



School of Engineering and Computer Science

COMP 307 — Lecture 05

Machine Learning 2

3-K Techniques: K-Nearest Neighbour, K-fold Cross Validation and K-Means Clustering

Mengjie Zhang (Meng)

mengjie.zhang@ecs.vuw.ac.nz

COMP307

ML2 (3-K techniques): 3

Nearest Neighbour

- Given a training set with a number of instances
- Each instance is represented as a simple feature vector and a desired/target class label
- Nearest neighbour method
 - Each unseen instance (in the test set) is compared with all the instances in the training set
 - Find the “nearest neighbour” (instance) from the training set
 - the unseen instance is classified as the class of the nearest neighbour
- K-Nearest Neighbour method:
 - Similar to the 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
- But how do we find the nearest neighbours?

COMP307

ML2 (3-K techniques): 2

Outline

- Nearest neighbour method
 - Basic nearest neighbour method
 - K-Neighbour method
 - Distance measure/Similarity measure
- K-fold cross validation
 - Leave-one out cross validation
 - k-fold cross validation vs validation set
- K-means clustering

COMP307

ML2 (3-K techniques): 4

Nearest Neighbour — Distance Measures

- Given two feature vectors with numeric values

$$A = (a_1, a_2, \dots, a_n) \text{ and } B = (b_1, b_2, \dots, b_n)$$

- Use the *distance measure*:

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

R_i is the *range* of the i th component

- The (k-)nearest neighbour method is simple, easy to use, and can achieve good results in many cases
- What problem can you find?
 - Does this method explicitly learn a classifier? If yes, what is it?
 - Efficient?

K-fold Cross-Validation

- Idea: chop the available data into K equal chunks
- For each chunk in turn:
 - Treat it as the *test* data set
 - Treat the rest $K - 1$ chunks as the *training* data set
 - The classifier trained/learned from the training set is applied to the test set
- The process is then repeated K times (the folds), with each of the K chunks used exactly once as the test data 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 measure the performance of a particular algorithm *when the data set is small*.

Cross-validation vs Validation Set

- Validation set (vs training set vs training set)
 - The validation set is a *data set*
 - Validation set is a separate data set from the training set and the test set.
 - Validation set is used for monitoring the training process but is not directly used for learning the classifier
 - the *validation set* is used for avoid *overfitting* or *overtraining*
 - Assume: 100 examples
 - 50 vs 50
 - 40 vs 30 vs 30
- A figure

Leave-one-out Cross Validation

- It is very similar to the K -fold cross-validation method
- Every time, it only uses *one instance* as the test data set
- The process needs to be repeated n times, where n is the number of total examples/instances in the entire data set
- K -fold cross validation (including leave-one-out cross validation) is NOT a machine learning or classification method or technique
- It is an *experimental design method* for setting up experiments for supervised learning tasks such as classification and regression

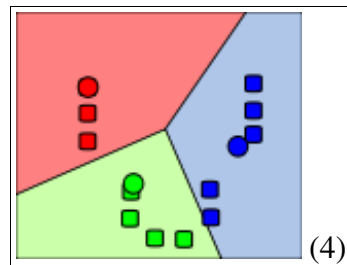
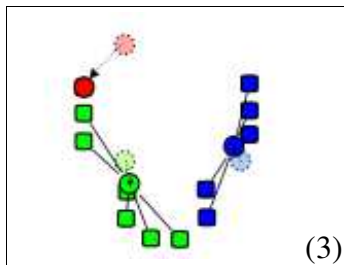
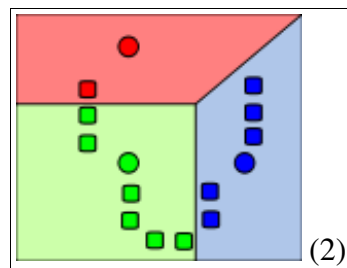
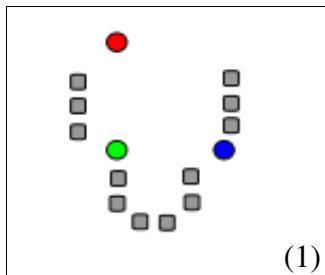
Cross-validation vs Validation Set

- Cross Validation
 - Is a *experimental design method*, NOT a *data set*
 - In this method, there are only training sets and test sets
 - **No validation set** exists
 - Example: 100 instances for 5-fold cross validation
 - Splitting into 5 chunks 20, 20, 20, 20, 20
 - first 20 are used as the test set, other 80 as the training set
 - second 20 are used as the test set, other 80 as the training set
 - third 20 are used as the test set, other 80 as the training set
 - fourth 20 are used as the test set, other 80 as the training set
 - fifth 20 are used as the test set, other 80 as the training set
 - Final performance: average on the five test set
- A figure

K-Means Clustering

- In many situations, we only have the instances that are not labelled.
- We do not know their class labels in advance but expect to obtain good partitions for the instances using learning techniques.
- Need clustering techniques
- k-means clustering is a method of cluster analysis which aims to partition n instances into k clusters in which each instance belongs to the cluster with the nearest mean.
- Need some kind of distance measure such as Euclidean distance
- Need to assume the number of clusters

K-Means Clustering: An Example



K-Means Clustering: Algorithm

1. Set k initial "means" (in this case $k=3$) randomly from the data set (shown in color).
2. Create k clusters by associating every instance with the nearest mean based on a distance measure.
3. Replace the old means with the centroid of each of the k clusters (as the new means).
4. Repeat the above two steps until convergence.

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 learning method*
 - *validation set is a data set, NOT a method*
- 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