香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Introduction to Data Science

Lecture 23 Machine Learning: Model Selection

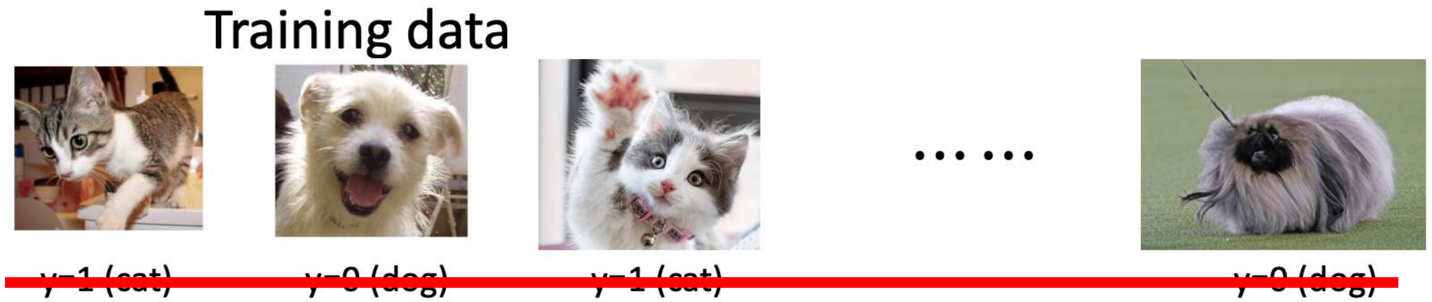
Zicheng Wang

Recap

Unsupervised Learning

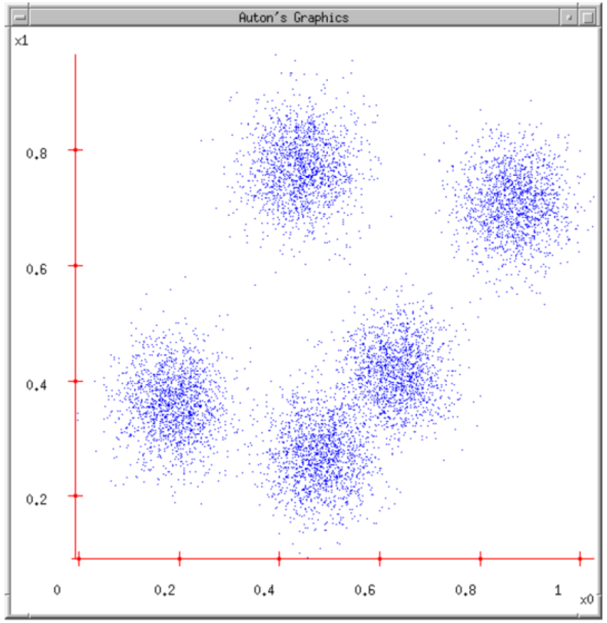
- Data lacks structured or objective answers, such as labels.

- In other words, for all samples (x^i, y^i), where $i = 1, \ldots N$, you can observe x^i but y^i remains unseen.



Clustering

- The algorithm figures out the grouping of objects based on the chosen **similarity/dissimilarity function**
  - **Points within a cluster are similar**
  - **Points across clusters are not so similar**



Dissimilarity/Similarity Function

- Desired properties of dissimilarity functions
  - Symmetry: $d(x, y) = d(y, x)$
    - *Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex"*
  - Positive separability: $d(x, y) = 0$, if and only if $x = y$
    - *Otherwise there are objects that are different, but you cannot tell apart*
  - Triangular inequality: $d(x, y) \leq d(x, z) + d(z, y)$
    - *Otherwise you could claim "Alex is very like Bob, and Alex is very like Carl, but Bob is very unlike Carl"*

Distance functions for vectors

- Given two data points, both in R^n
  - $x = (x_1, x_2, \dots, x_n)^\top$
  - $y = (y_1, y_2, \dots, y_n)^\top$

- Euclidian distance: $d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$

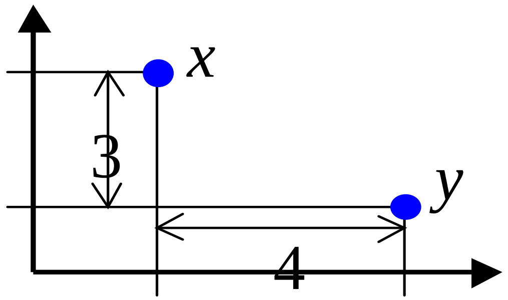
- Minkowski distance: $d(x, y) = \sqrt[p]{\sum_{i=1}^{n} (x_i - y_i)^p}$

  - Euclidian distance: $p = 2$

  - Manhattan distance: $p = 1, d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$

  - “inf”-distance: $p = \infty, d(x, y) = \max_{i=1}^{n} |x_i - y_i|$

Distance example



- Euclidian distance: $\sqrt{4^2 + 3^2} = 5$
- Manhattan distance: $4 + 3 = 7$
- “inf”-distance: $max\{4,3\} = 4$

K-Means Clustering

- The commonality within the same group is represented by the average value of data points.
    - Cluster centers

- The nearest cluster centers for any two points within the same group are the same

K-Means Clustering

- Given m data points, $\{x^1, x^2, \dots x^m\}$
- Find k cluster centers, $\{c^1, c^2, \dots, c^k\}$
- And assign each data point i to one cluster, $\pi(i) \in \{1, \dots, k\}$
- Such that the sum of the distances from each data point to its respective cluster center is minimized

$$\min_{c,\pi} \sum_{i=1}^{m} d\left(x^i, c^{\pi(i)}\right)$$

$$\operatorname*{argmin}_x f(x) = \{x | f(x) = \min_{x'} f(x')\}$$

$$\min_x f(x) = \{f(x) | f(x) < f(x_0) \forall x_0 \in R\}$$

K-Means Algorithm

- Step 1: Initialize k cluster centers, $\{c^1, c^2, \ldots, c^k\}$, randomly
- Step 2: Do
    - Decide the cluster memberships of each data point, x^i, by assigning it to the nearest cluster center (cluster assignment)

$$\pi(i) = argmin_{j=1,\ldots,k} \left\|x^i - c^j\right\|^2$$

    - Adjust the cluster centers (center adjustment)

$$c^j = \frac{1}{|\{i: \pi(i) = j\}|} \sum_{i:\pi(i)=j} x^i$$

- While any cluster center undergoes changes, go to Step 2

More on K-Means Clustering

Questions

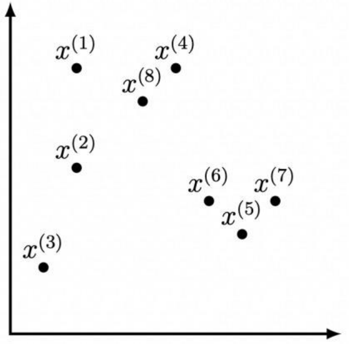
- Will different initialization lead to different results?
  - Yes
  - No

- Will the algorithm always stop after some iteration?
  - Yes (but it may not stop at the best clustering)
  - No

Exercise

[**12 points**]Suppose we would like to use the K-means algorithm and L2-norm distance (Euclidian distance) to cluster the 8 data points given in Figure 3 below into K = 3 clusters. The L2-norm distance between points $\mathbf{x} = (x_1, x_2)$ and $\mathbf{y} = (y_1, y_2)$ is $d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$. The coordinates of the data points are:

$$x^1 = (2,8) \quad x^2 = (2,5) \quad x^3 = (1,2) \quad x^4 = (5,8)$$
$$x^5 = (7,3) \quad x^6 = (6,4) \quad x^7 = (8,4) \quad x^8 = (4,7)$$



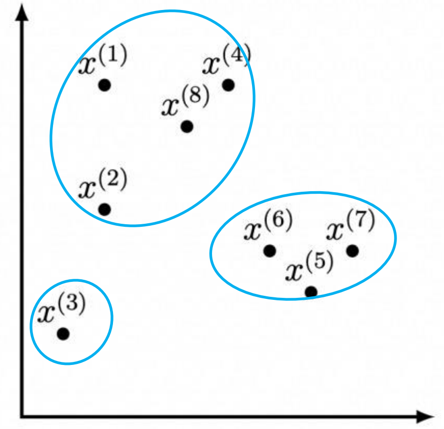
Suppose x^1, x^3, x^5 are chosen as the initial cluster centers. Report the coordinates of the updated cluster centers in the next step and assign each data point according to the new cluster centers.

Exercise

$d(x^1, c^1) = 0$
$d(x^2, c^1) = 3$
$d(x^3, c^1) = \sqrt{37}$
$d(x^4, c^1) = 3$
$d(x^5, c^1) = \sqrt{50}$
$d(x^6, c^1) = 32$
$d(x^7, c^1) = \sqrt{52}$
$d(x^8, c^1) = \sqrt{5}$

$d(x^1, c^2) = \sqrt{37}$
$d(x^2, c^2) = \sqrt{10}$
$d(x^3, c^2) = 0$
$d(x^4, c^2) = \sqrt{52}$
$d(x^5, c^2) = \sqrt{37}$
$d(x^6, c^2) = \sqrt{29}$
$d(x^7, c^2) = \sqrt{53}$
$d(x^8, c^2) = \sqrt{34}$

$d(x^1, c^3) = \sqrt{50}$
$d(x^2, c^3) = \sqrt{29}$
$d(x^3, c^3) = \sqrt{37}$
$d(x^4, c^3) = \sqrt{29}$
$d(x^5, c^3) = 0$
$d(x^6, c^3) = \sqrt{2}$
$d(x^7, c^3) = \sqrt{2}$
$d(x^8, c^3) = 5$



New cluster centers: (13/7, 7) (1,2) (7,11/3)

Model Selection

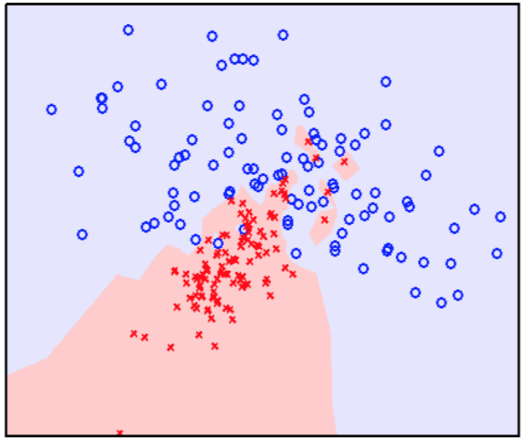
Question 1: Which supervised learning method should be selected?

Question 2: How many neighbors (K) should be chosen in KNN methods?

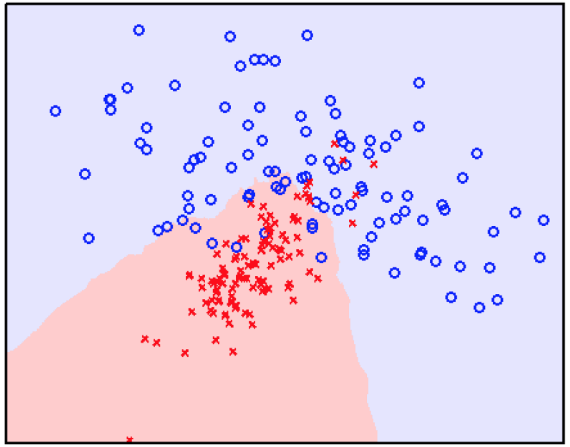
KNN

Overfitting

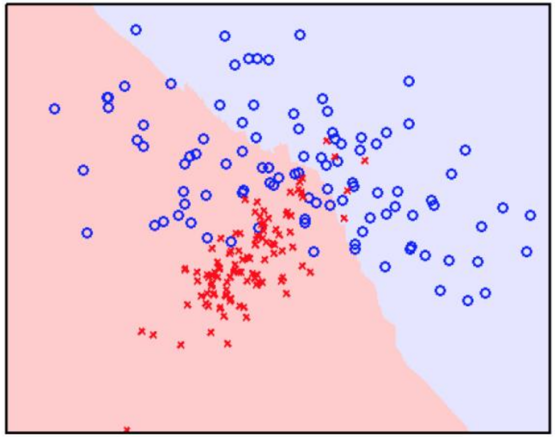
Underfitting



K = 1



K = 25



K = 101

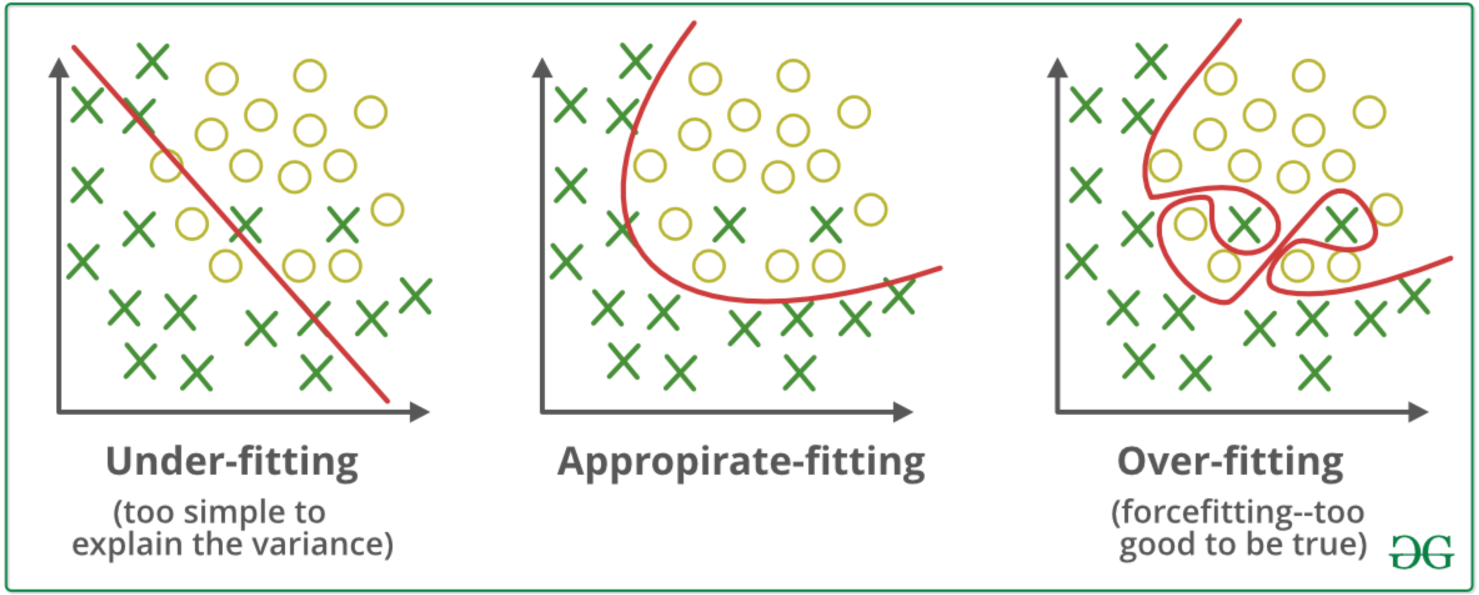
Overfitting and Underfitting

- Machine learning models are built to learn from training and test data, enabling them to make predictions on new, unseen datasets.

- A machine learning model is said to **overfit** the data when it learns patterns specific to the training data and make accurate predictions only on the training data.

- A machine learning model is said to **underfit** when it fails to capture the key patterns or relationships between variables in both the training and test data.

Overfitting and Underfitting


Under-fitting
(too simple to explain the variance)
Appropirate-fitting
Over-fitting
(forcefitting--too good to be true)

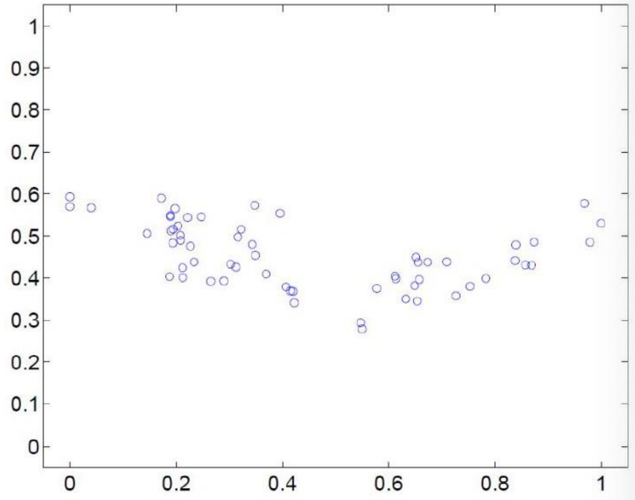
Example: Nonlinear Regression Model

- Want to fit a polynomial regression model

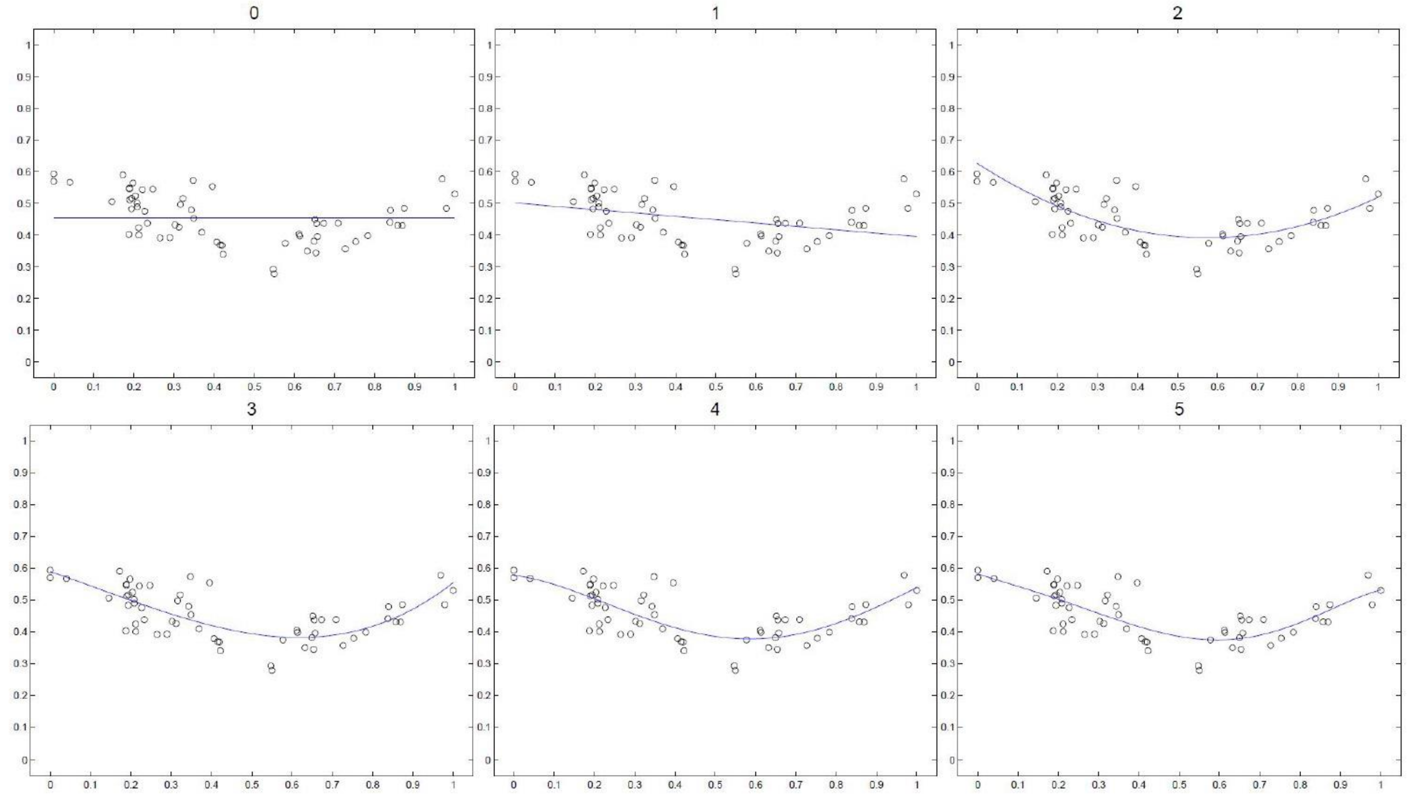
$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_n x^n + \epsilon$$

- Let $x = (1, x, x^2, \ldots, x^n)^\top$ and $\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)^\top$

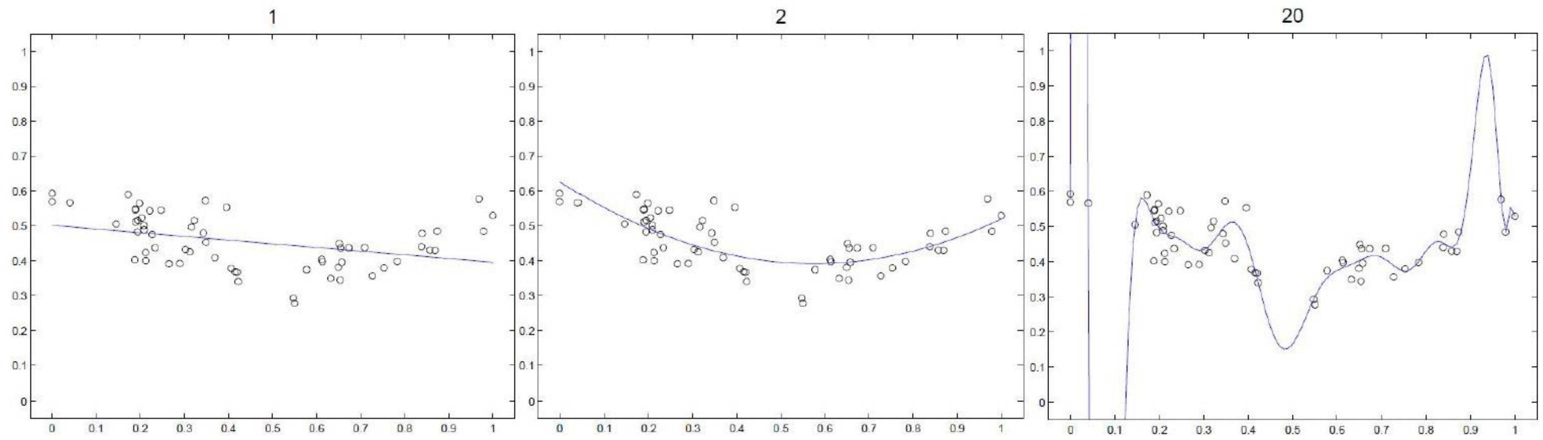
$$y = \theta^\top x$$



Increasing the maximal degree



Which one is better?



- Can we increase the maximal polynomial degree to very large, such that the curve passes through all training points?

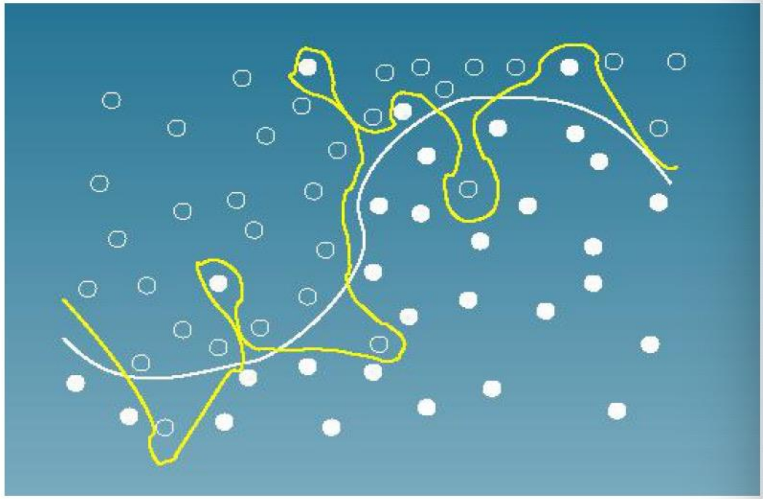
- The optimization does not prevent us from doing that

Example: Classification Model

- Logistic regression with polynomial features

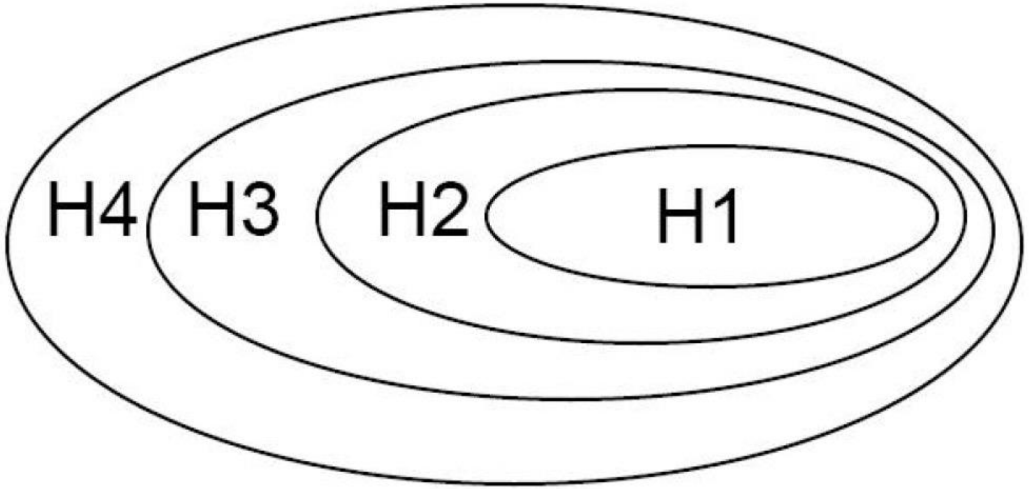
$$p(y = 1|x, \theta) = \frac{1}{1 + \exp\left(-\theta^\top x\right)}$$

- Let $x = (1, x, x^2, \ldots, x^n)^\top$ and $\theta = (\theta_0, \theta_1, \theta_2, \ldots, \theta_n)^\top$



Model space

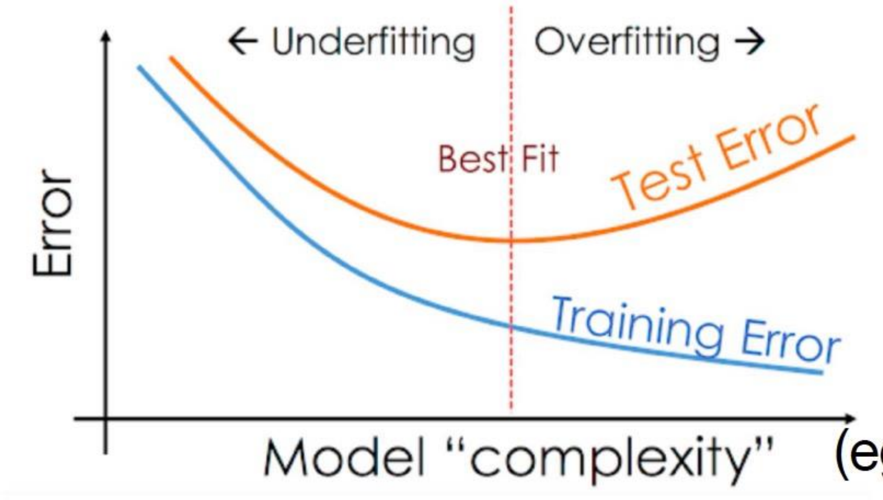
- Which model space should we choose?
- The more complex the model, the large the model space
- Eg. Polynomial function of degree 1, 2, … corresponds to space H1, H2 …



We Need To Select A Proper Model

Intuition of model selection

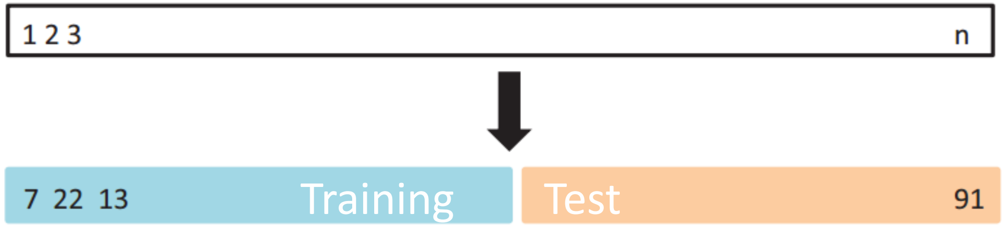
- Find the right model family s.t. test error becomes minimum



(eg, polynomial degree)

Validation Set Approach

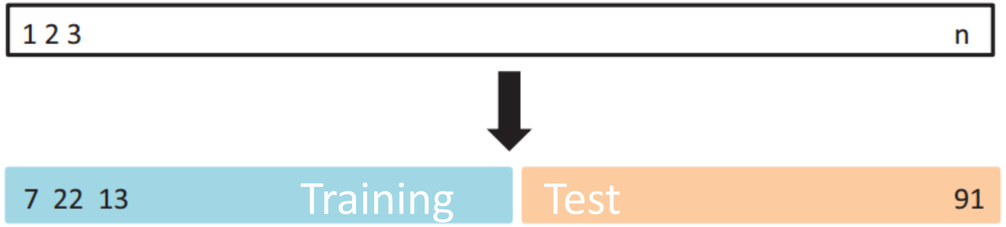
- Divide samples in to **training data** and **test data**.



- A set of model to choose {1,2,…, M}. (e.g, KNN, choose K=1,2,…M)
- For each model m,
  - use training data to train model
  - use the learned model to calculate the prediction error for test data. Err_m
- Choose model that has the smallest test error (min Err_m)

Validation Set Approach

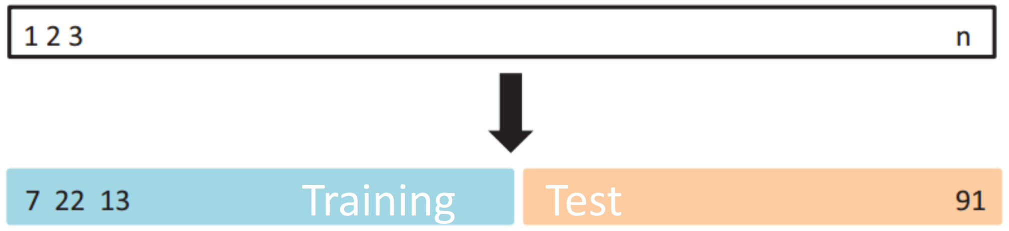
- Divide samples in to training data and test data.



- The selected model highly depends on your separation of the data.
- How to solve?

Validation Set Approach

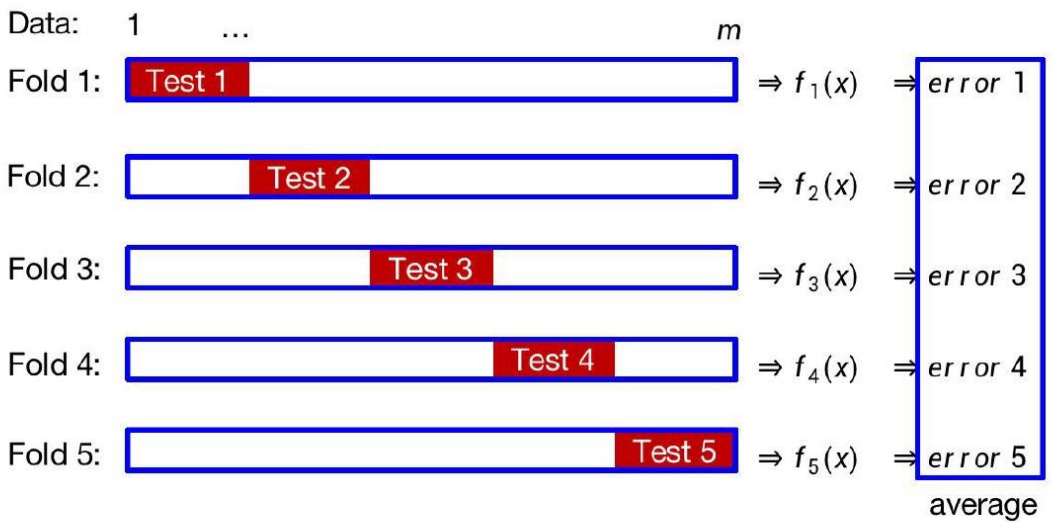
- Divide samples in to training data and test data.



- The selected model highly depends on your separation of the data.
- How to solve?
- **Repeat** the process several time; each time with different training and test data.

K-fold Validation

- 5-fold cross-validation (blank: training; red: test)

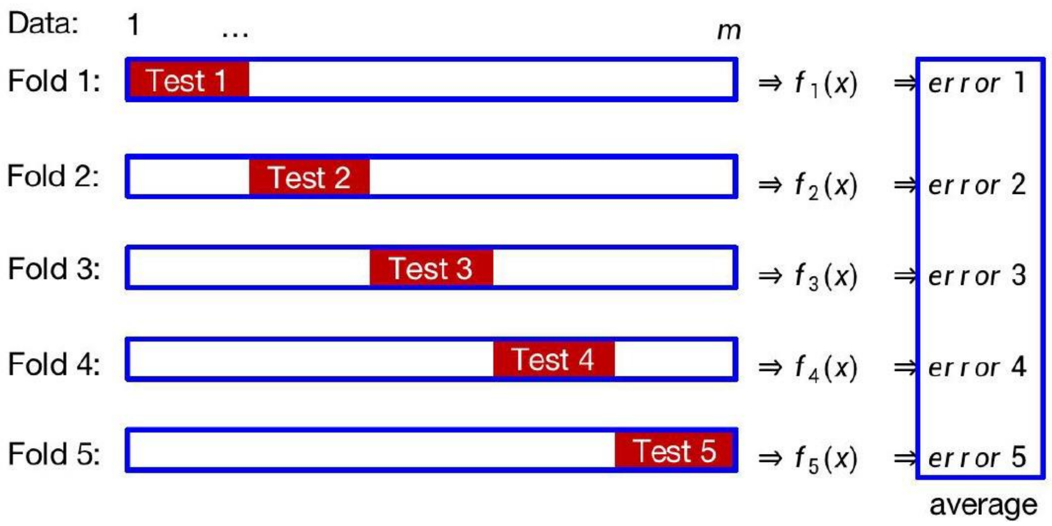


Data: 1 ... m
Fold 1: Test 1 ⇒ $f_1(x)$ ⇒ error 1
Fold 2: Test 2 ⇒ $f_2(x)$ ⇒ error 2
Fold 3: Test 3 ⇒ $f_3(x)$ ⇒ error 3
Fold 4: Test 4 ⇒ $f_4(x)$ ⇒ error 4
Fold 5: Test 5 ⇒ $f_5(x)$ ⇒ error 5
average

  - f_i is fitted by the training data in Fold i.
  - For each fold, use test data to test the prediction error.
  - Use the **average** error as the model's prediction error.

K-fold Validation

- 5-fold cross-validation (blank: training; red: test)



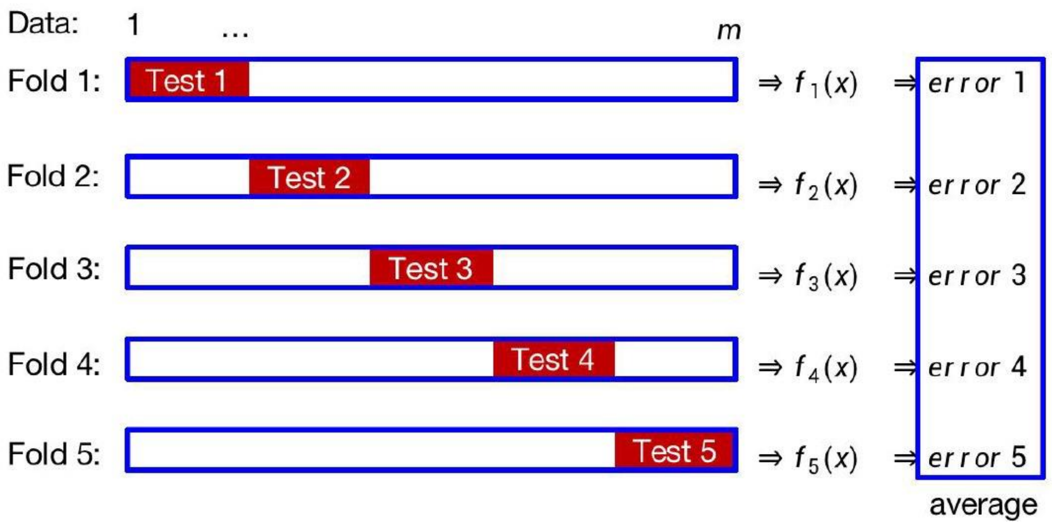
Data: 1 … m
Fold 1: Test 1 ⇒ $f_1(x)$ ⇒ error 1
Fold 2: Test 2 ⇒ $f_2(x)$ ⇒ error 2
Fold 3: Test 3 ⇒ $f_3(x)$ ⇒ error 3
Fold 4: Test 4 ⇒ $f_4(x)$ ⇒ error 4
Fold 5: Test 5 ⇒ $f_5(x)$ ⇒ error 5
average

Model 1 $h(\theta, X)$	Model 2 $g(\gamma, X)$

  - f_i is fitted by the training data in Fold i.
  - For each fold, use test data to test the prediction error.
  - Use the **average** error as the model's prediction error.

K-fold Validation

- 5-fold cross-validation (blank: training; red: test)


Data: 1 ... m
Fold 1: Test 1 ⇒ $f_1(x)$ ⇒ error 1
Fold 2: Test 2 ⇒ $f_2(x)$ ⇒ error 2
Fold 3: Test 3 ⇒ $f_3(x)$ ⇒ error 3
Fold 4: Test 4 ⇒ $f_4(x)$ ⇒ error 4
Fold 5: Test 5 ⇒ $f_5(x)$ ⇒ error 5
average

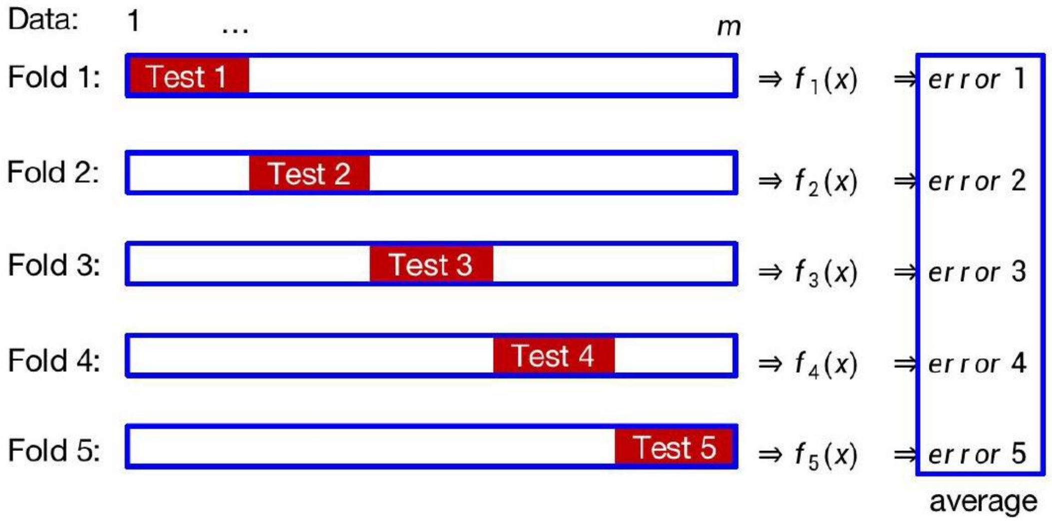
  - f_i is fitted by the training data in Fold i.
  - For each fold, use test data to test the prediction error.
  - Use the **average** error as the model's prediction error.

Use training data to train model

Model 1 h(θ, X)	**Model 2 g(γ, X)**
h($\boldsymbol{\theta}_1, \boldsymbol{X}$)	g($\boldsymbol{\gamma}_1, \boldsymbol{X}$)
h($\boldsymbol{\theta}_2, \boldsymbol{X}$)	g($\boldsymbol{\gamma}_2, \boldsymbol{X}$)
h($\boldsymbol{\theta}_3, \boldsymbol{X}$)	g($\boldsymbol{\gamma}_3, \boldsymbol{X}$)
h($\boldsymbol{\theta}_4, \boldsymbol{X}$)	g($\boldsymbol{\gamma}_4, \boldsymbol{X}$)
h($\boldsymbol{\theta}_5, \boldsymbol{X}$)	g($\boldsymbol{\gamma}_5, \boldsymbol{X}$)

K-fold Validation

- 5-fold cross-validation (blank: training; red: test)


Data: 1 ... m
Fold 1: Test 1 ⇒ $f_1(x)$ ⇒ error 1
Fold 2: Test 2 ⇒ $f_2(x)$ ⇒ error 2
Fold 3: Test 3 ⇒ $f_3(x)$ ⇒ error 3
Fold 4: Test 4 ⇒ $f_4(x)$ ⇒ error 4
Fold 5: Test 5 ⇒ $f_5(x)$ ⇒ error 5
average

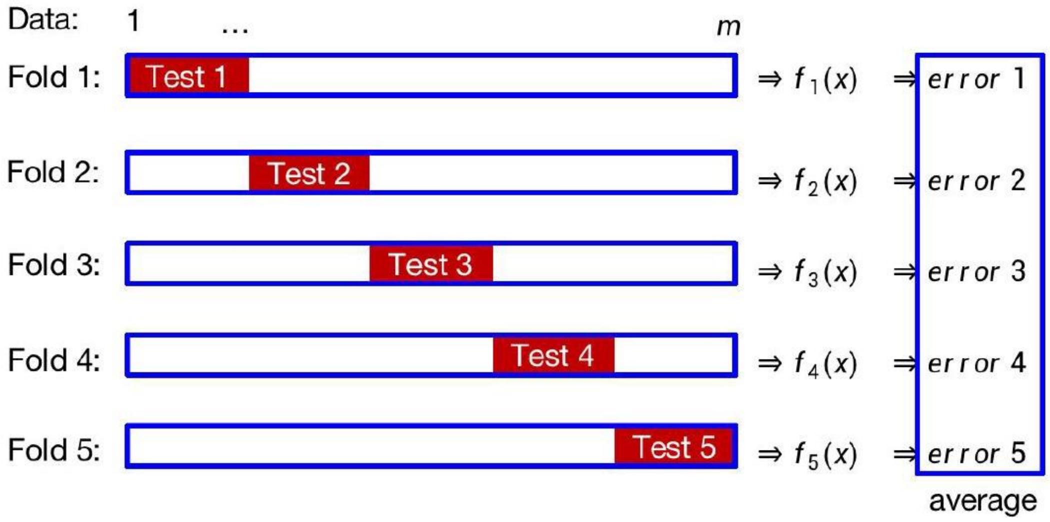
Error

Model 1 $h(\boldsymbol{\theta}, X)$	Model 2 $g(\boldsymbol{\gamma}, X)$
$\mathrm{Err}_h(\boldsymbol{\theta}_1)$	$\mathrm{Err}_g(\boldsymbol{\gamma}_1)$
$\mathrm{Err}_h(\boldsymbol{\theta}_2)$	$\mathrm{Err}_g(\boldsymbol{\gamma}_2)$
$\mathrm{Err}_h(\boldsymbol{\theta}_3)$	$\mathrm{Err}_g(\boldsymbol{\gamma}_3)$
$\mathrm{Err}_h(\boldsymbol{\theta}_4)$	$\mathrm{Err}_g(\boldsymbol{\gamma}_4)$
$\mathrm{Err}_h(\boldsymbol{\theta}_5)$	$\mathrm{Err}_g(\boldsymbol{\gamma}_5)$

  - f_i is fitted by the training data in Fold i.
  - For each fold, use test data to test the prediction error.
  - Use the **average** error as the model's prediction error.

K-fold Validation

- 5-fold cross-validation (blank: training; red: test)



Data: 1 ... m
Fold 1: Test 1 ⇒ $f_1(x)$ ⇒ error 1
Fold 2: Test 2 ⇒ $f_2(x)$ ⇒ error 2
Fold 3: Test 3 ⇒ $f_3(x)$ ⇒ error 3
Fold 4: Test 4 ⇒ $f_4(x)$ ⇒ error 4
Fold 5: Test 5 ⇒ $f_5(x)$ ⇒ error 5
average

  - f_i is fitted by the training data in Fold i.
  - For each fold, use test data to test the prediction error.
  - Use the **average** error as the model's prediction error.

Error

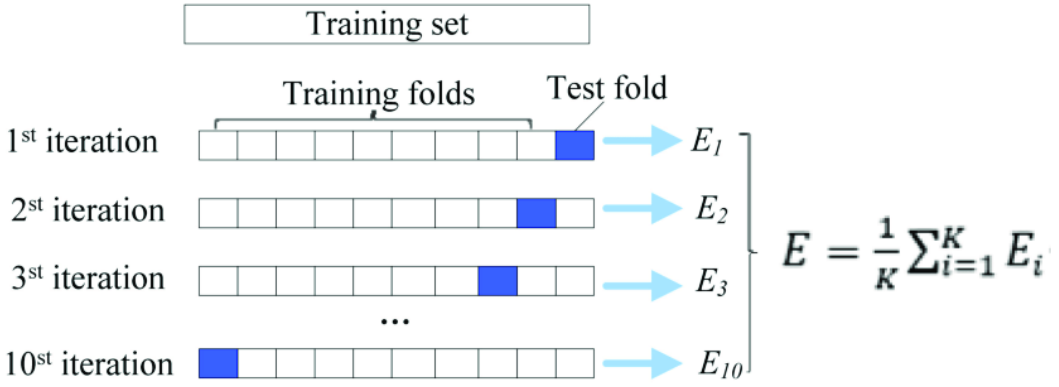
Model 1 h(θ, X)	**Model 2 g(γ, X)**
$\mathrm{Err}_h(\boldsymbol{\theta}_1)$	$\mathrm{Err}_g(\boldsymbol{\gamma}_1)$
$\mathrm{Err}_h(\boldsymbol{\theta}_2)$	$\mathrm{Err}_g(\boldsymbol{\gamma}_2)$
$\mathrm{Err}_h(\boldsymbol{\theta}_3)$	$\mathrm{Err}_g(\boldsymbol{\gamma}_3)$
$\mathrm{Err}_h(\boldsymbol{\theta}_4)$	$\mathrm{Err}_g(\boldsymbol{\gamma}_4)$
$\mathrm{Err}_h(\boldsymbol{\theta}_5)$	$\mathrm{Err}_g(\boldsymbol{\gamma}_5)$

Model 1 is better iff.

$$\frac{1}{K}\sum_i Err_h(\theta_i) < \frac{1}{K}\sum_i Err_g(\gamma_i)$$

K-fold Validation

- How to decide the values for K (or a)
  - Commonly used $K = 10$ or $(a = 0.1)$.
  - Large K makes it time-consuming.



Error

Model 1 $\mathbf{h}(\boldsymbol{\theta}, \boldsymbol{X})$	**Model 2** $\mathbf{g}(\boldsymbol{\gamma}, \boldsymbol{X})$
$\mathrm{Err_h}(\boldsymbol{\theta}_1)$	$\mathrm{Err_g}(\boldsymbol{\gamma}_1)$
$\mathrm{Err_h}(\boldsymbol{\theta}_2)$	$\mathrm{Err_g}(\boldsymbol{\gamma}_2)$
$\mathrm{Err_h}(\boldsymbol{\theta}_3)$	$\mathrm{Err_g}(\boldsymbol{\gamma}_3)$
$\mathrm{Err_h}(\boldsymbol{\theta}_4)$	$\mathrm{Err_g}(\boldsymbol{\gamma}_4)$
$\mathrm{Err_h}(\boldsymbol{\theta}_5)$	$\mathrm{Err_g}(\boldsymbol{\gamma}_5)$

Model 1 is better iff.

$$\frac{1}{K}\sum_i Err_h(\theta_i) < \frac{1}{K}\sum_i Err_g(\gamma_i)$$