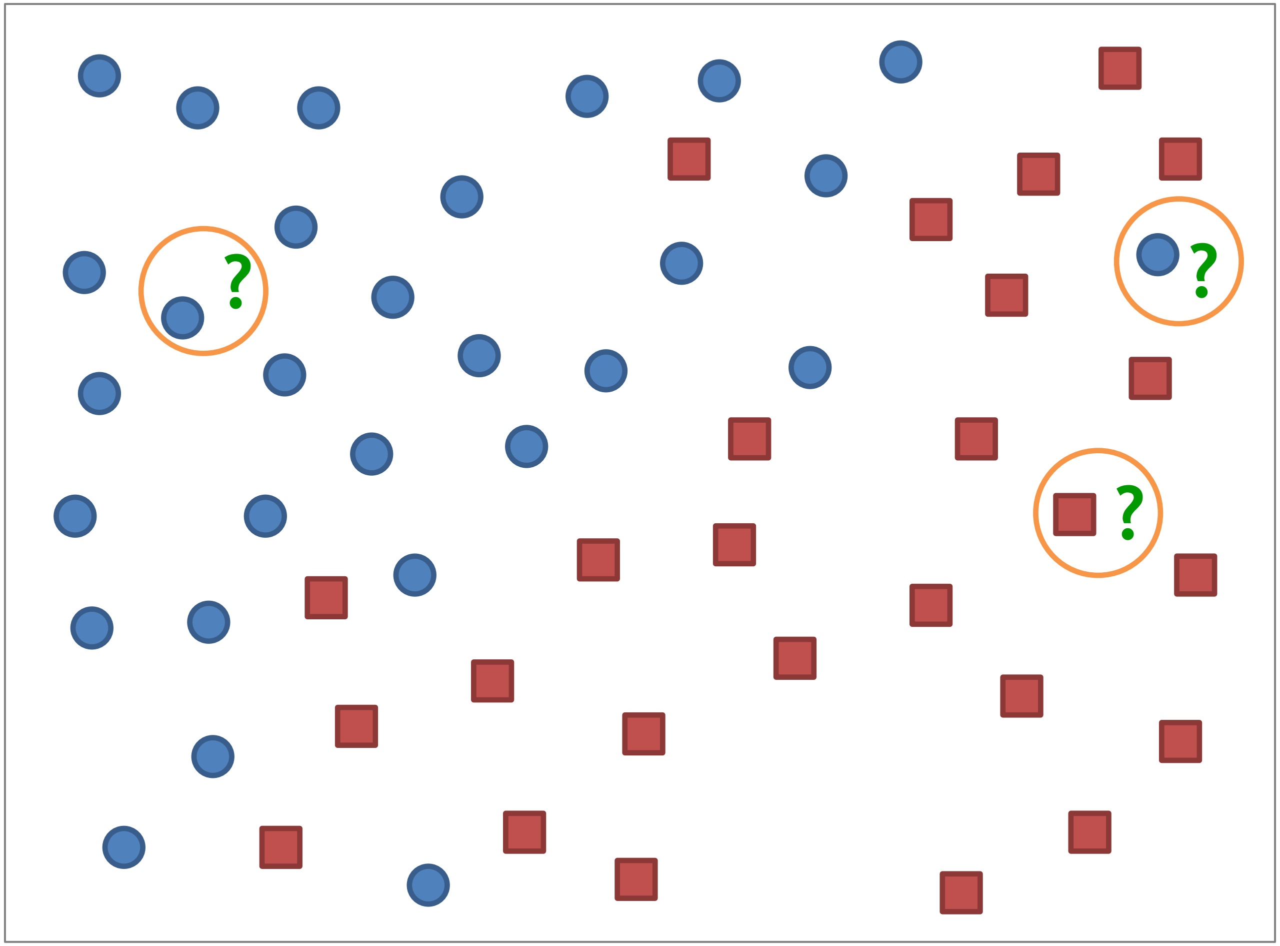
K-NEAREST NEIGHBORS

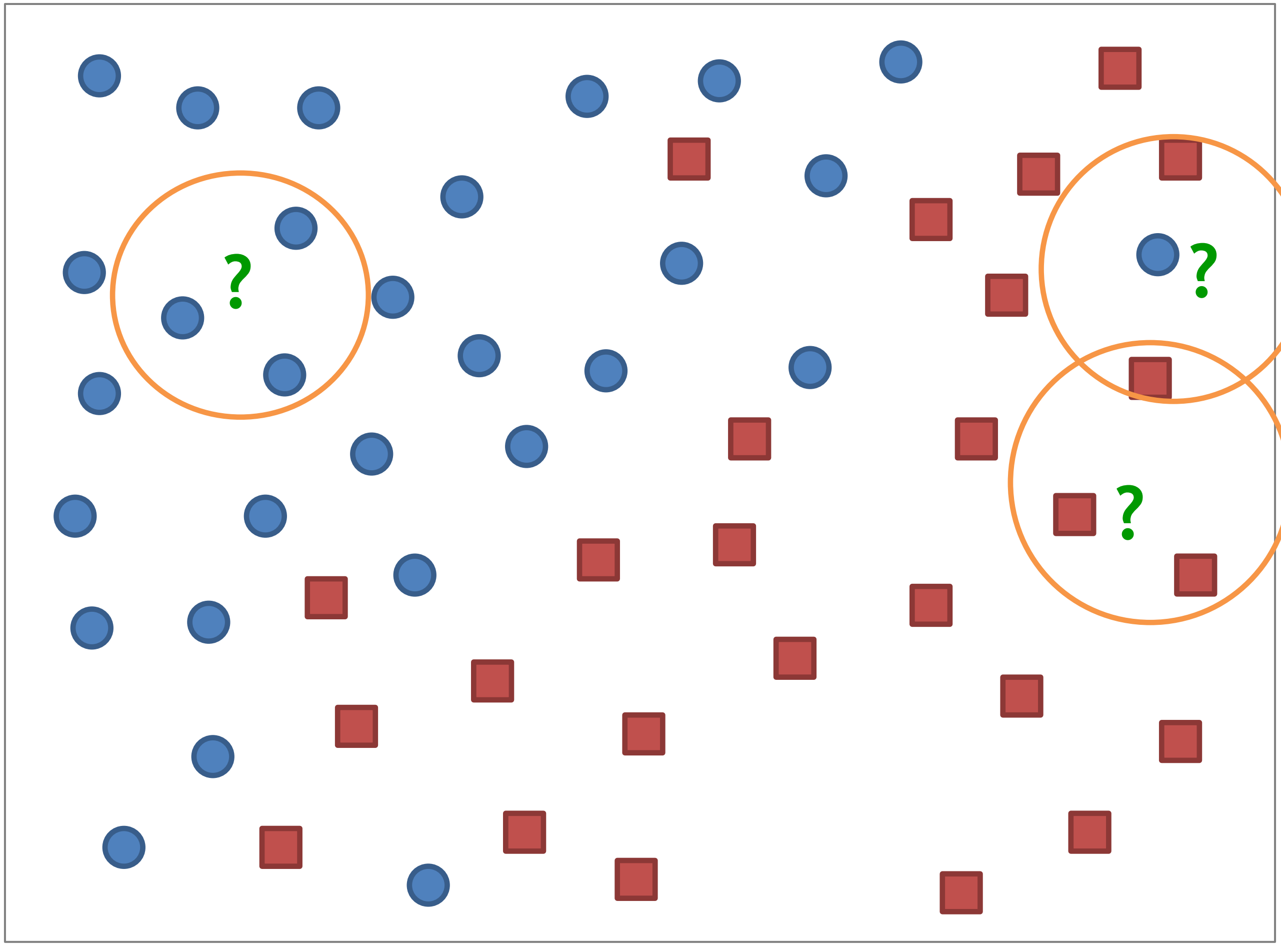
k-NN classification: Example

- Classify a new instance to the nearest neighbor's class



k-NN classification: Example

- Classify a new instance to the 3 nearest neighbors' class



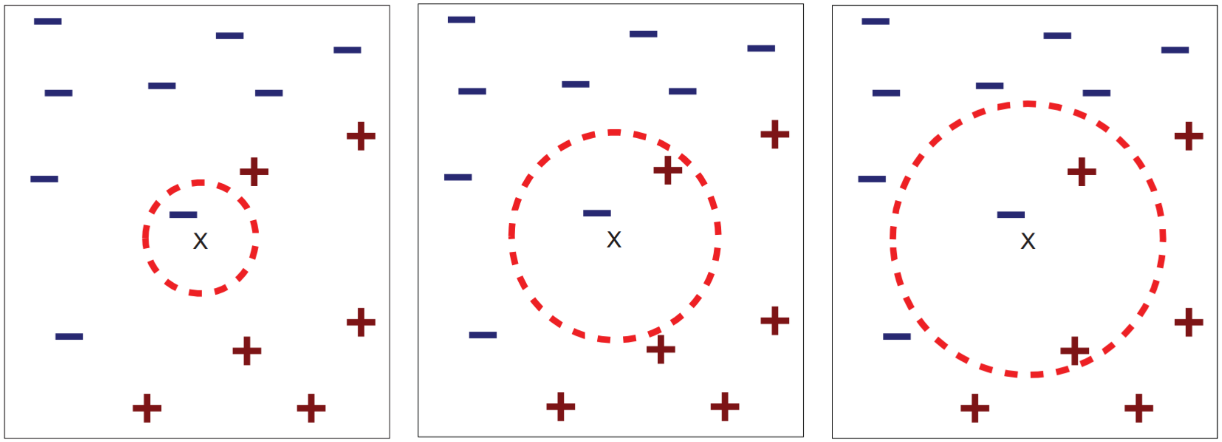
k-NN: Inference

- **Given training data $(\mathbf{X}, \mathbf{Y})$**
  - $\mathbf{X}$: Input variables
  - $\mathbf{Y}$: Output variable
- **Suppose there is a new point Q**
  - For i in range(1,number of training points)
    - Compute $\boxed{\text{distance } d(X_i, Q)}$
  - Compute set I containing indices for the $\boxed{k}$ smallest distances $d(X_i, Q)$
  - $\boxed{\textbf{Return } \hat{y} \text{ corresponding to the new point } Q \text{ using } \{y_i \text{ for } i \in I\}}$

k-nearest neighbors (k-NN)

❖ k-nearest neighbors (k-최근접 이웃) algorithm

- Predict the class (value) of a query point based the information of the k number of nearest neighbors
- One of the simplest machine learning algorithms

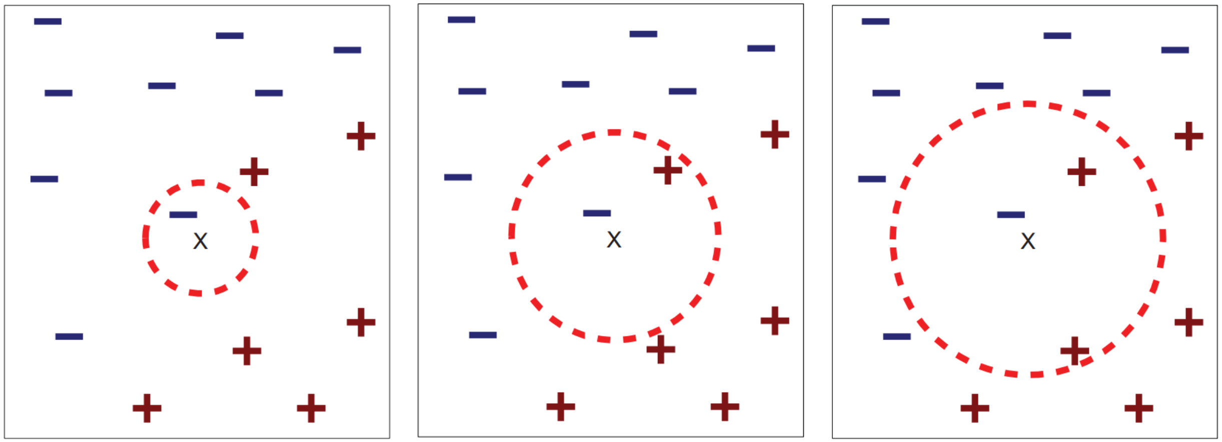


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

k-nearest neighbors (k-NN)

❖ k-nearest neighbors (k-최근접 이웃) algorithm

- Predict the class (value) of a query point based the information of the k number of nearest neighbors
- One of the simplest machine learning algorithms
- No explicit training or model
- Can be used both for classification and regression



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

k-NN: Distance and similarity measures

How to measure the distance $d(X, Q)$

training point: $X = (x_1,\ x_2, \cdots, x_p)^T$

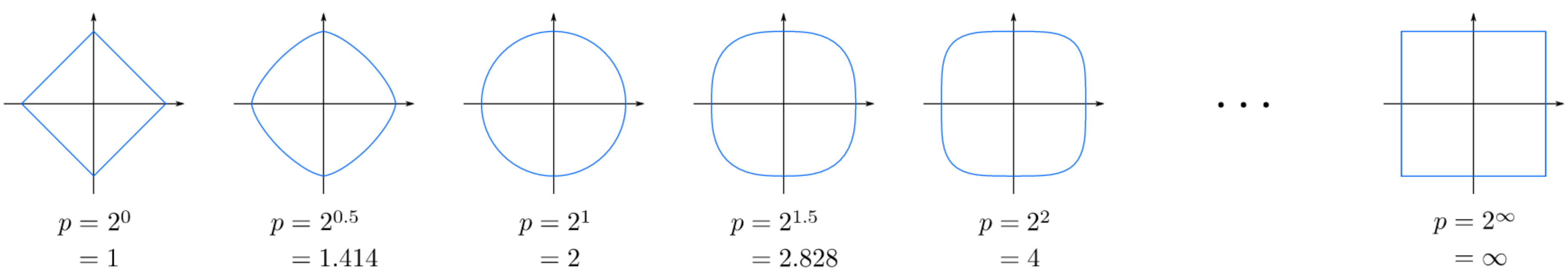
query point: $Q = (q_1,\ q_2, \cdots, q_p)^T$

- **Minkovski distance with order *p***

$$d(X,Q) = \left(\sum_{i=1}^{p} |x_i - q_i|^p \right)^{\frac{1}{p}}$$

  - When p = 2, it is the Euclidean distance.

$$d(X,Q) = \left(\sum_{i=1}^{p} |x_i - q_i|^2 \right)^{\frac{1}{2}} = \sqrt{(x_1 - q_1)^2 + \cdots + (x_p - q_p)^2}$$



<Minkovski distance>

k-NN: Distance and similarity measures

How to measure the distance $d(X, Q)$

= How to measure the similarity $sim(X, Q)$

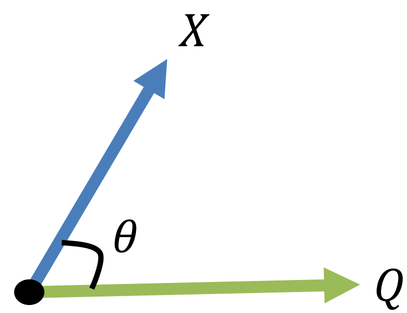
training point: $X = (x_1,\ x_2, \cdots, x_p)^T$

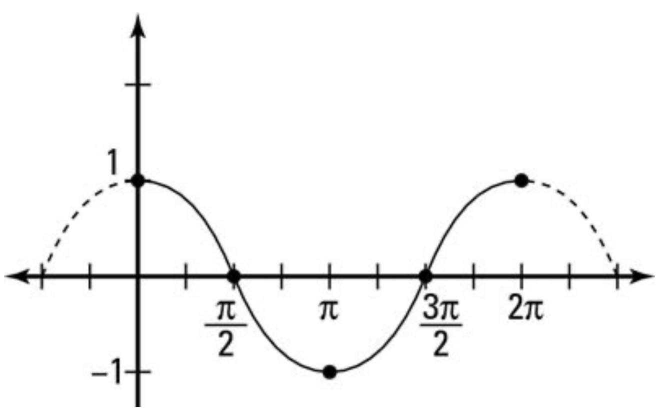
query point: $Q = (q_1,\ q_2, \cdots, q_p)^T$

❖ Cosine similarity

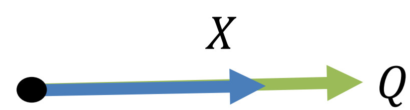
- Bounded between 0 and 1 if *X* and *Q* are nonnegative

$$sim(X, Q) = \cos\theta = \frac{X \bullet Q}{\|X\|\|Q\|} = \frac{\sum_{i=1}^{p} x_i q_i}{\sqrt{\sum_{i=1}^{p} x_i^2}\sqrt{\sum_{i=1}^{p} q_i^2}}$$





$$\cos 0 = 1$$



$$\cos 90^\circ = 0$$

k-NN: Distance and similarity measures

How to measure the distance $d(X,Q)$

= How to measure the similarity $sim(X,Q)$

training point: $X = (x_1,\ x_2, \cdots, x_p)^T$

query point: $Q = (q_1,\ q_2, \cdots, q_p)^T$

❖ Pearson's correlation coefficient

- Bounded between -1 and 1
- Equal to cosine similarity with zero-centered *X* and *Q*

$$sim\left(X,Q\right) = r(X,Q) = \frac{\sum_{i=1}^{p}(x_i - \overline{x})(q_i - \overline{q})}{\sqrt{\sum_{i=1}^{p}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{p}(q_i - \overline{q})^2}}$$

k-NN: Distance and similarity measures

Euclidean distance

- Most popular
- To normalize the feature vector (scaling)
  - ex) Income varies 10,000-1,000,000 while height varies 1.5-1.8 meters
  - Normalization or Standardization!
- If dimensionality increase, distance would also increase

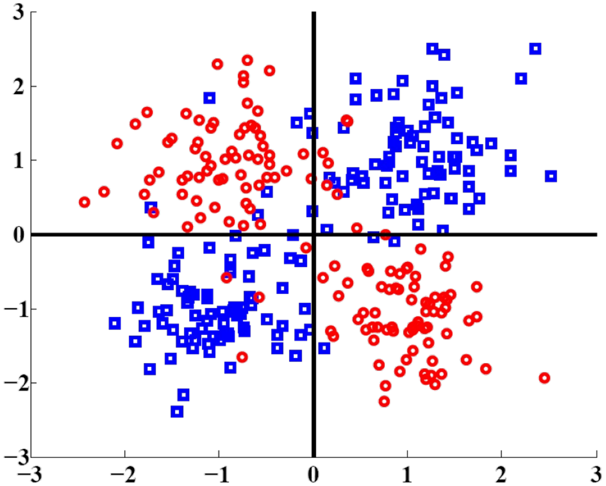
Cosine similarity and Pearson’s correlation coefficient

- Usually used for Sparse matrix,
  - ex) Document-term matrix for text mining

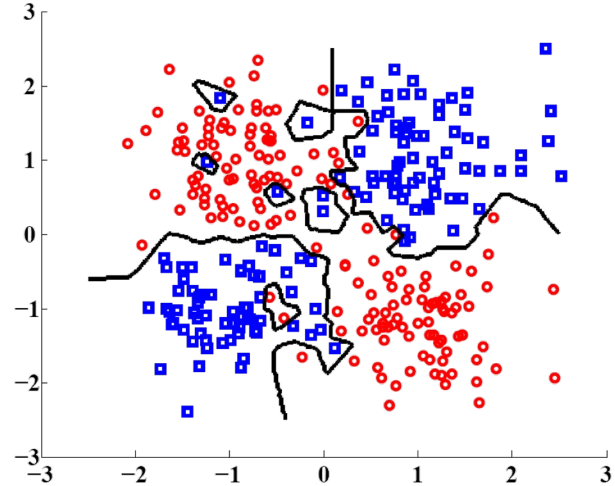
k-NN: How to select *k*

- **Compare the validation/test performances across various k values**
  - Compare Predictive performances (for classification or regression)
- If k is too small, it can lead to over-fitting and be sensitive to local noise.
- If k is too large, it may not effectively capture local data structures.

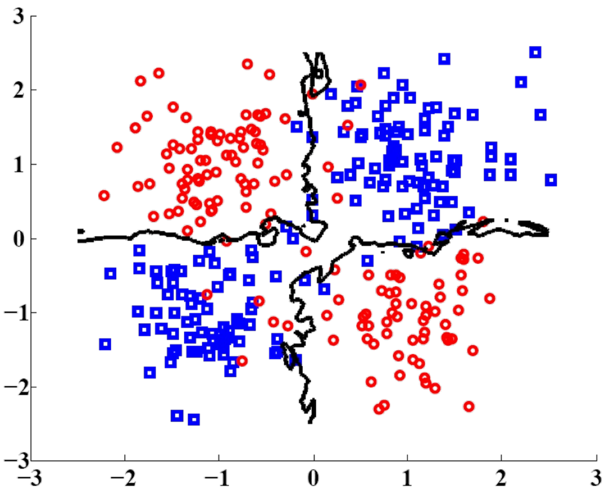
k-NN: How to select *k*



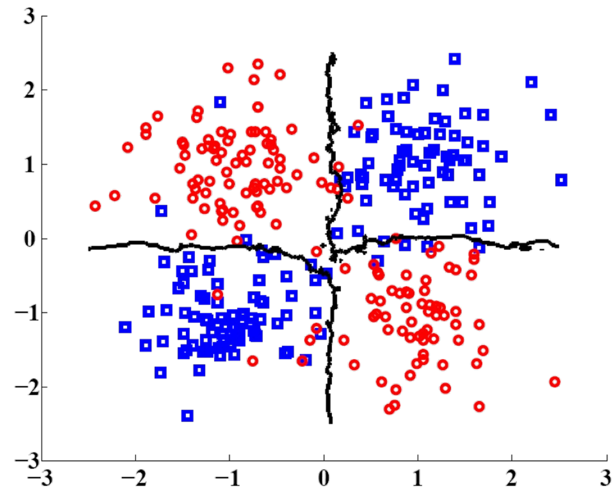
(a) The ideal boundary.



(b) k-NN classification with $k = 1$.



(c) k-NN classification with $k = 10$.



(d) k-NN classification with $k = 50$.

k-NN: How to classify a new point

Majority voting vs. Weighted voting

- Majority voting
  - Classify a new point as the majority class
- Weighted voting
  - Assign 'weight' to the contribution of the neighbors.
  - Common weighting scheme

- distance between a new point and i^{th} neighbor: d_i

- weight for i^{th} neighbor : $w_i = \dfrac{1/d_i}{\sum_{j=1}^{k}(\frac{1}{d_j})}$

- Sum of weights: $\sum_{i=1}^{k} w_i = 1$

k-NN: How to classify a new point

Example 1: k=5

For a new point **Q**

Neighbor	Class	Distance	1/distance	Weight
N1	M	1	1.00	0.44
N2	F	2	0.50	0.22
N3	M	3	0.33	0.15
N4	F	4	0.25	0.11
N5	F	5	0.20	0.08

- Majority voting: $P(\hat{Y} = \mathrm{M}) = \frac{2}{5} = 0.4,\ \ P(\hat{Y} = \mathrm{F}) = 1 - 0.4 = 0.6$
- Weighted voting: $P(\hat{Y} = \mathrm{M}) = 0.44 + 0.15 = 0.59,$

$$P(\hat{Y} = \mathrm{F}) = 1 - 0.59 = 0.41$$

- Q is classified as <u>**F** by the majority voting</u>, while classified as <u>**M** by the weighted voting</u>

k-NN: Pros and Cons

Pros

- Simple and powerful. No need for tuning complex parameters to build a model.
- No training involved (“lazy”). New training examples can be added easily.

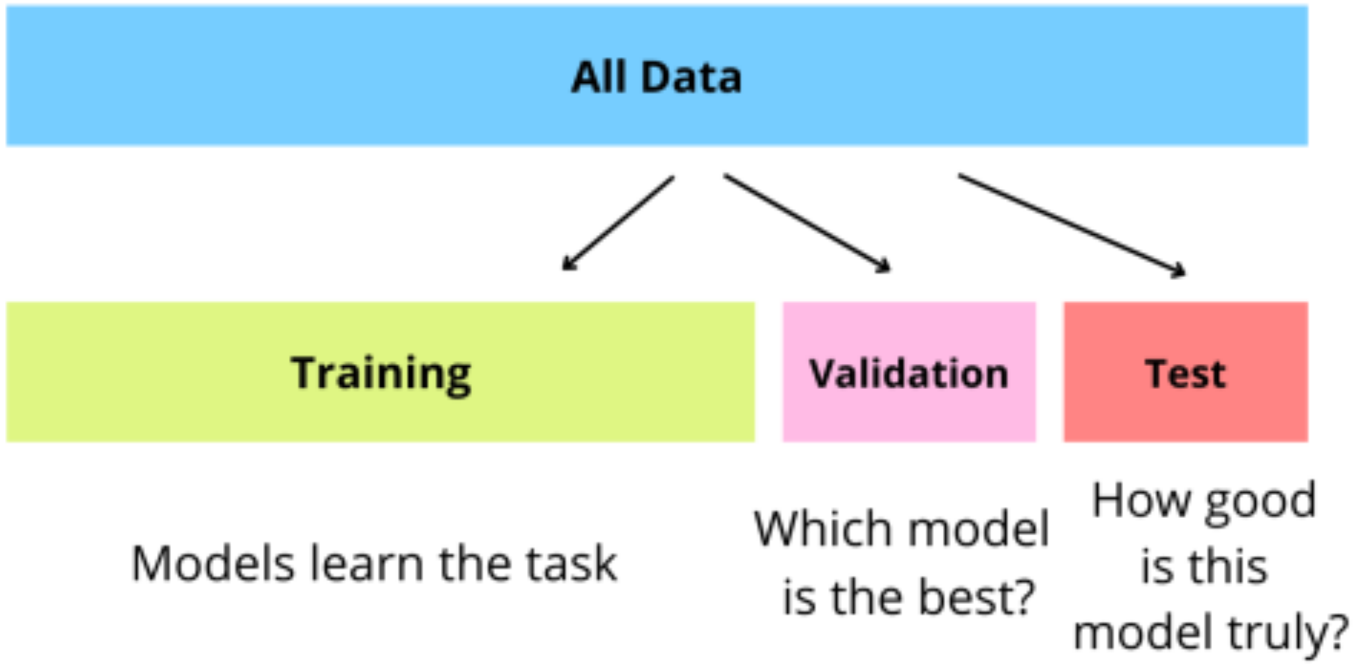
k-NN: Pros and Cons

Cons

- Expensive and slow: O(md), m= # examples, d= # dimensions
  - To determine the nearest neighbor of a new point x, must compute the distance to all m training examples. Runtime performance is slow, but can be improved.
    - Pre-sort training examples into fast data structures
    - Compute only an approximate distance
    - Remove redundant data (condensing)

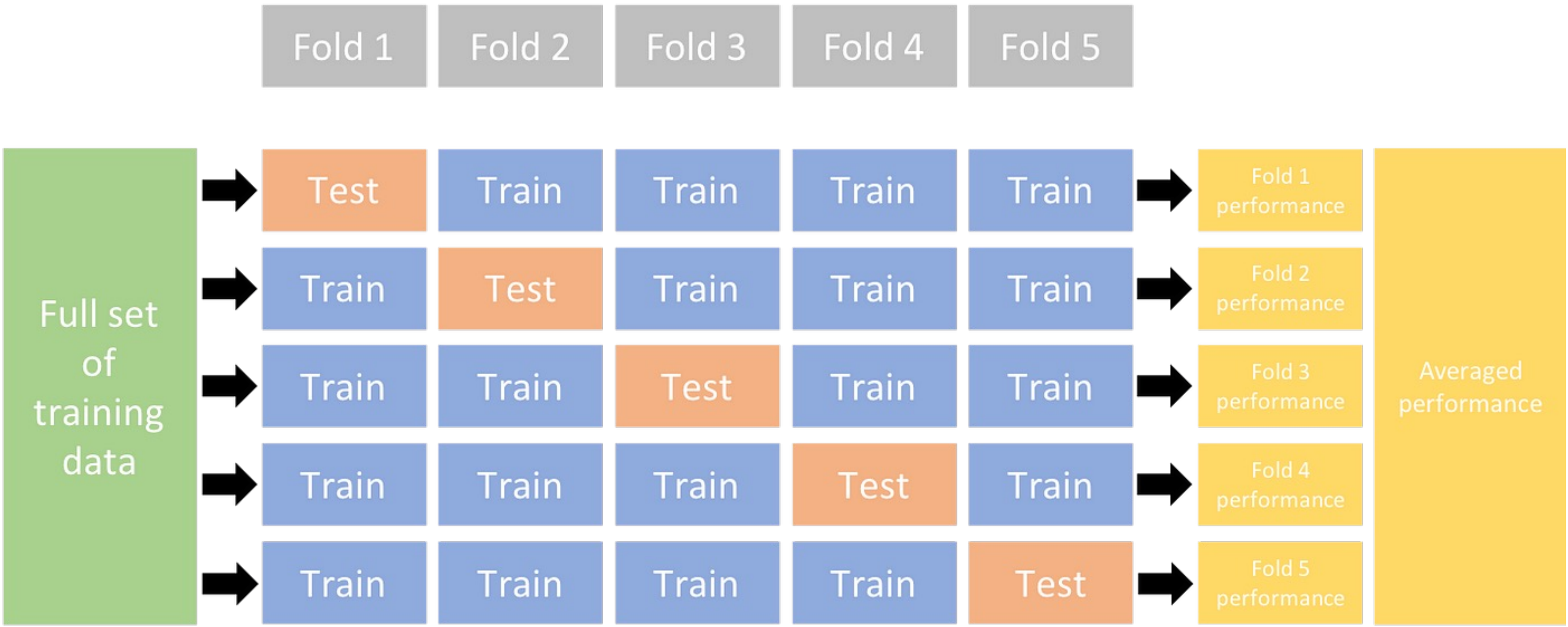
MISCELLANEOUS CONTENTS

Training, Validation and Testing



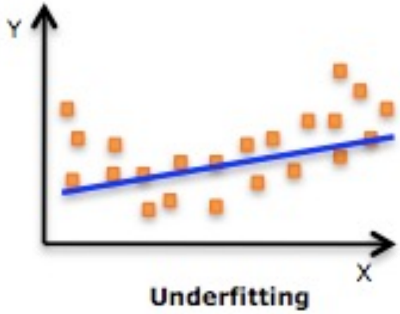
Cross-Validation

- dividing the data into multiple folds, using each fold once as a test set and th e rest as a training set
- Ex) 5-fold CV
  - the data is split into 5 parts
  - rotating each part as the test set while using the remaining 4 parts for training
- In situations with limited data, it is possible to achieve robust learning.



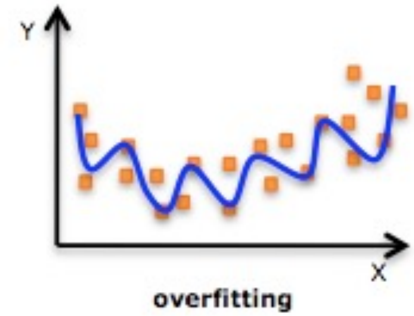
Overfitting & Model Selection

$$y = a_2 x^2 + a_1 x + a_0 + Noise$$



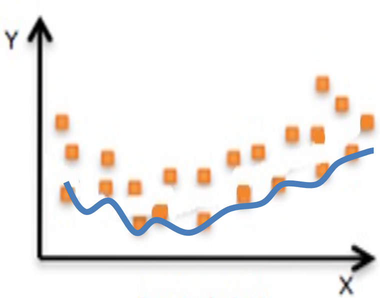
$$y = a_1 x + a_0$$

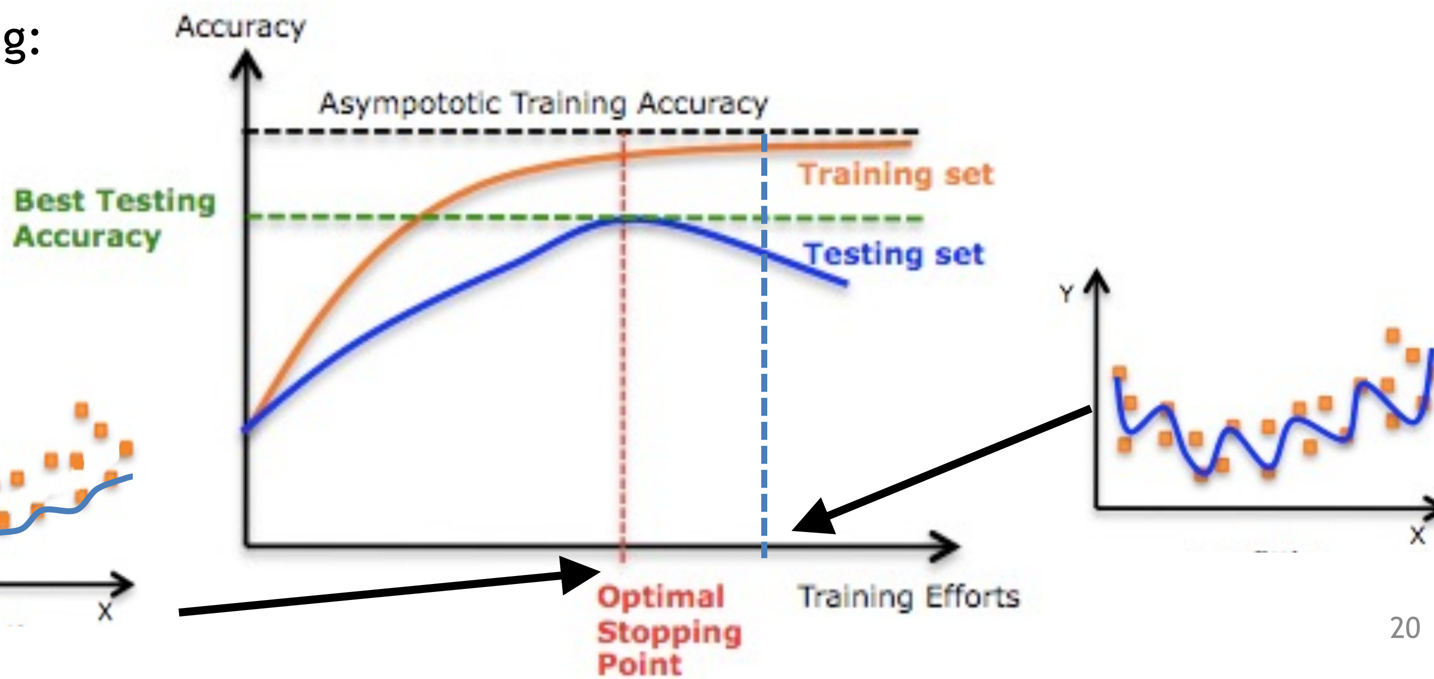
$$y = a_2 x^2 + a_1 x + a_0$$



$$y = a_{10} x^{10} + \ldots + a_1 x + a_0$$

Early Stopping:





Occam’s Razor & Model Selection

- Simpler explanations are, other things being equal, generally better than more complex ones.
  - http://homa.egloos.com/page/6
- ↔ Hickam’s dictum
- Model of “True Nature”?
  - Shortage of data

    → Simple model

    → Reasonable test error
  - More data or Regularization