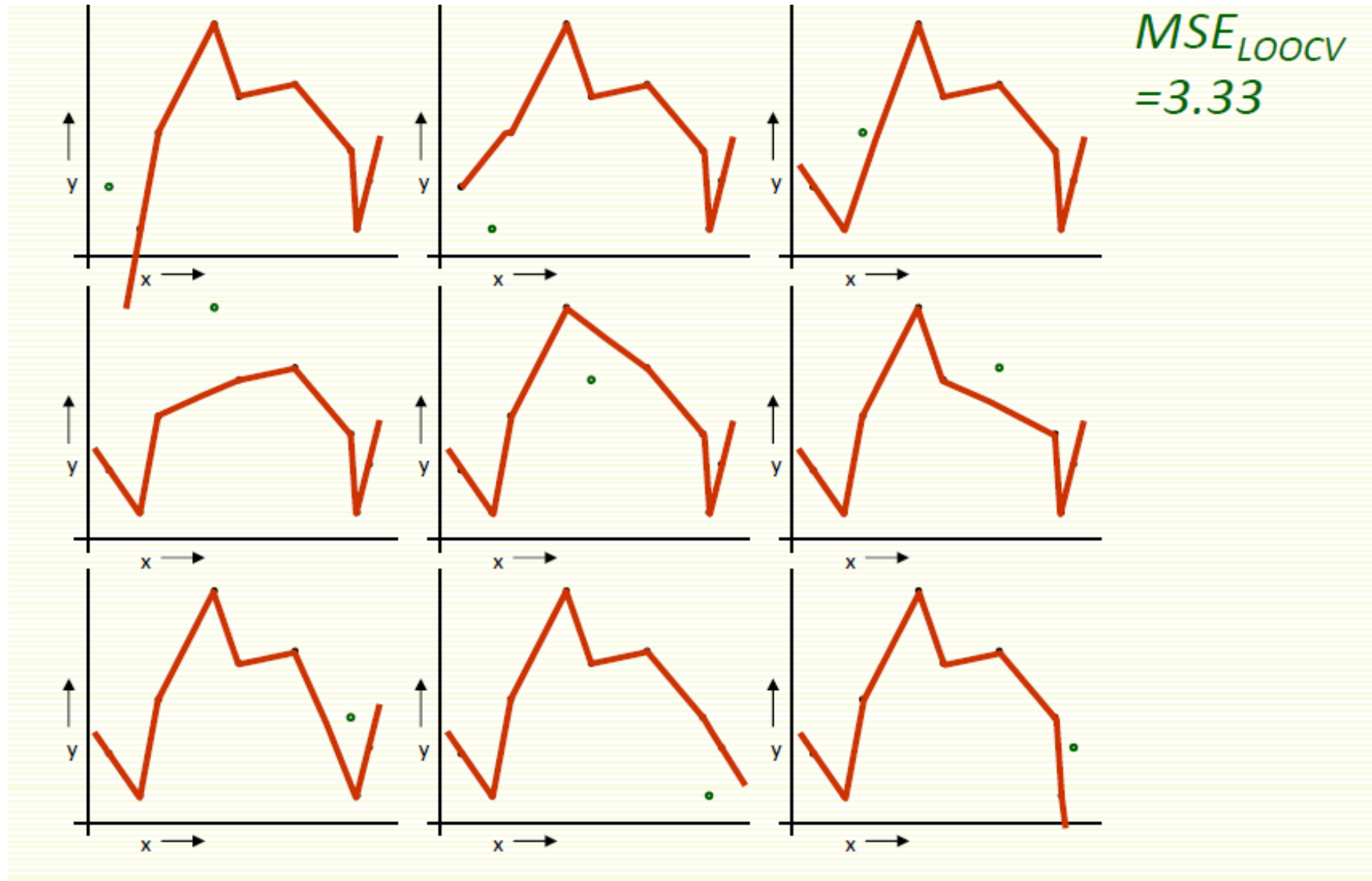
# LOOCV (Leave-one-out Cross Validation)



# LOOCV for Quadratic Regression



# LOOCV for Join The Dots



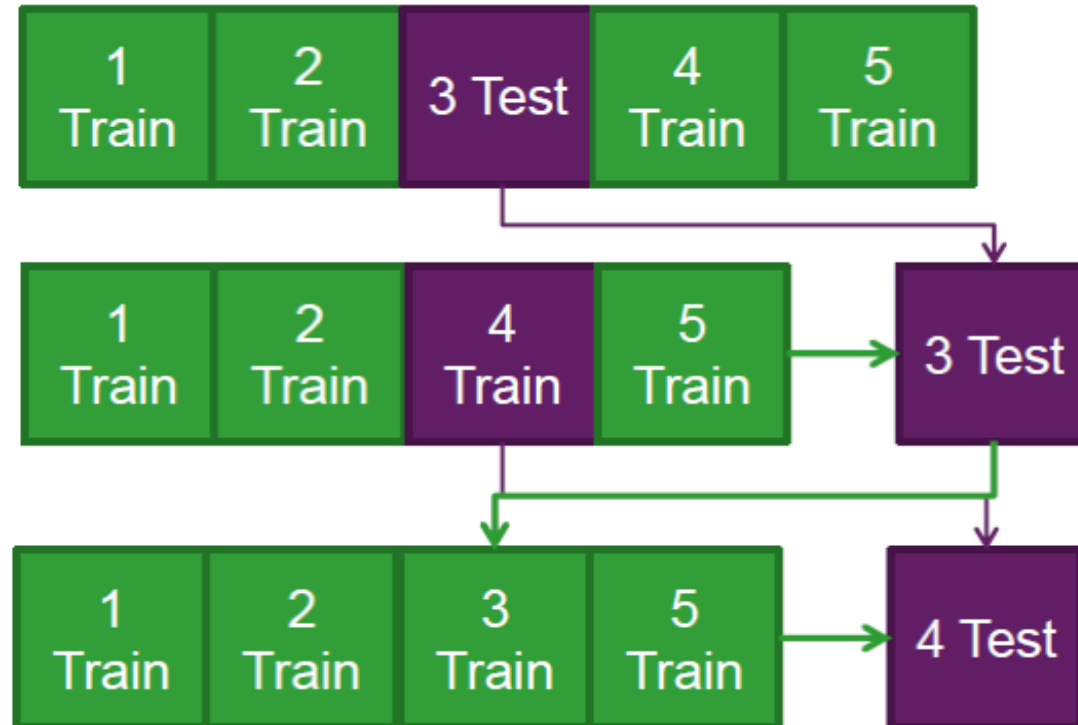
# Which kind of Cross Validation?

	Downside	Upside
Test-set	may give unreliable estimate of future performance	cheap
Leave-one-out	expensive	doesn't waste data

# K-Fold Cross Validation

1. Fix model parameters to some value (for example:  $p$  in the polynomial)
2. Split Training dataset (TRAIN) into  $K$  chunks.
3. For  $k=1,2,\dots,K$ :
  - (a) Set  $\text{TRAIN}_{\text{validation}}$  to be the  $k$ th chunk of data, and  $\text{TRAIN}_{\text{fit}}$  to be the other  $K - 1$  chunks.
  - (b) Fit each model to  $\text{TRAIN}_{\text{fit}}$  and evaluate how well it does on  $\text{TRAIN}_{\text{validation}}$ .
4. Pick the model that has the best average test score.
5. Retrain that model on all of TRAIN, and output that.

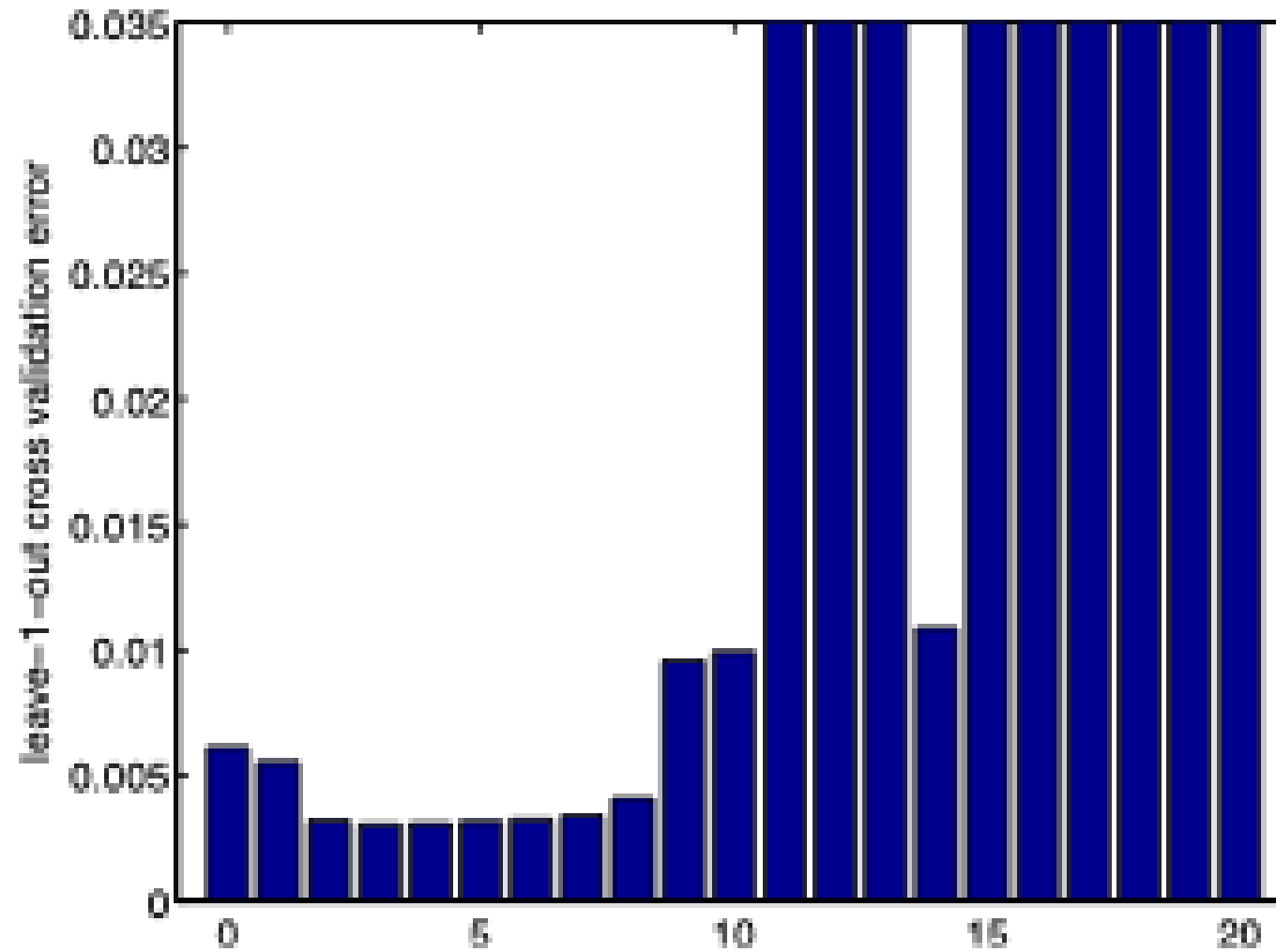
## K-fold Cross Validation Example



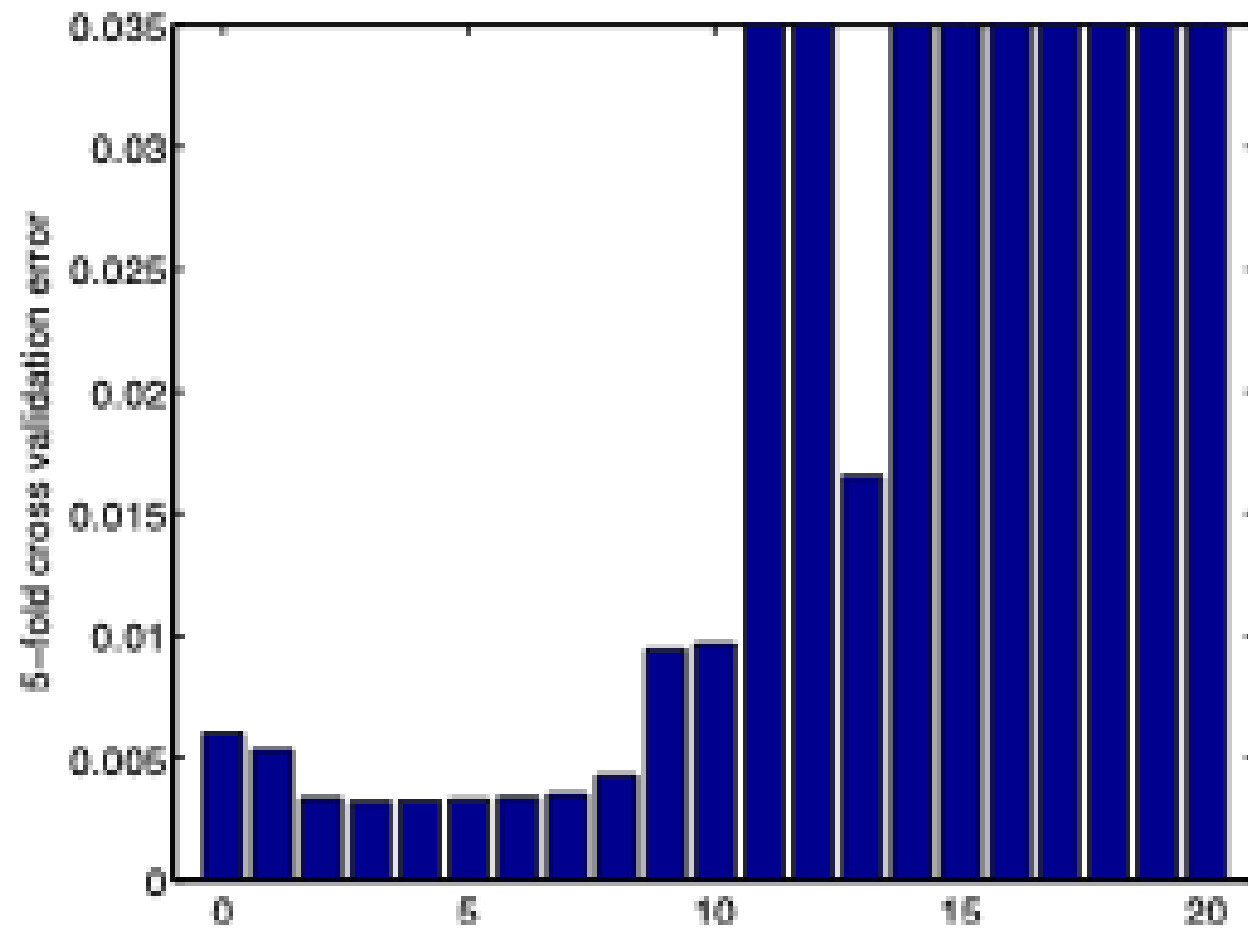
1. Split the data into 5 samples
2. Fit a model to the training samples and use the test sample to calculate a CV metric.
3. Repeat the process for the next sample, until all samples have been used to either train or test the model



## Example1: Leave One Out Error



## Example 1: Revisit (Results with 5-fold Cross Validation)



# Which kind of Cross Validation?

	Downside	Upside
Test-set	may give 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 $n$ times
3-fold	wastes more data than 10-fold, more expensive than test set	slightly better than test-set
N-fold	Identical to Leave-one-out	

# Improve cross-validation

- Even better: *repeated cross-validation*

Example:

10-fold cross-validation is repeated 10 times and results are averaged (reduce the variance)