

# CROSS-VALIDATION AND MODEL SELECTION

# Validation

- Validation addresses the problem of **over-fitting**.
- Internal Validation: Validate your model on your current data set (cross-validation)
- External Validation: Validate your model on a completely new dataset

# How to check if a model fit is good?

- The  $R^2$  statistic has become the almost universally standard measure for model fit in linear models.
- What is  $R^2$ ?

$$R^2 = 1 - \frac{\sum(y_i - f_i)^2}{\sum(y_i - \bar{y})^2}$$

← Model error  
← Variance in the dependent variable

- It is the ratio of error in a model over the total variance in the dependent variable.
- Hence the lower the error, the higher the  $R^2$  value.

# How to check if a model fit is good?

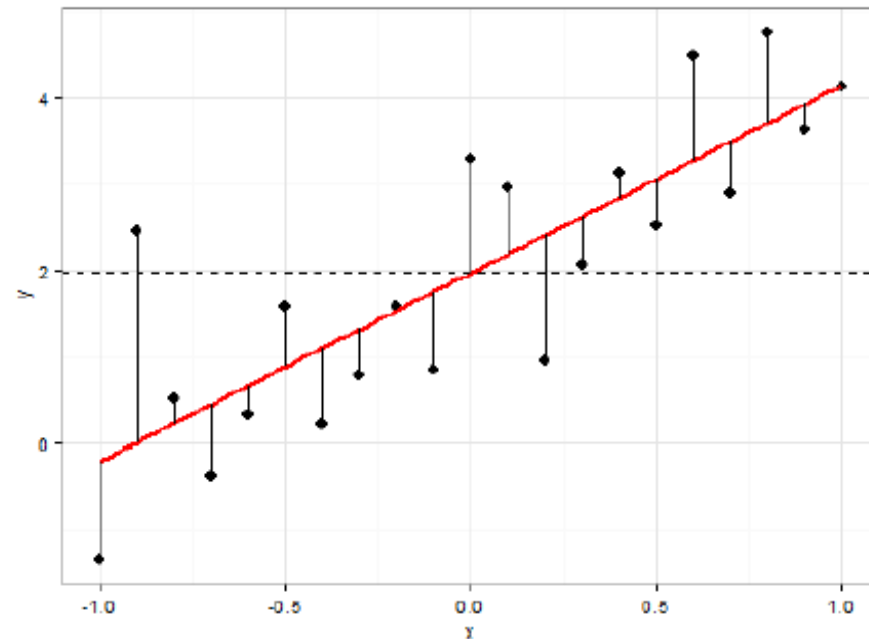
$$\sum (y_i - f_i)^2 = 18.568$$

$$\sum (y_i - \bar{y})^2 = 55.001$$

$$R^2 = 1 - \frac{18.568}{55.001}$$

$$R^2 = 0.6624$$

A decent model fit!



# How to check if a model fit is good?

$$\sum (y_i - f_i)^2 = 15.276$$

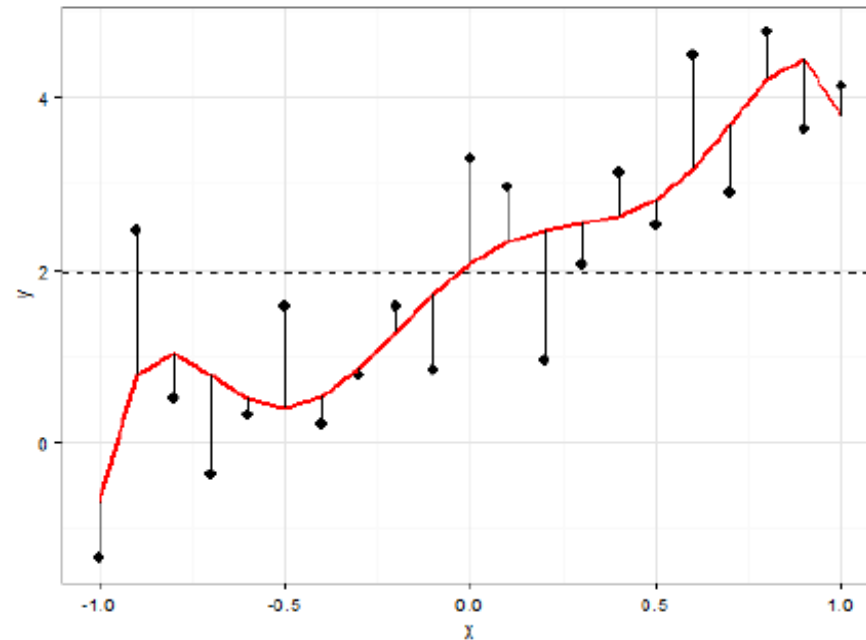
$$\sum (y_i - \bar{y})^2 = 55.001$$

$$R^2 = 1 - \frac{15.276}{55.001}$$

$$R^2 = 0.72$$

Is this a better model?

No, **overfitting!**



# OVERFITTING

- Modeling techniques tend to overfit the data.
- Multiple regression:
  - ✓ *Every* time you add a variable to the regression, the model's  $R^2$  goes up.
  - ✓ Naïve interpretation: *every* additional predictive variable helps to explain yet more of the target's variance. But that can't be true!
  - ✓ Left to its own devices, Multiple Regression will fit *too many* patterns.
  - ✓ A reason why modeling requires subject-matter expertise.

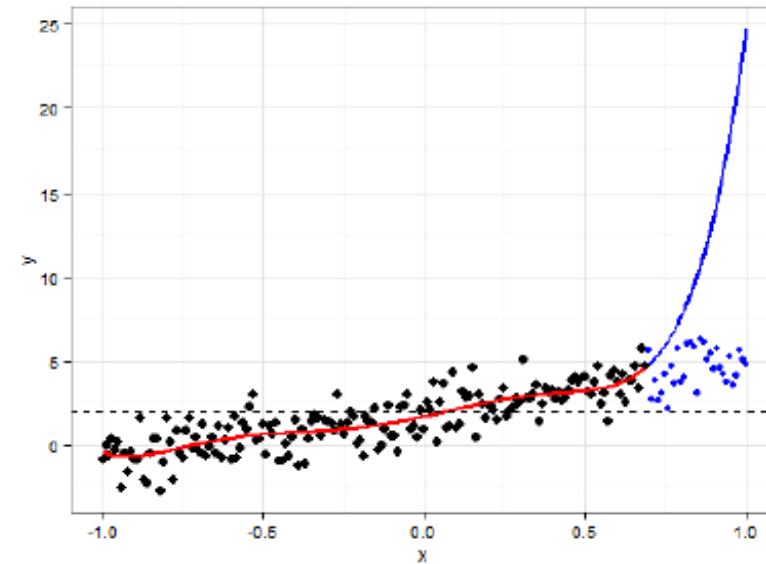
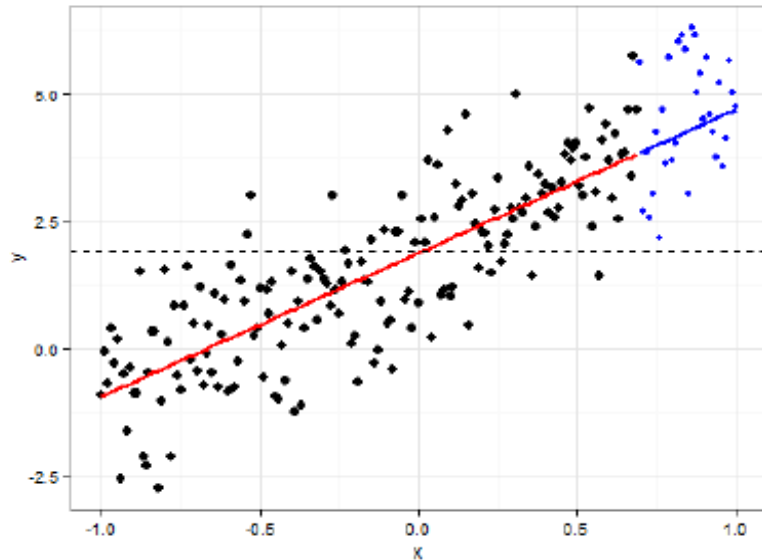
# OVERFITTING

- Error on the dataset used to *fit* the model can be misleading
  - › Doesn't predict future performance.
- Too much complexity can diminish model's accuracy on future data.
  - › Sometimes called the Bias-Variance Tradeoff.



# OVERFITTING

- What are the consequences of overfitting?
  - › “Overfitted models will have high  $R^2$  values, but will perform poorly in predicting out-of-sample cases”





# WHY WE NEED CROSS-VALIDATION?

- $R^2$ , also known as coefficient of determination, is a popular measure of quality of fit in regression. However, it does not offer any significant insights into how well our regression model can predict future values.
- An alternative, more practical procedure is *cross-validation*.

# When is cross-validation used?

- Very important in microarray experiments (“ $p$  is very larger than  $N$ ”).  
 $N$ =number of cases sampled for the study and  $p$ =true proportion exposed in all cases in the larger population.
- Anytime you want to prove that your model is not over-fit, that it will have good prediction in new datasets.

# CROSS-VALIDATION

- In cross-validation the original sample is split into two parts. One part is called the training (or *derivation*) sample, and the other part is called the *validation (or validation + testing)* sample.

## **1) What portion of the sample should be in each part?**

If sample size is very large, it is often best to split the sample in half. For smaller samples, it is more conventional to split the sample such that  $\frac{2}{3}$  of the observations are in the derivation sample and  $\frac{1}{3}$  are in the validation sample.

# CROSS-VALIDATION

## **2) How should the sample be split?**

The most common approach is to divide the sample randomly, thus theoretically eliminating any systematic differences. One alternative is to define matched pairs of subjects in the original sample and to assign one member of each pair to the derivation sample and the other to the validation sample.

- Modeling of the data uses one part only. The model selected for this part is then used to predict the values in the other part of the data. A valid model should show good predictive accuracy.
- One thing that R-squared offers no protection against is overfitting. On the other hand, cross validation, by allowing us to have cases in our testing set that are different from the cases in our training set, inherently offers protection against overfitting.

# Holdout validation

- One way to validate your model is to fit your model on half your dataset (your “training set”) and test it on the remaining half of your dataset (your “test set”).
- If over-fitting is present, the model will perform well in your training dataset but poorly in your test dataset.
- Of course, you “waste” half your data this way, and often you don’t have enough data to spare...

# Alternative strategies:

- Leave-one-out validation (leave one observation out at a time; fit the model on the remaining training data; test on the held out data point).
- **K-fold cross-validation**

# 10-fold cross-validation (one example of K-fold cross-validation)

- 1. Randomly divide your data into 10 pieces, 1 through k.
- 2. Treat the 1<sup>st</sup> tenth of the data as the test dataset. Fit the model to the other nine-tenths of the data (which are now the training data).
- 3. Apply the model to the test data (e.g., for logistic regression, calculate predicted probabilities of the test observations).
- 4. Repeat this procedure for all 10 tenths of the data.
- 5. Calculate statistics of model accuracy and fit (e.g., ROC curves) from the test data only.

# Robust Model

- If a model has no predictive power, you have a 50-50 chance of correctly classification.
- For example a model with 79% chance of correct classification can be considered quite an improvement over 50%.
- AUC=Area Under Curve (ROC Curve) Since we don't have extra data lying around, we can use 10-fold cross-validation to get a better estimate of the AUC...
- Before cross validation AUC=0.79 and After cross-validation, the AUC was 0.78.
- This shows that the model is robust.



# 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

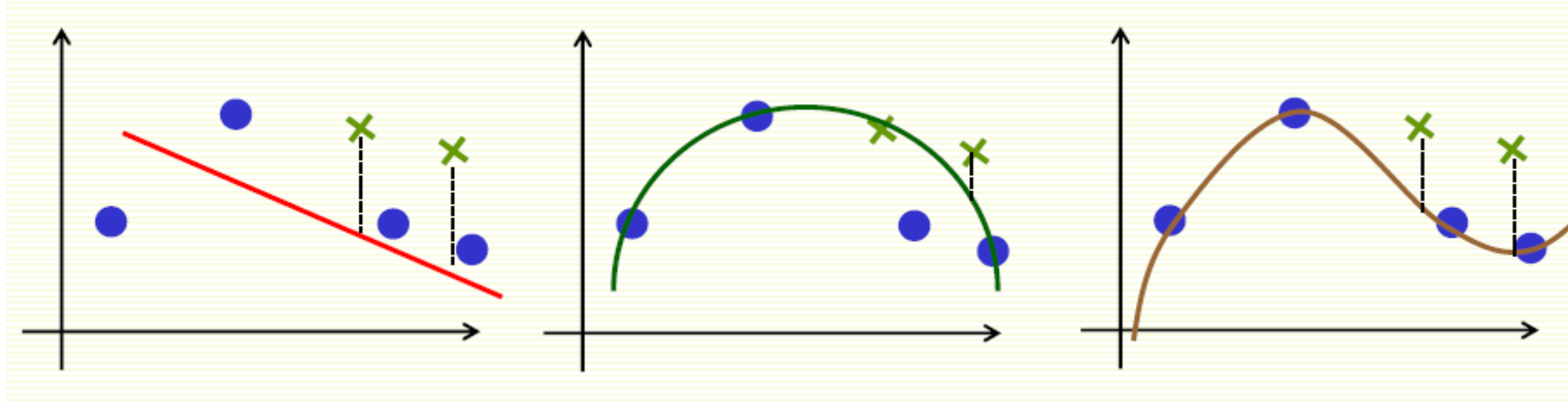
Talked about splitting data in training/test sets

- 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

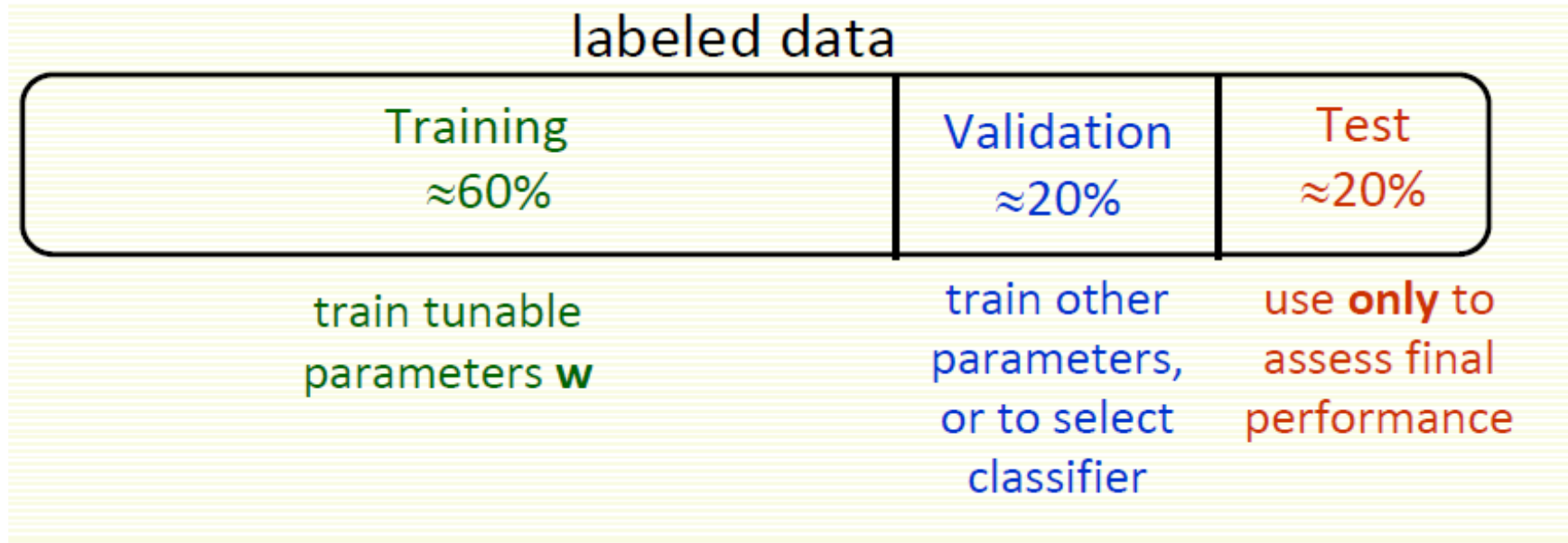
Except what do we report as the test error now?

- 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

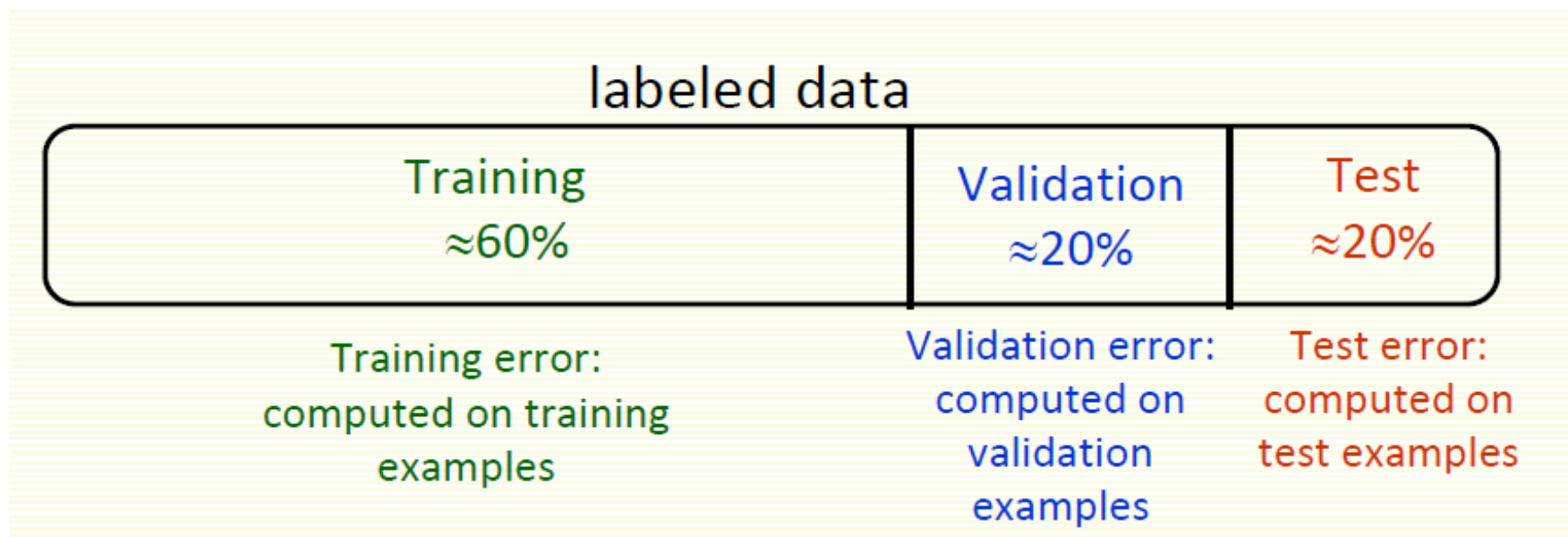
# VALIDATION DATA

Same question when choosing among several classifiers

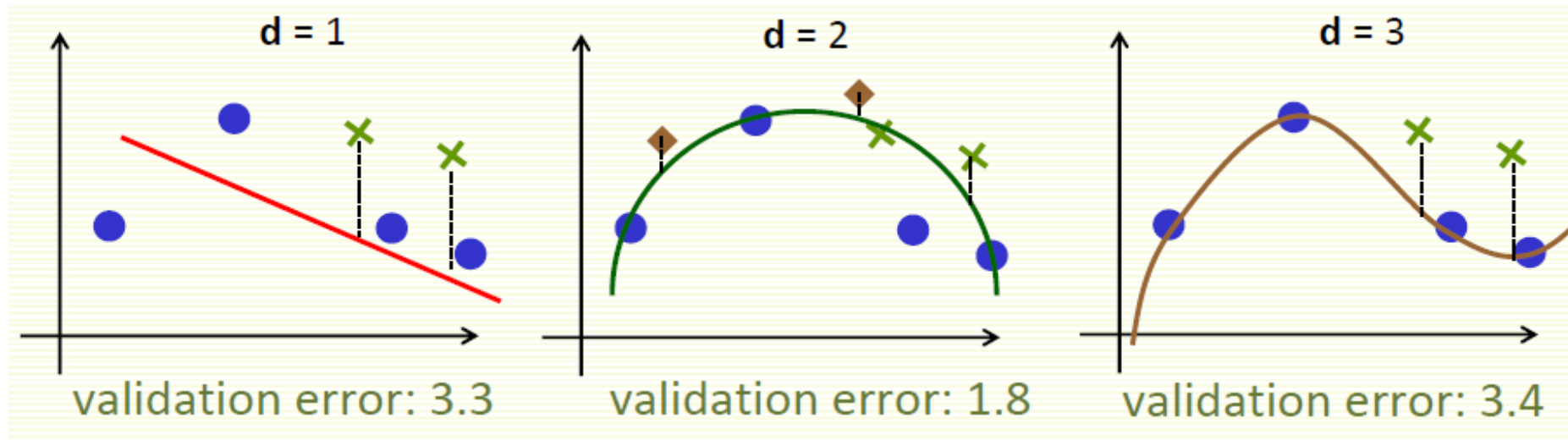
- our polynomial degree example can be looked at as choosing among 3 classifiers (degree 1, 2, or 3)
- Solution: split the labeled data into three parts



# TRAINING/ VALIDATION

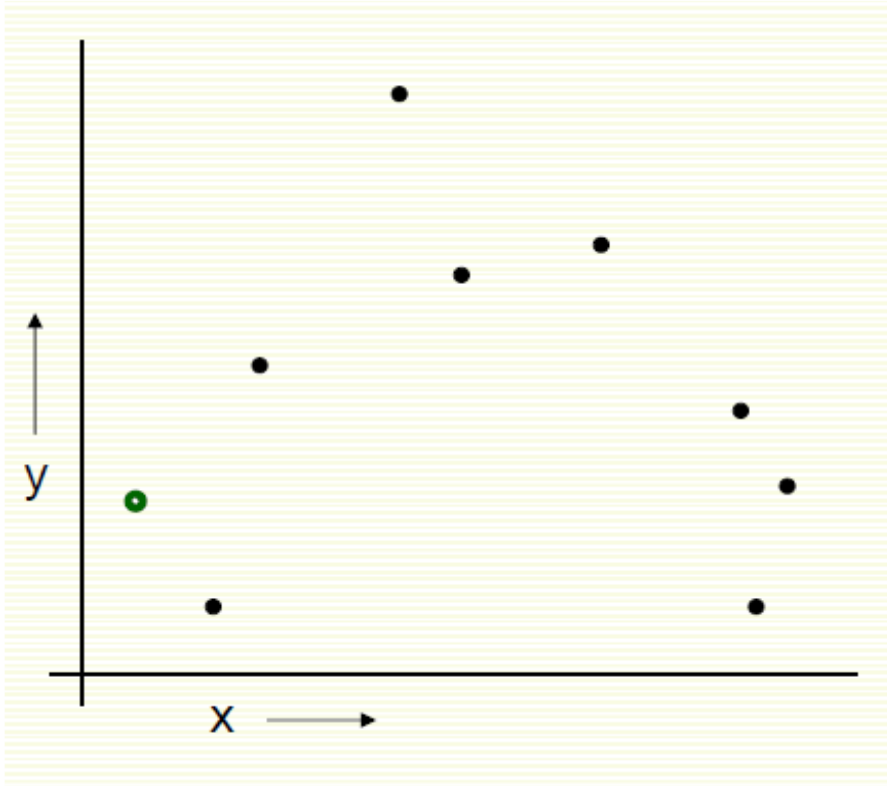


# Training/Validation/Test Data



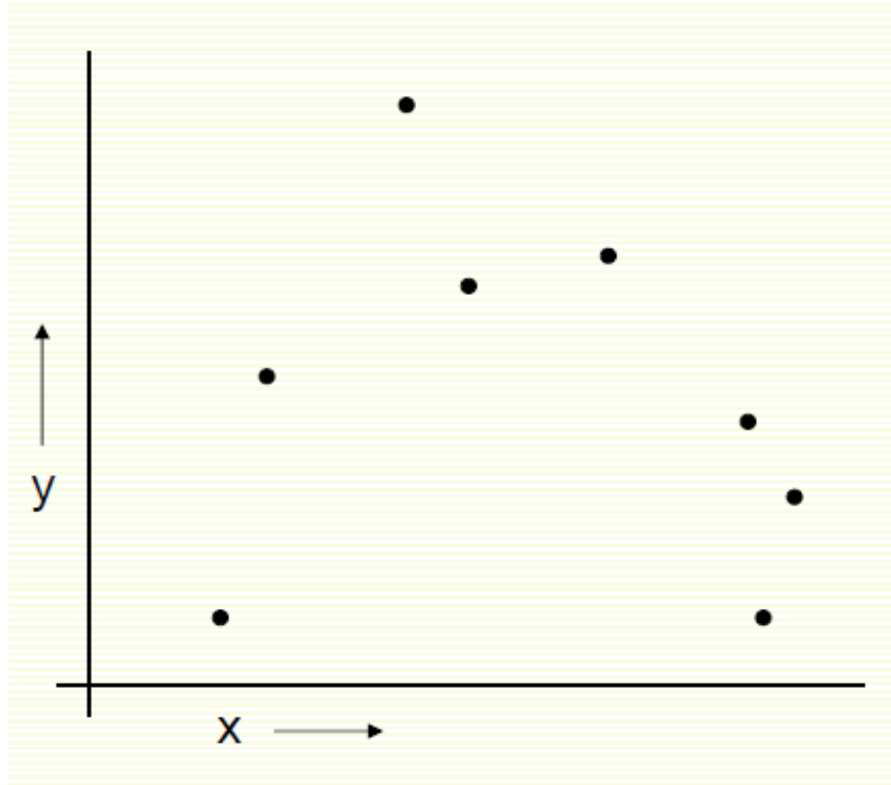
- Training Data
- Validation Data
  - $d = 2$  is chosen
- Test Data
  - 1.3 test error computed for  $d = 2$

# LOOCV (Leave-one-out Cross Validation)



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

# LOOCV (Leave-one-out Cross Validation)

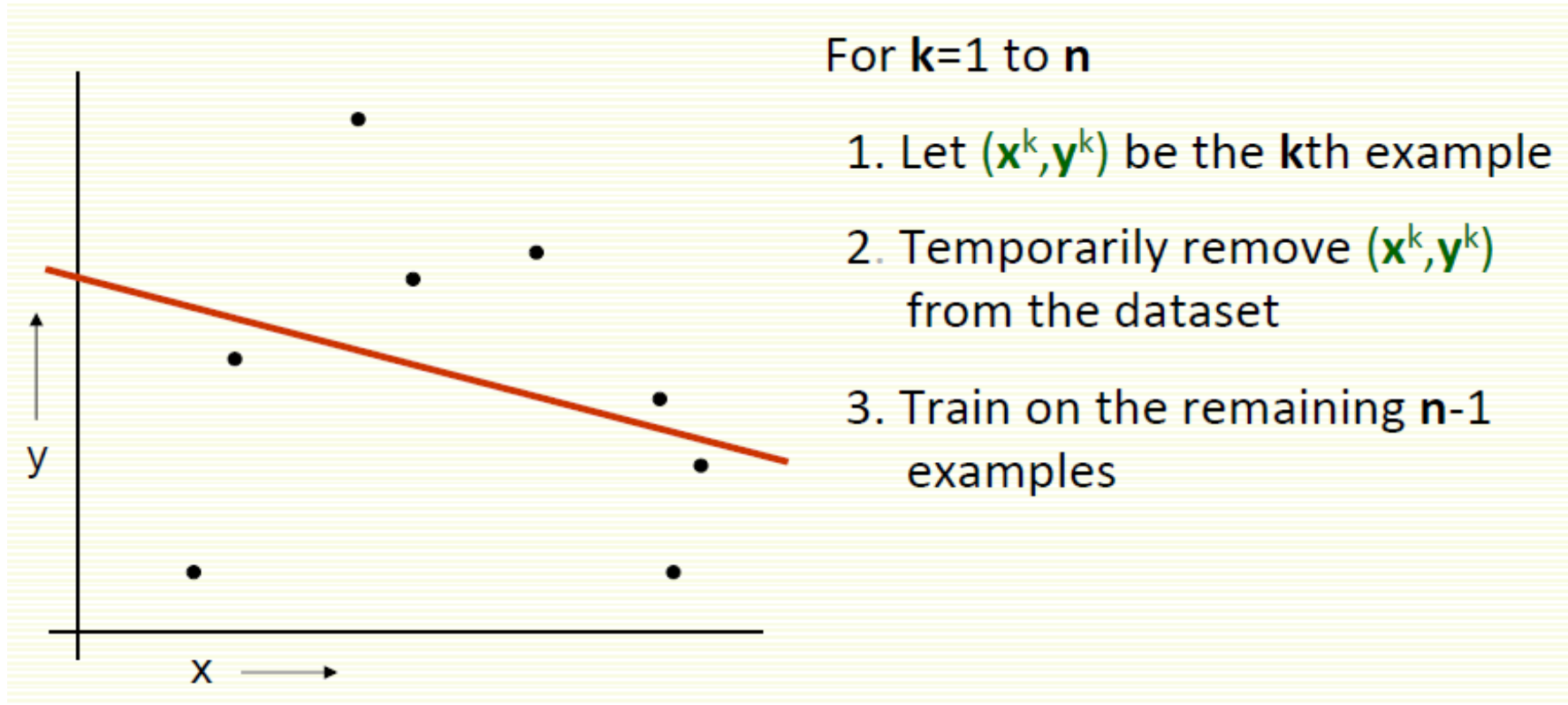


For  $k=1$  to  $n$

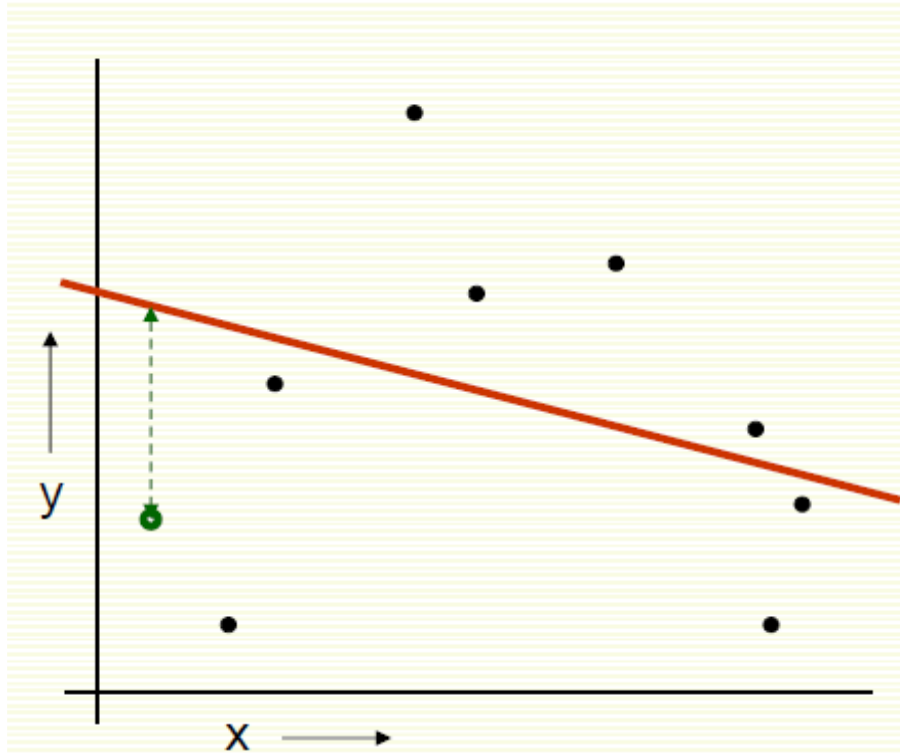
1. Let  $(\mathbf{x}^k, \mathbf{y}^k)$  be the  $k$ th example
2. Temporarily remove  $(\mathbf{x}^k, \mathbf{y}^k)$  from the dataset



# LOOCV (Leave-one-out Cross Validation)



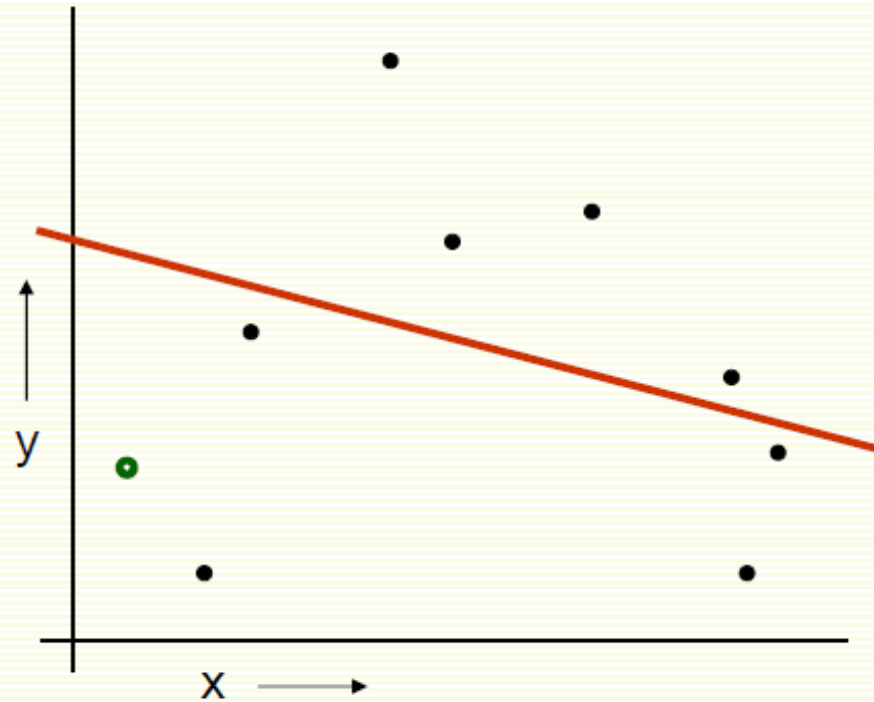
# LOOCV (Leave-one-out Cross Validation)



For  $k=1$  to  $n$

1. Let  $(\mathbf{x}^k, \mathbf{y}^k)$  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  $(\mathbf{x}^k, \mathbf{y}^k)$

# LOOCV (Leave-one-out Cross Validation)

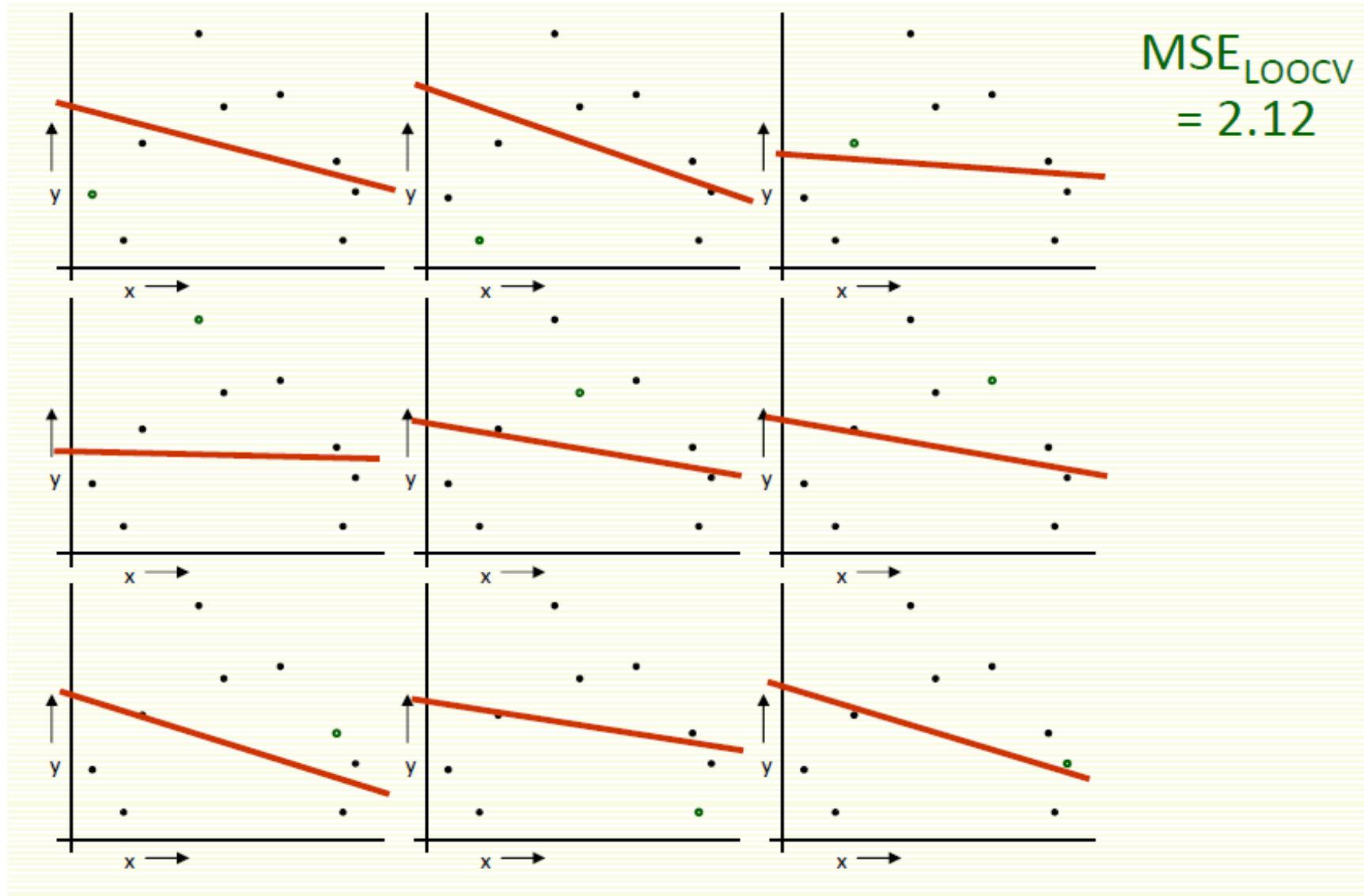


For  $k=1$  to  $n$

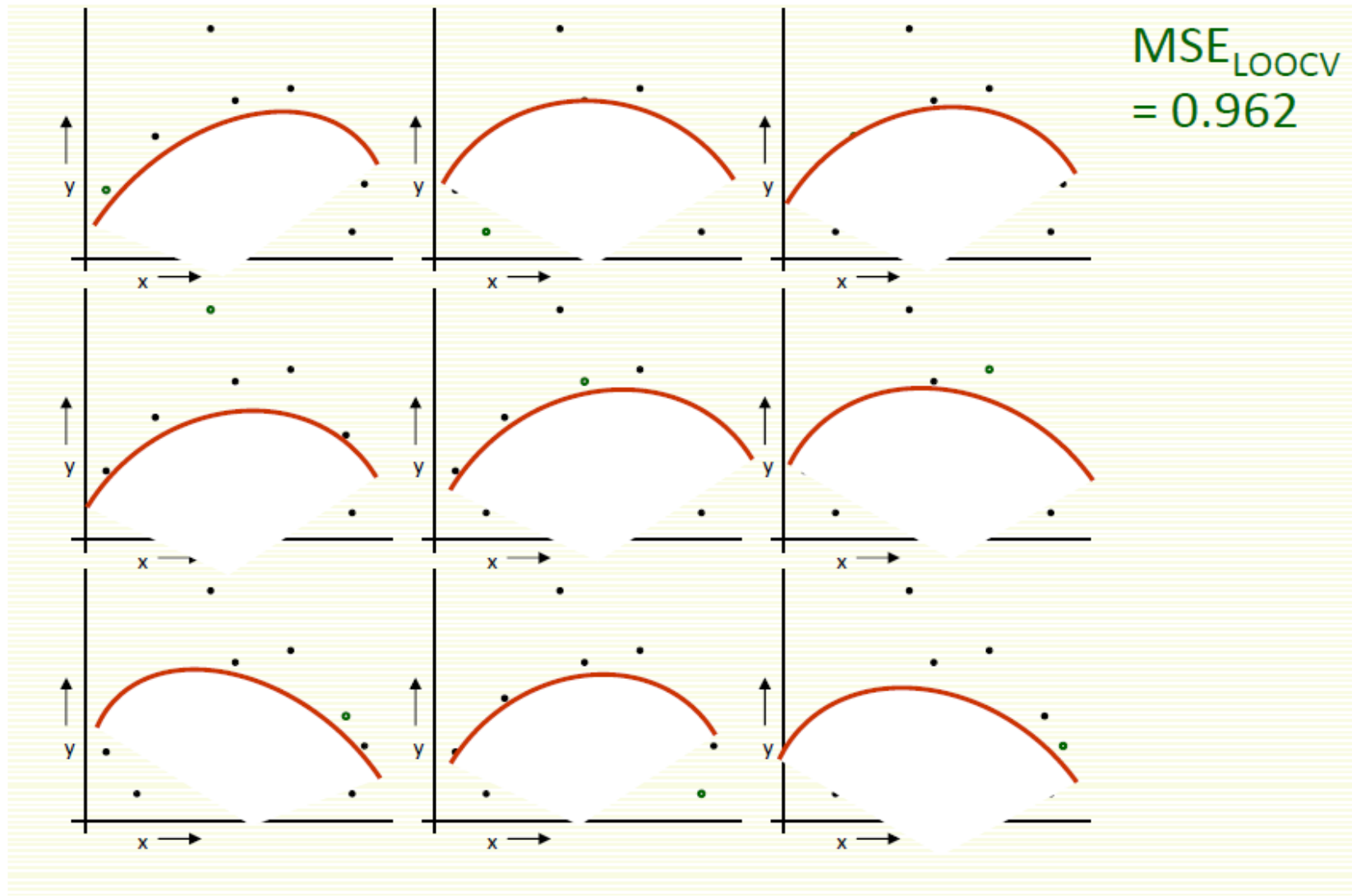
1. Let  $(\mathbf{x}^k, \mathbf{y}^k)$  be the  $k$ th example
2. Temporarily remove  $(\mathbf{x}^k, \mathbf{y}^k)$  from the dataset
3. Train on the remaining  $n-1$  examples
4. Note your error on  $(\mathbf{x}^k, \mathbf{y}^k)$

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

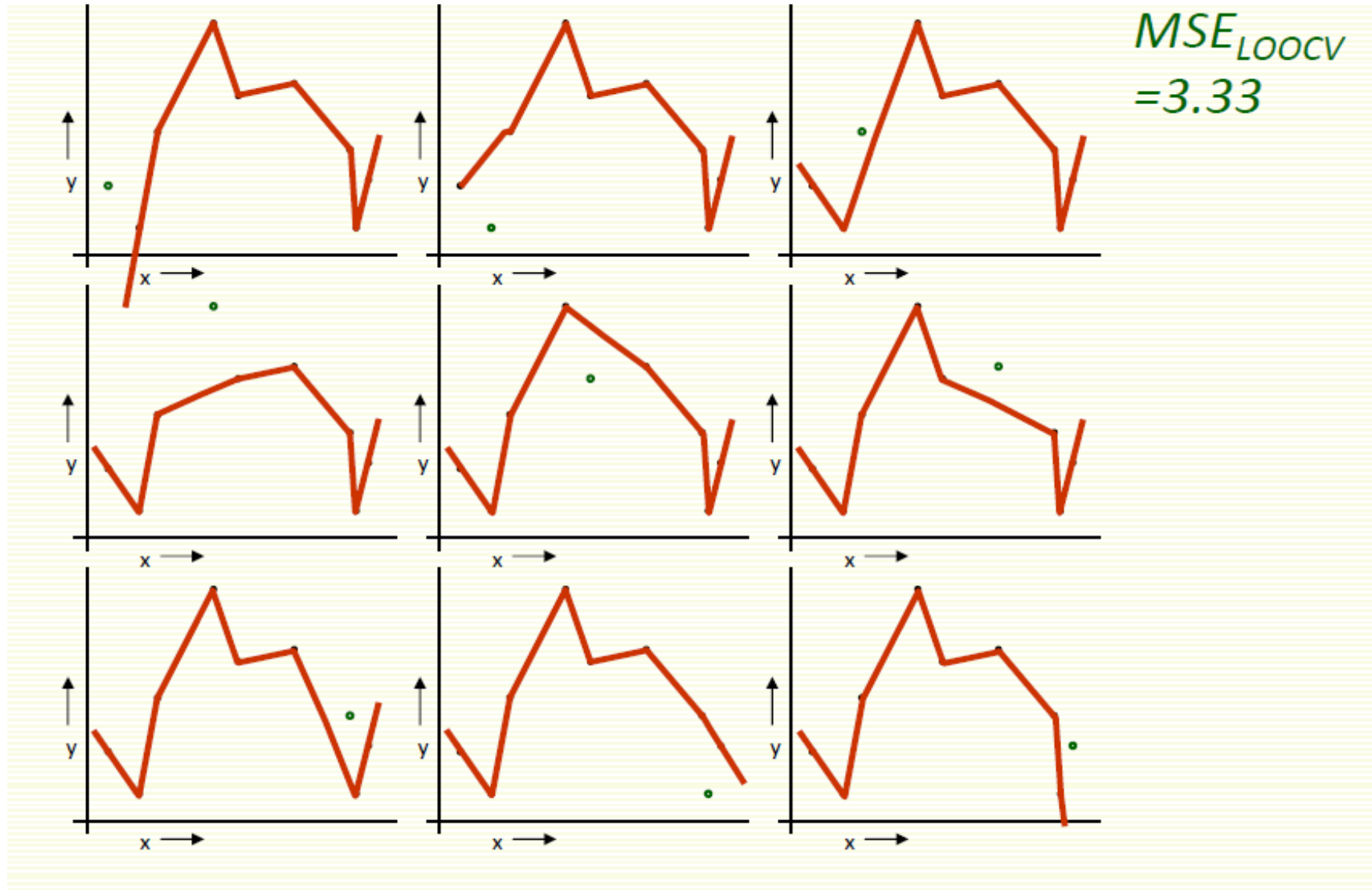
# LOOCV (Leave-one-out Cross Validation)



# LOOCV for Quadratic Regression



# LOOCV for Join The Dots



# Which kind of Cross Validation?

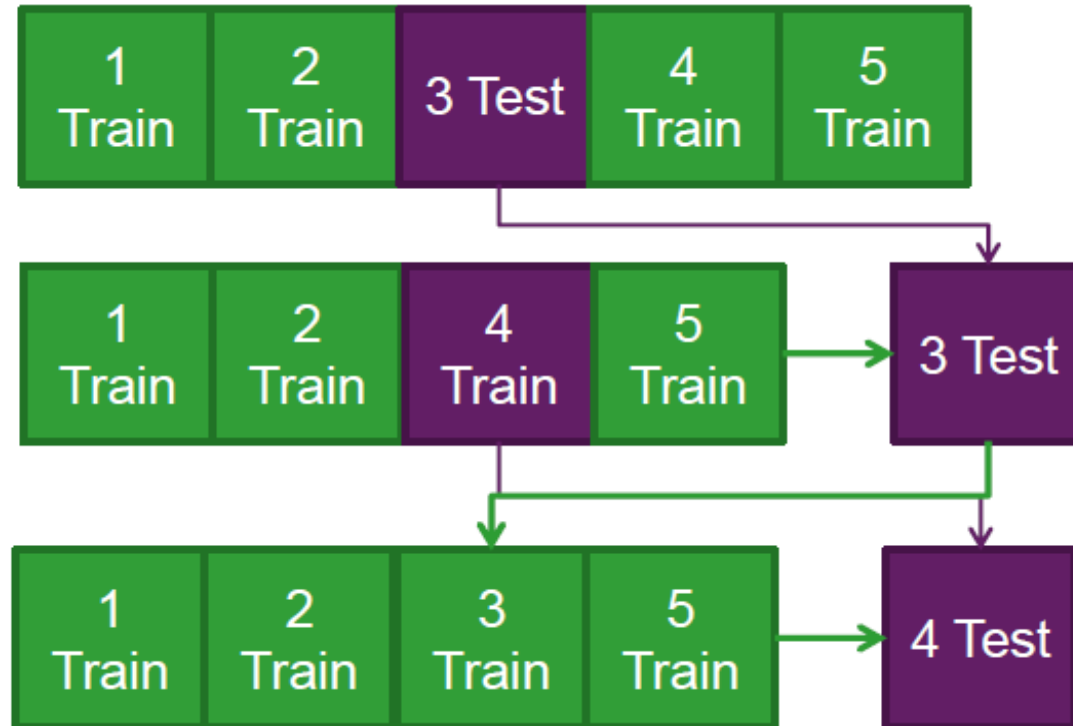
|               | Downside   | Upside             |
|---------------|--|--------------------|
| Test-set      | may give unreliable estimate of future performance | cheap              |
| Leave-one-out | expensive  | doesn't waste data |

# K-FOLD CROSS VALIDATION

- › Since data are often scarce, there might not be enough to set aside for a validation sample
- › To work around this issue k-fold CV works as follows:
  1. Split the sample into  $k$  subsets of equal size
  2. For each fold estimate a model on all the subsets except one
  3. Use the left out subset to test the model, by calculating a CV metric of choice
  4. Average the CV metric across subsets to get the CV error
- › This has the advantage of using all data for estimating the model, however finding a good value for  $k$  can be tricky



# 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

# 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)

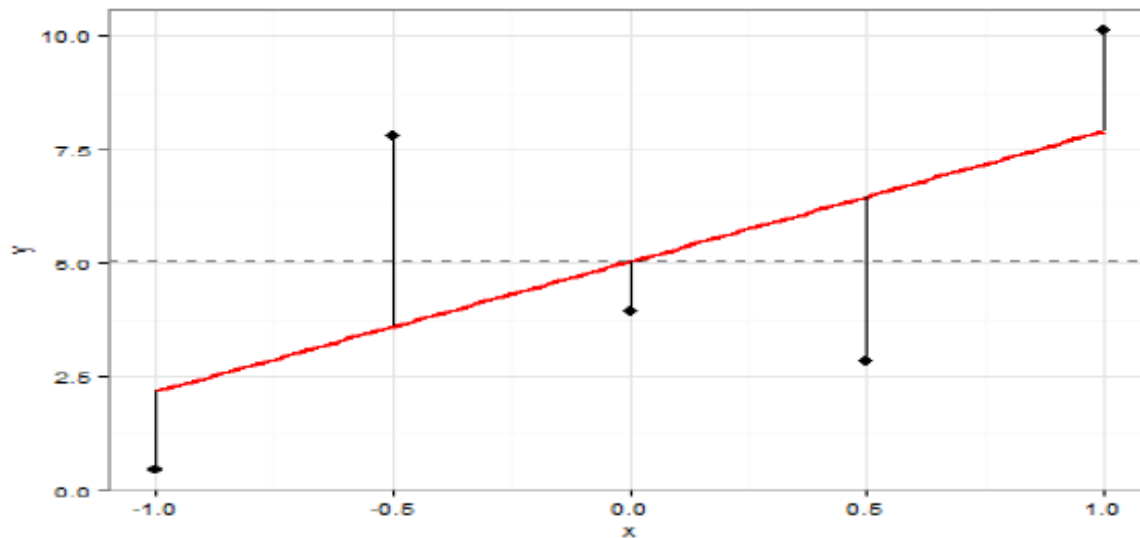
# Cross Validation - Metrics

- How do we determine if one model is predicting better than another model?

The basic relation:

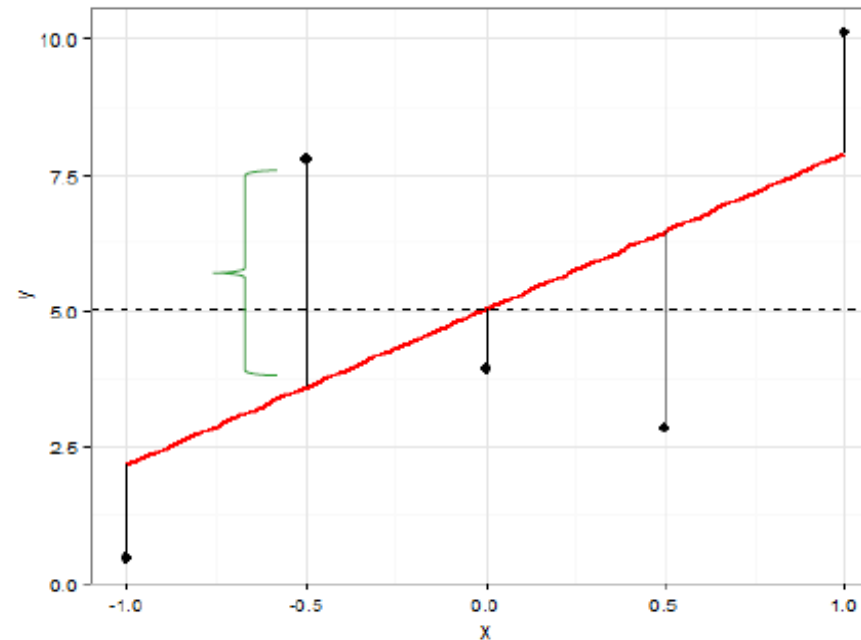
›  $Error_i = y_i - f_i$

← The difference between observed ( $y$ ) and predicted value ( $f$ ), when applying the model to unseen data



# Cross Validation Metrics

- › Mean Squared Error (MSE)
  - ›  $1/n \sum (y_i - f_i)^2$
  - › 7.96
- › Root Mean Squared Error (RMSE)
  - ›  $\sqrt{1/n \sum (y_i - f_i)^2}$
  - › 2.82
- › Mean Absolute Percentage Error (MAPE)
  - ›  $(1/n \sum |\frac{y_i - f_i}{y_i}|) * 100$
  - › 120%



# Best Practice for Reporting Model Fit

1. Use Cross Validation to find the best model
2. Report the RMSE and MAPE statistics from the cross validation procedure
3. Report the R Squared from the model as you normally would.

The added cross-validation information will allow one to evaluate not how much variance can be explained by the model, but also the predictive accuracy of the model. **Good models should have a high predictive AND explanatory power!**

# MEASURING THE MODEL ACCURACY

| Technique                      | Abbrev | Measures   |
|--------------------------------|--------|--|
| Mean Squared Error             | MSE    | The average of squared errors over the sample period   |
| Mean Error                     | ME     | The average dollar amount or percentage points by which forecasts differ from outcomes                 |
| Mean Percentage Error          | MPE    | The average of percentage errors by which forecasts differ from outcomes                               |
| Mean Absolute Error            | MAE    | The average of absolute dollar amount or percentage points by which a forecast differs from an outcome |
| Mean Absolute Percentage Error | MAPE   | The average of absolute percentage amount by which forecasts differ from outcomes                      |

# MEASURING THE MODEL ACCURACY

## 1. Mean Squared Error

The formula used to calculate the mean squared error is:

$$MSE = \frac{1}{n} \sum_{t=1}^n (a_t - f_t)^2$$

## 2. Mean Percentage Error

The formula used to calculate the mean percentage error is:

$$MPE = \frac{1}{n} \sum_{t=1}^n \frac{(a_t - f_t)}{a_t} \times 100$$

## 3. Mean Absolute Error

The formula used to calculate the mean absolute error is:

$$MAE = \frac{1}{n} \sum_{t=1}^n |(a_t - f_t)|$$



# MEASURING THE MODEL ACCURACY

## 4. Mean Absolute Percentage Error

The formula used to calculate the mean absolute percentage error is:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|(a_t - f_t)|}{a_t} \times 100$$