University of Canberra

Faculty of Science and Technology

**Programming for Data Science G (11521)**

**Week 9 Tutorial**

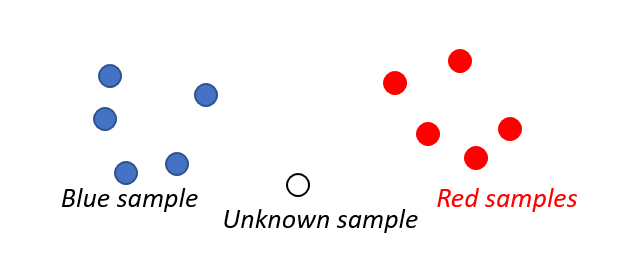
**Vector Quantisation and Nearest Centroid Classifier (or Nearest Prototype Classifier)**

**Objectives**

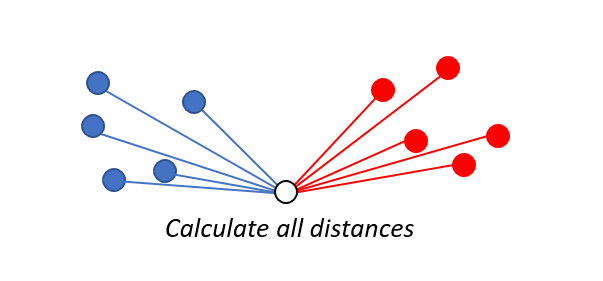
* To use K-Means Clustering to build Vector Quantisation and combine with Nearest Neighbour Classifier to build Nearest Centroid Classifier (or Nearest Prototype Classifier)

**Vector Quantisation Classifier**

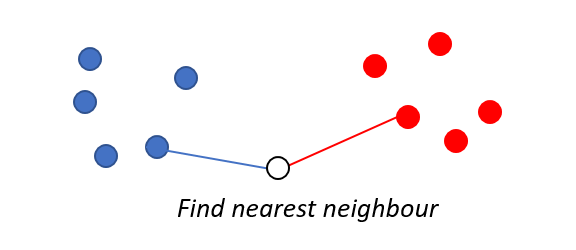
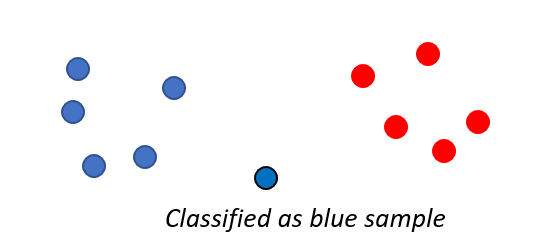
* Assume that you have two data samples red and blue, and an unknown sample as follows



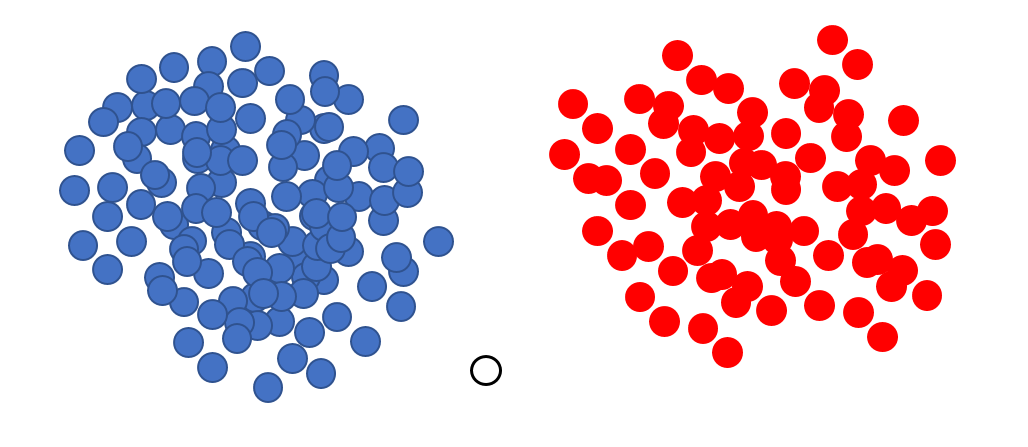
* To classify the unknown sample using nearest neighbour classifier, you calculate all distances from the unknown sample to all samples.



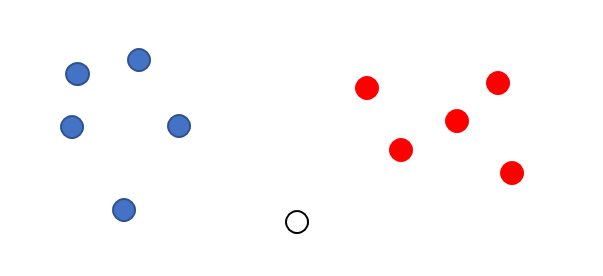
* Then find nearest sample. In this example, the unknown sample is classified as blue sample.

* If data sets are large, it will take plenty of time for calculation of distances. To reduce the calculation time, the red data set and blue data set will be replaced with their cluster centre set. As the numbers of cluster centres are small, the calculation time is short.



Data samples are replaced by cluster centres



Apply nearest neighbour classifier on unknown sample and cluster centres.

This data compression technique is called **Vector Quantisation** and the classifier is called **Nearest Centroid Classifier** or **Nearest Prototype Classifier**.

A cluster centre dataset is called **codebook** and a cluster centre is called **codevector**.

**Your task: Implement the following program**

Repeat the following steps 3 times with K = 2, 3, and 4:

* Run **K-means Clustering** program to cluster the given blue and red datasets in Assignment 1 and output their cluster centres to file as follows
  + blue\_2d.txt 🡪 K-means clustering 🡪 blue\_model\_2d.txt
  + red\_2d.txt 🡪 K-means clustering 🡪 red\_model\_2d.txt
  + blue\_4d.txt 🡪 K-means clustering 🡪 blue\_model\_4d.txt
  + red\_4d.txt 🡪 K-means clustering 🡪 red\_model\_4d.txt
  + blue\_8d.txt 🡪 K-means clustering 🡪 blue\_model\_8d.txt
  + red\_8d.txt 🡪 K-means clustering 🡪 red\_model\_8d.txt
* Run **Nearest Neighbour Classifier** program on the given unknown datasets and the model datasets output from the K-Means Clustering program as follows
  + blue\_model\_2d.txt, red\_model\_2d.txt, and unknown\_2d.txt 🡪 Nearest Neighbour Classifier 🡪 results
  + blue\_model\_4d.txt, red\_model\_4d.txt, and unknown\_4d.txt 🡪 Nearest Neighbour Classifier 🡪 results
  + blue\_model\_8d.txt, red\_model\_8d.txt, and unknown\_8d.txt 🡪 Nearest Neighbour Classifier 🡪 results
* Compare the above results with the results you got from the red, blue, and unknown datasets for Assignment 1.

**Hint**: download the **io\_data\_module.py** and use the **run\_kmeans** and **run\_nearest\_neighbour** functions to implement your program. Below is a sample code for using these functions

#Main program

blue\_files = ['datasets/blue\_2d.txt', 'datasets/blue\_4d.txt', 'datasets/blue\_8d.txt']

red\_files = ['datasets/red\_2d.txt', 'datasets/red\_4d.txt', 'datasets/red\_8d.txt']

unknown\_files = ['datasets/unknown\_2d.txt', 'datasets/unknown\_4d.txt', 'datasets/unknown\_8d.txt']

K = 2

index = 0

blue\_model\_list = iodata.run\_kmeans(K, blue\_files[index])

print(blue\_model\_list)

print()

result = iodata.run\_nearest\_neighbour(blue\_files[index], red\_files[index], unknown\_files[index])

print(result)

**Output**:

[[-0.7839650277777778, -0.7586059444444447], [0.4957978888888888, 0.49544179629629626]]

0.678713 0.951598 blue

-1.169512 -0.957855 blue

0.631947 1.236489 blue

-0.131799 -0.324218 blue

-0.199565 -0.229171 blue

0.900907 0.448131 blue

0.699395 0.414314 blue

-1.132646 -1.013298 blue

1.004178 1.355361 blue

-0.127378 -0.097121 blue

6.08804 3.457729 red

4.147974 5.275341 red

6.538759 3.670323 red

4.579573 4.03559 red

4.756026 4.184762 red

5.221742 2.872705 red

5.271773 3.158064 red

4.046376 5.19232 red

6.530952 3.171413 red

4.918007 4.142507 red