Introduction to Artificial Intelligence

COMP307/AIML420

Machine Learning 2: 3-K Techniques

Dr Andrew Lensen

Andrew.Lensen@vuw.ac.nz

Outline

- Revisiting **Generalisation**

- **K**-Nearest Neighbour method
  - Classification (Supervised learning)
  - Basic NN (1-NN)
  - K-Nearest Neighbour (KNN)
  - Distance/Similarity measure

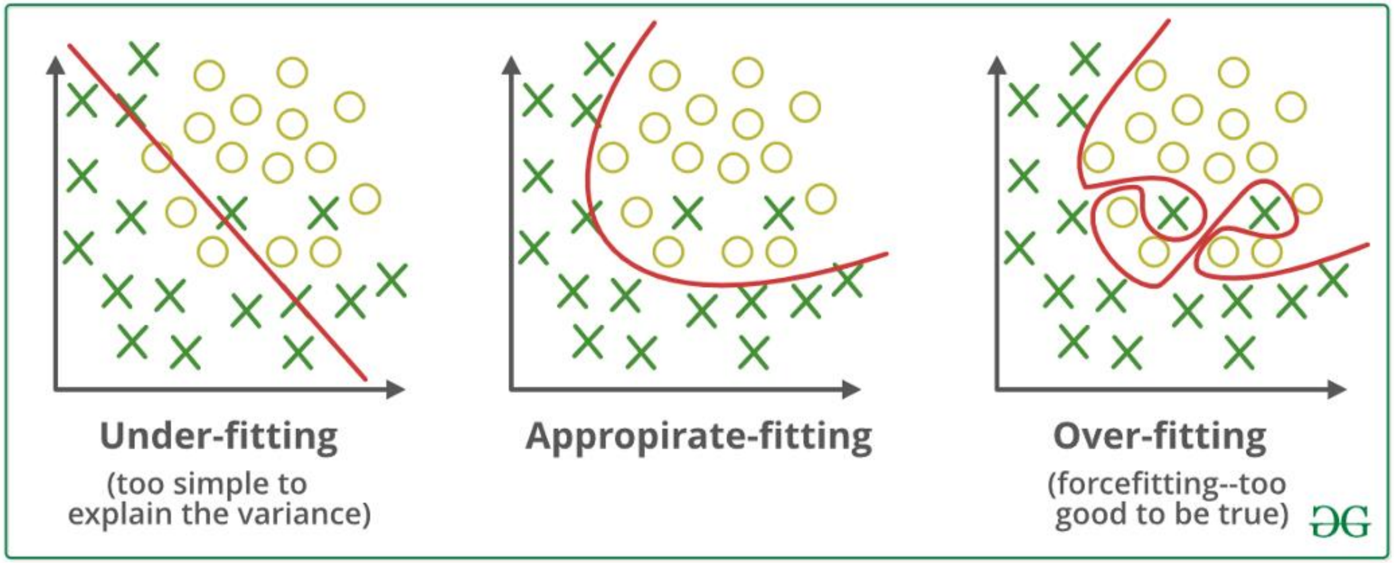
- **K**-fold cross validation
  - Leave-one out cross validation
  - K-fold cross validation vs validation set

- **K**-means clustering
  - Unsupervised learning

Generalisation

- We learn a classifier/predictor/model from the training data
- But performing well on training data is **NOT** enough!
- Important to evaluate the performance on the **test (unseen) data – generalisation**

- If too biased to the training data, this may cause *overfitting*: too good on the training data, but poor on test data

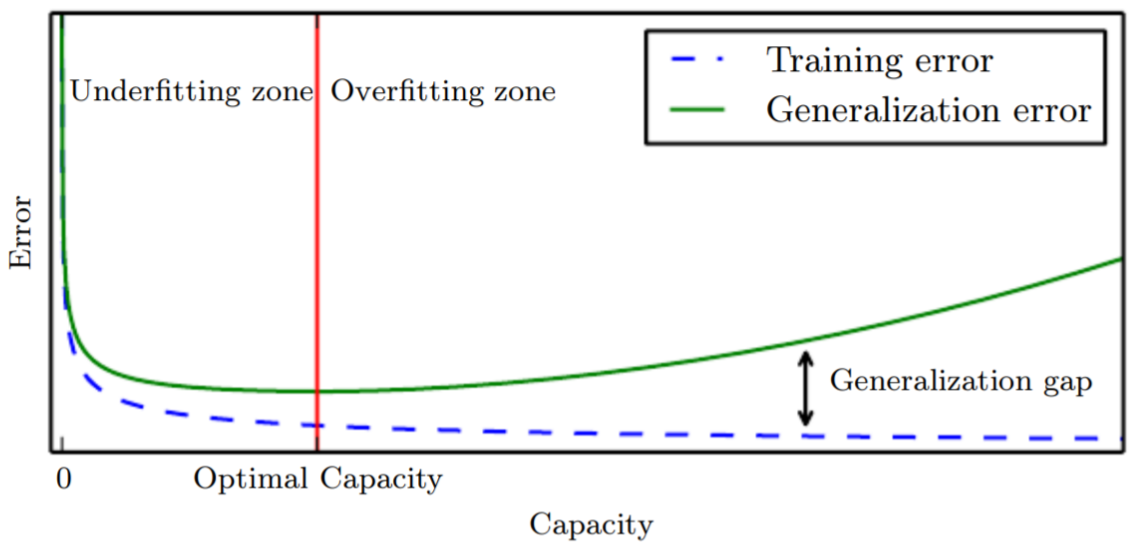


https://towardsdatascience.com/underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6fe4a8a49dbf

Generalisation

- Why? Our training data nearly always has some "signal" and some "noise".
- Learning **too** well means capturing the "noise"!

- E.g. one COMP307 student in 2020 is 2m tall, and gets an A+
  - Overfitted AI algorithm: "Students over 2m tall always get an A+!"
  - Well-fitted AI algorithm: doesn't consider height at all.



Classification for *Iris* Dataset

- **Three** classes of *Iris* flower
- **Four** features
  - **Length** (cms) and **width** (cms) of:
    - Sepals
    - Petals
- **150** instances (lines)
  - **50** for each class

Setosa

Versicolor

Virginica

```
5.1, 3.5, 1.4, 0.2, Iris-setosa
4.9, 3.0, 1.4, 0.2, Iris-setosa
…
7.0, 3.2, 4.7, 1.4, Iris-versicolor
6.4, 3.2, 4.5, 1.5, Iris-versicolor
…
6.3, 3.3, 6.0, 2.5, Iris-virginica
5.8, 2.7, 5.1, 1.9, Iris-virginica
…
```

(1-)Nearest Neighbour Classifier

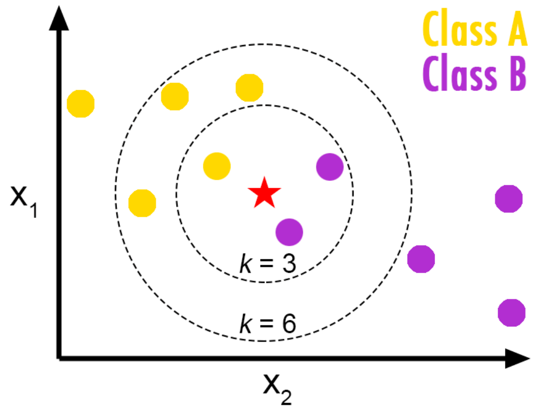
- **Directly** use the training instances to classify an unseen instance (e.g. test instance)
  - Assign class from the most similar training instance

1. **Compare** the unseen instance to all the training instances;
2. Find the "nearest neighbour" (the training instance that is closest to the unseen instance) from the training set;
3. Classify the unseen instance as the class of the nearest neighbour.

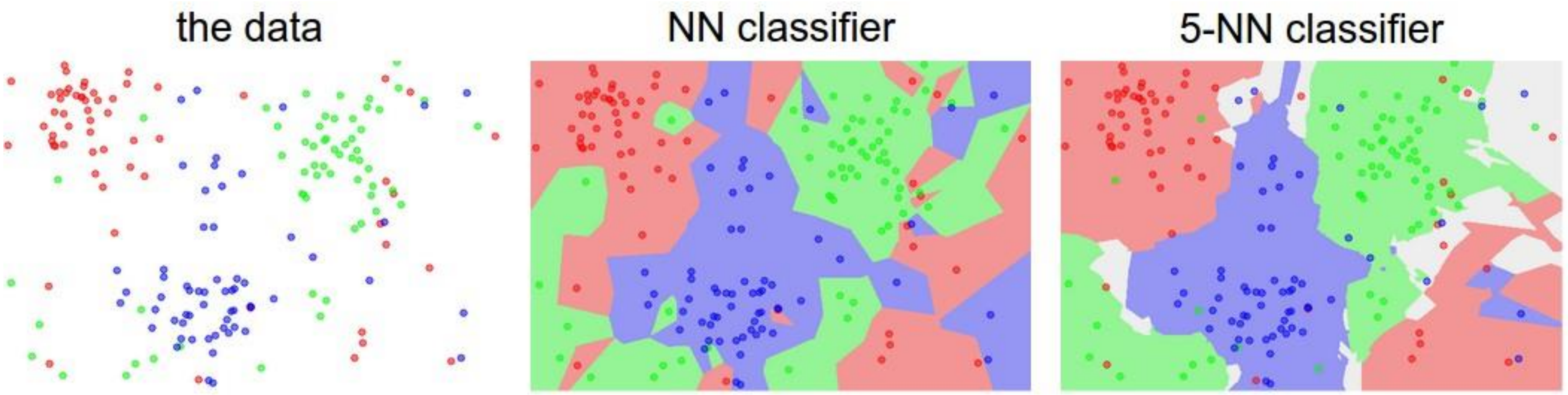
Q: Is this machine learning (ML)?

K-Nearest Neighbour

- Extension of 1-nearest neighbour method
- But find ***k*** nearest instances from the training set
- Then choose the **majority** class as the class label of the unseen instance



- K usually an odd number (why?)
- Results depend on k value
- Need a distance/similarity measure



https://en.wikipedia.org/wiki/Voronoi_diagram

Distance Measure

- Given two feature vectors with numeric values
- $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$

- **Distance measure (Euclidean distance)**
- $d = \sqrt{\sum_{i=1}^{n} \frac{(a_i - b_i)^2}{R_i^2}} = \sqrt{\frac{(a_1 - b_1)^2}{R_1^2} + \frac{(a_2 - b_2)^2}{R_2^2} + \cdots + \frac{(a_n - b_n)^2}{R_n^2}}$

Range of the *i*th feature

- Why divide by the range?

- Easy to use, can achieve good results in many cases
- Efficiency?

Training and Testing

- We usually divide dataset into training and test sets
- Train a classifier using the training set
- Calculate the test performance (test error) on the test/unseen set

- Different training/test division leads to different test performance

- Example:
  - Learning algorithms A and B
  - Dataset: instances 1~100
  - Train on 1~70, test on 71~100: A got 96% accuracy, B got 93% accuracy
  - Train on 31~100, test on 1~30: A got 91% accuracy, B got 95% accuracy
  - Which algorithm is better?

- *K-fold cross validation* makes estimation less division-dependent

K-fold Cross Validation

- Idea: chop the data into K equal subsets
- For each subset:
  - Treat it as the test set
  - Treat the rest K−1 subsets as the training set
  - Train classifier using the training set, apply it to the test set

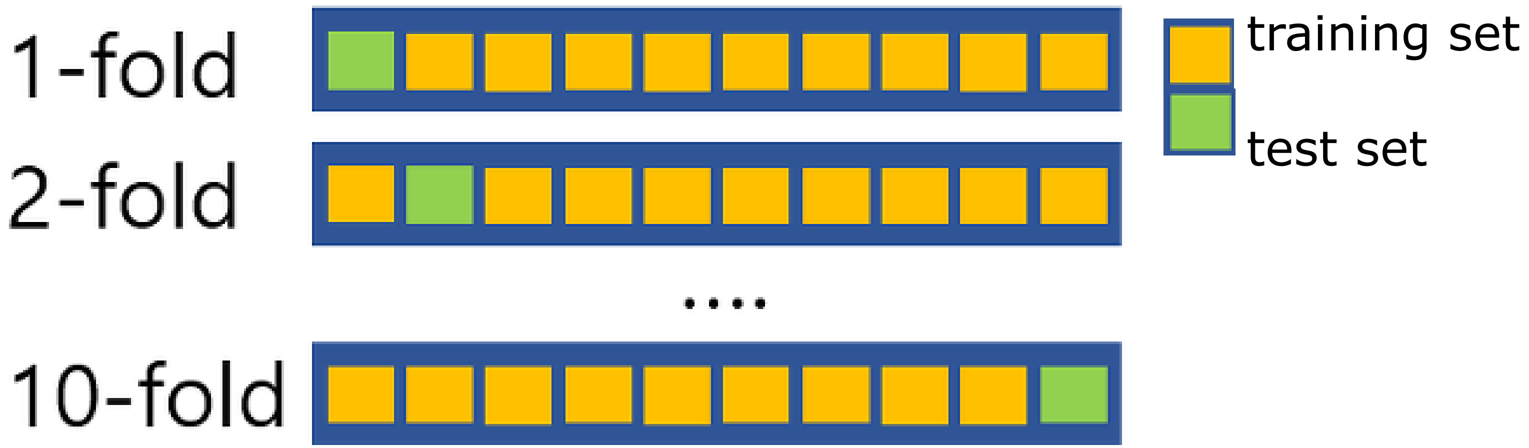
- The training/test process is repeated K times (the folds), with each of the K subsets used exactly once as the test set

- The K results from the folds can be then averaged (or otherwise combined) to produce a single estimation

- Can be used for comparing two algorithms, or to measure the performance of one algorithm when the data set is small

K-fold Cross Validation: Example

- 10-fold cross validation
  - Divide into 10 equally-sized subsets



Leave-one-out Cross Validation

- Very similar to k-fold cross validation
- Every time, it only uses one instance as the test set
- A special case where size of subsets is 1
- Repeated *m* times, where *m* is the number of instances in the entire dataset!

- K-fold cross validation is *an experiment design method* for setting up experiments to test algorithms on **supervised** learning tasks (classification and regression)

- It is **NOT** a ML or classification method/technique.

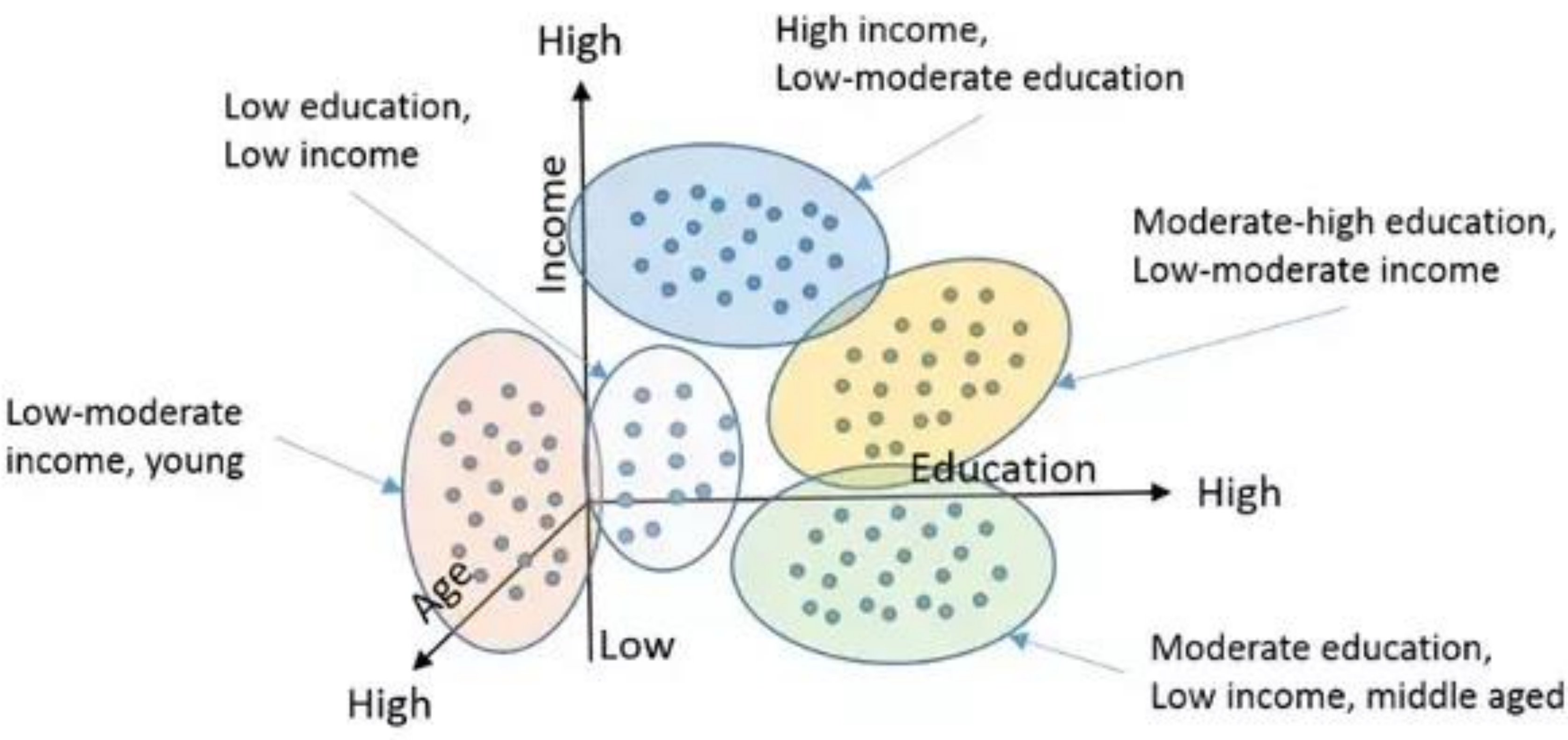
Validation Set vs Cross Validation

- Validation set
  - A pre-defined dataset
  - Separate dataset from the training and test datasets
  - Used for monitoring the training process but not directly used for learning the classifier
  - Helps to avoid overfitting

- Cross validation
  - Experimental design method
  - In this method, there are only training and test datasets
  - No validation set (usually)

K-means Clustering

- Unlabelled data
- Want to obtain a good *partition* for the data by using ML
- Need clustering techniques, type of unsupervised learning



K-means Clustering

- K-means clustering is a method of cluster analysis which aims to partition ***m*** instances into ***k*** clusters in which each instance belongs to the cluster with the nearest **mean**.

- Need a distance measure such as Euclidean distance
- Generally, need to assume the number of clusters

K-means Clustering: Algorithm

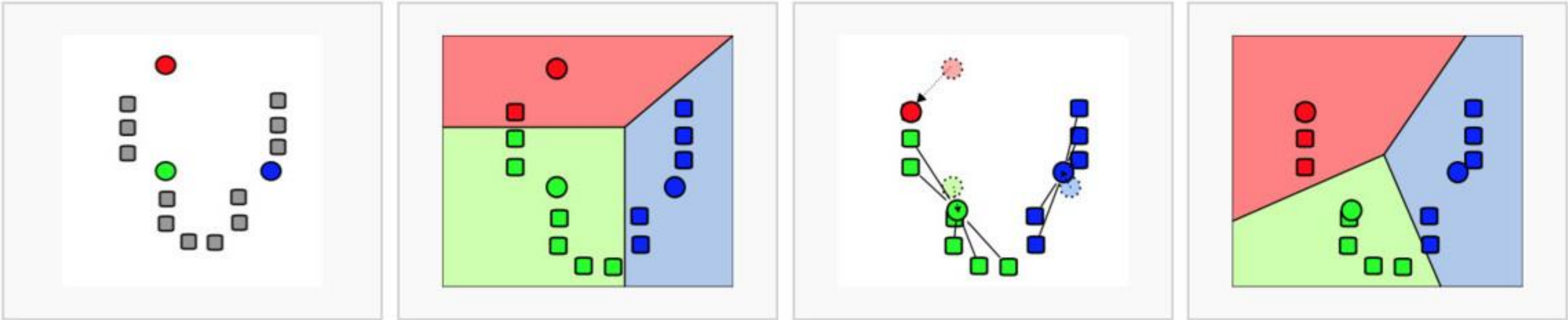
- 1. Initialise k initial "means" randomly from the data set
- 2. Create k clusters by assigning every instance to the nearest cluster: based on the nearest mean according to the distance measure

$$c(x) = \arg\min_{i=1,\dots,k} dist(c_i, x)$$

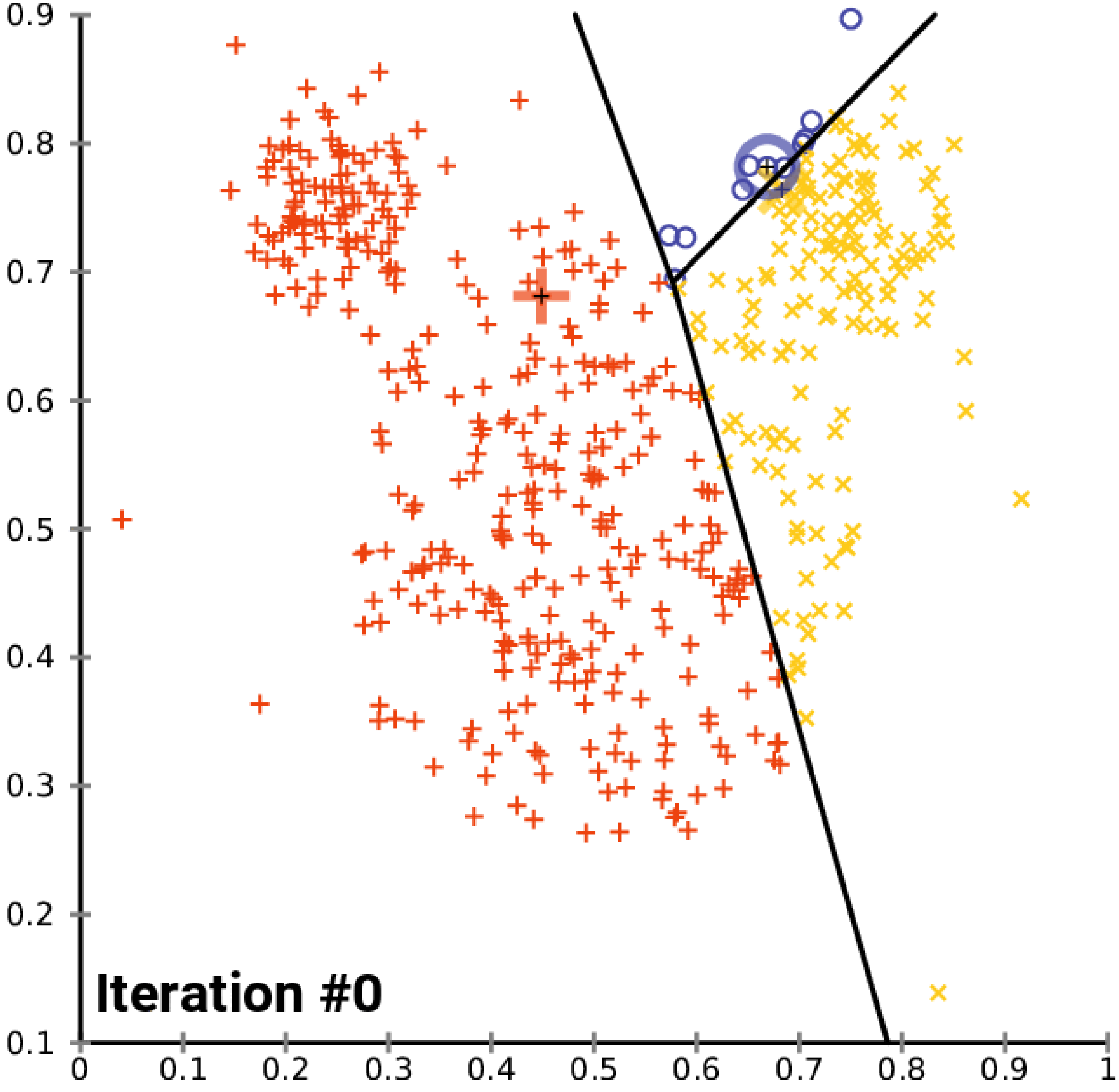
- 3. Replace the old means with the centroid (mean) of each cluster

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i$$

- 4. Repeat the above two steps until convergence (no change in each cluster centroid).
- *Centroid is not an instance*

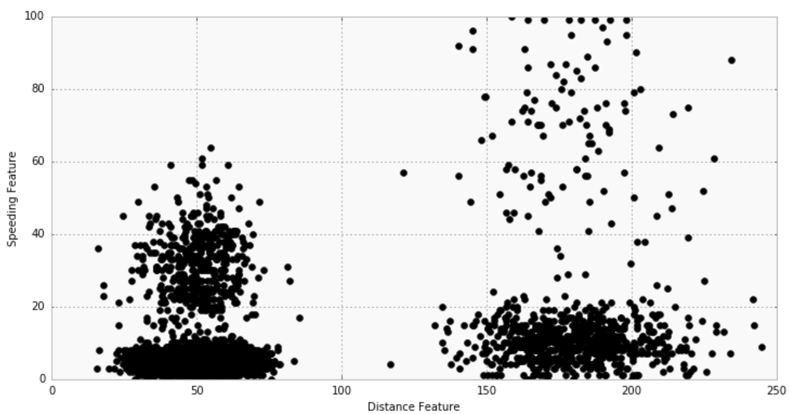


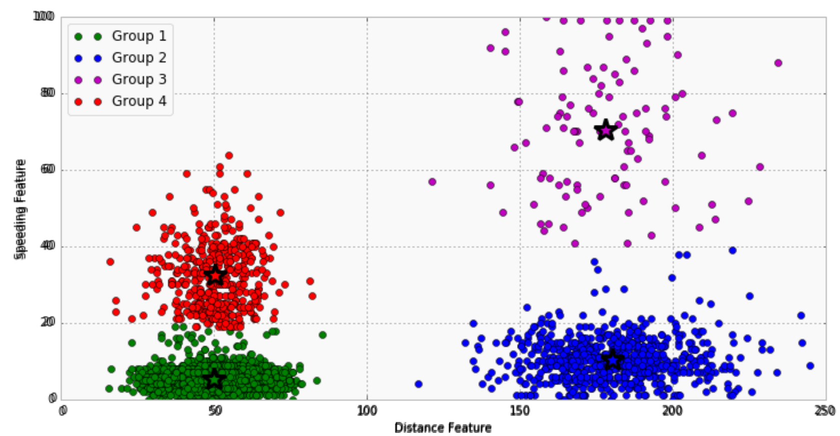
K-Means Clustering: An Example



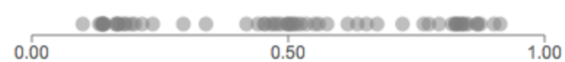
Choosing K

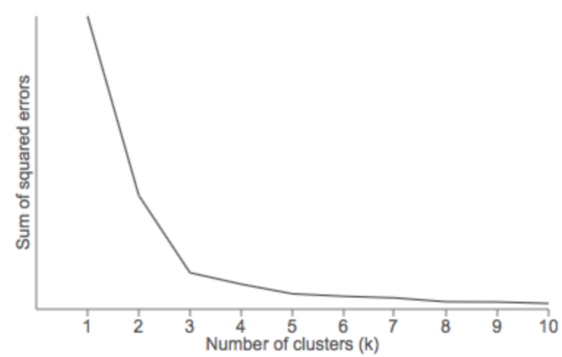
- Number of clusters
- A very important parameter
- But hard to determine in real world (need domain knowledge)





- Many methods to estimate K
  - Elbow method
  - X-mean clustering
  - Silhouette method
  - …





Summary

- Nearest neighbour method for classification
  - K-Nearest neighbour method — classification method
  - Measures of comparing two feature vectors

- K-fold cross validation
  - experimental design method, NOT a ML method
  - validation set is a pre-defined data set

- K-means method – clustering method, NOT for classification

- Next Lecture: Decision tree learning for classification
- Suggested reading: Section 18.3 (both 2nd and 3rd editions) and online materials