

# EE2211 Pre-Tutorial 11

Dr Feng LIN

feng\_lin@nus.edu.sg



# Agenda

- Recap
- Self-learning
- Tutorial 11



# Recap

- Introduction of unsupervised learning
- K-means Clustering
  - The most popular clustering technique
- Fuzzy Clustering

# Unsupervised Learning

## Introduction

**Motivation:** we do not always have labeled data.

In **unsupervised learning**, the dataset is a collection of **unlabeled examples**  $\{\mathbf{x}_i\}_{i=1}^M$ .

Evaluation of unsupervised learning is hard:

- The **absence of labels** representing the desired behavior for your model means **the absence of a solid reference point to judge the quality of your model**.

# Unsupervised Learning

## Main Tasks/Approaches

- **Clustering**
  - ✓ Groups a set of objects in such a way that objects in the same group (called a **cluster**) are **more similar** (in some sense) to each other than to those in other groups (clusters).
- **Density Estimation**
  - ✓ Models the probability density function (pdf) of the unknown probability distribution from which the dataset has been drawn.
- **Component Analysis**
  - ✓ Breaks down the data from the perspective of signal analysis.
- **Unsupervised Neural Networks**
  - ✓ Autoencoder



# K-Means Clustering

Distance based grouping method

- Feature vectors that are closed to each other

K-means

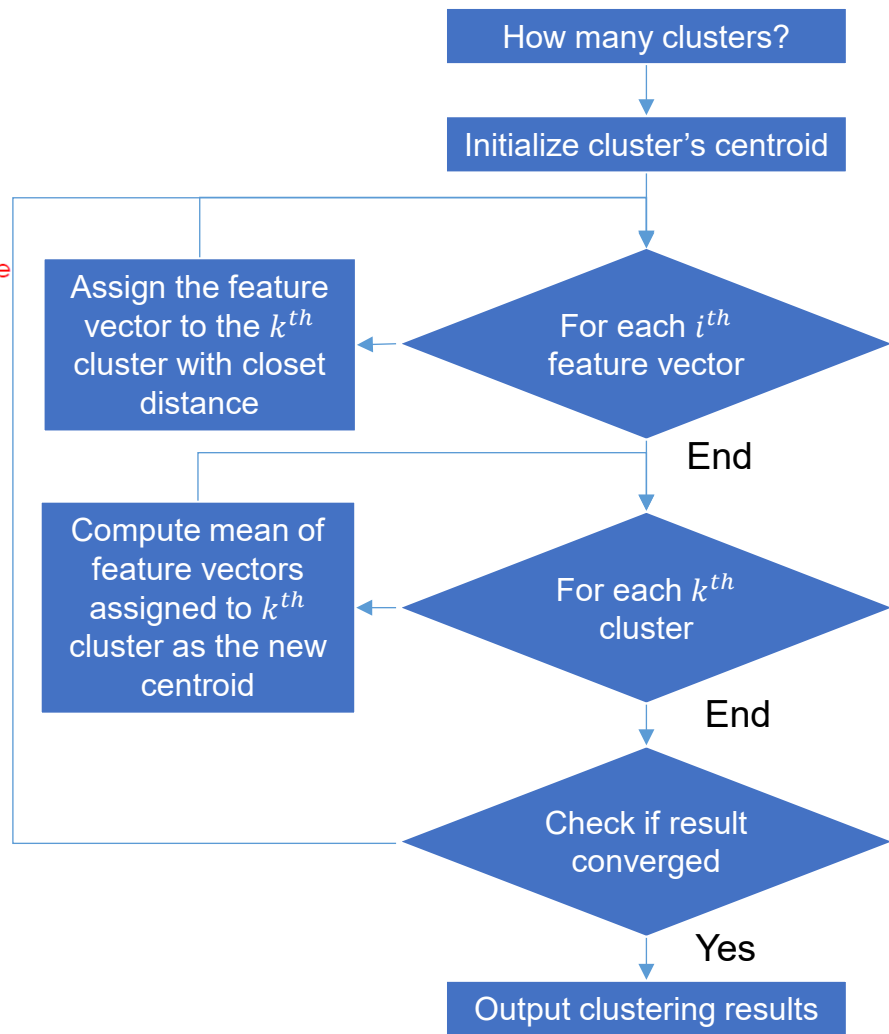
- $K$  = number of clusters
- Means = average or centroid is the average of feature vectors

# K-means Clustering

## Basic/Naïve K-means Clustering

Looping between  
Assignment and Centroid Update

1. First, we choose  $K$  — the number of clusters. Then we randomly select  $K$  feature vectors, called **centroids**, to the feature space.
2. Next, compute the distance from each example  $x$  to each centroid  $c$  using some metric, like the Euclidean distance. Then we assign the closest centroid to each example (like if we labeled each example with a centroid id as the label).
3. For each centroid, we calculate the average feature vector of the examples labeled with it. These average feature vectors become the new locations of the centroids.
4. We recompute the distance from each example to each centroid, modify the assignment and repeat the procedure until the assignments don't change after the centroid locations are recomputed.
5. Finally, we conclude the clustering with a list of assignments of centroids IDs to the examples.



# K-means Clustering

## Optimization Objective Function (within-cluster variance)

Minimize  $J$

$m$ : # of samples;  $i$ : index of samples

$K$ : # of clusters;  $k$ : index of clusters

$$J = \sum_{i=1}^m \sum_{k=1}^K w_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2 \quad (1)$$

The term  $w_{ik}$  is equal to 1 for data point  $\mathbf{x}_i$  if the data point belongs to cluster  $S_k$ , else  $w_{ik} = 0$ .

Note: The optimization objective function was called  $C(\mathbf{w})$  in Lecture 8. Here, we use  $J$  (with parameters  $w_{ik}$  and  $\mathbf{c}_k$ ) so that it is differentiated from the centroids  $\mathbf{c}_k$ .



# K-means Clustering

## Naïve K-means Algorithm

### 1. Assignment Step (fix $\mathbf{c}$ and update $w$ ):

Computing distances to  
all centroids

$$\mathbf{x}_i \in S_k \ (w_{ik} = 1) \text{ if } \|\mathbf{x}_i - \mathbf{c}_k\|^2 < \|\mathbf{x}_i - \mathbf{c}_j\|^2 \text{ (else } w_{ik} = 0), \\ i = 1, \dots, m; \ j, k = 1, \dots, K.$$

### 2. Update Step (fix $w$ and update $\mathbf{c}$ ):

$$\frac{\partial J}{\partial \mathbf{c}_k} = -2 \sum_{i=1}^m w_{ik} (\mathbf{x}_i - \mathbf{c}_k) = 0 \Rightarrow \mathbf{c}_k = \frac{\sum_{i=1}^m w_{ik} \mathbf{x}_i}{\sum_{i=1}^m w_{ik}}$$

Solving an optimization, i.e., setting derivative to 0

Note:  $\|\mathbf{x} - \mathbf{c}\| = \sqrt{\sum_{d=1}^D (x_d - c_d)^2}$  is called the Euclidean distance.

where  $\mathbf{x} = (x_1, x_2, \dots, x_D)$ ,  $\mathbf{c} = (c_1, c_2, \dots, c_D)$

# K-means Clustering

## 1. Assignment Step (fix $\mathbf{c}$ and update $w$ ):

$$\mathbf{x}_i \in S_k \text{ (} w_{ik} = 1 \text{) if } \|\mathbf{x}_i - \mathbf{c}_k\|^2 < \|\mathbf{x}_i - \mathbf{c}_j\|^2 \text{ (else } w_{ik} = 0), \\ i = 1, \dots, m; \quad j, k = 1, \dots, K.$$

## 2. Update Step (fix $w$ and update $\mathbf{c}$ ):

$$\frac{\partial J}{\partial \mathbf{c}_k} = -2 \sum_{i=1}^m w_{ik} (\mathbf{x}_i - \mathbf{c}_k) = 0 \Rightarrow \mathbf{c}_k = \frac{\sum_{i=1}^m w_{ik} \mathbf{x}_i}{\sum_{i=1}^m w_{ik}}$$

By repeating these two steps, the total loss  $J = \sum_{i=1}^m \sum_{k=1}^K w_{ik} \|\mathbf{x}_i - \mathbf{c}_k\|^2$ , is **guaranteed to NOT increase (i.e., remain the same or decrease)** until convergence.

Why? **At Step 2:** we compute the new mean, by solving an optimization, i.e., compute the derivative and set to zero, and solve  $\mathbf{c}_k$ . This means that, the new  $\mathbf{c}_k$  is guaranteed to give a smaller  $J$  value.

**At Step 1:** we only change the assignment, if the distance to the new centroid is smaller! In other words, we either remain in the old group, or change to a new group that is closer (i.e., gives a smaller  $J$ )

# An Example of K-means

Consider the following unlabelled one-dimensional dataset (so that the samples are all scalar)

$$x_1 = -2, \quad x_2 = 0, \quad x_3 = x_4 = 2.$$

Consider the first initialization

$$c_1^{(1)} = -3, \quad c_2^{(1)} = 3.5$$

Then, once we run the Assignment step, we see that

$$k_1 = k_2 = 1, \quad k_3 = k_4 = 2.$$

This means that samples 1 and 2 are in group one and samples 3 and 4 are in group two. Thus,

$$c_1^{(2)} = -1, \quad c_2^{(2)} = 2.$$

The total cost function is

$$J = 1^2 + 1^2 + 0^2 + 0^2 = 2,$$

which turns out to be the optimum partitioning.

# An Example of K-means

Now, instead consider the second initialization

$$c_1^{(1)} = -3, \quad c_2^{(1)} = 2.5$$

Then, once we run the Assignment step, we see that

$$k_1 = 1, \quad k_2 = k_3 = k_4 = 2.$$

Thus,

$$c_1^{(2)} = -2, \quad c_2^{(2)} = 4/3$$

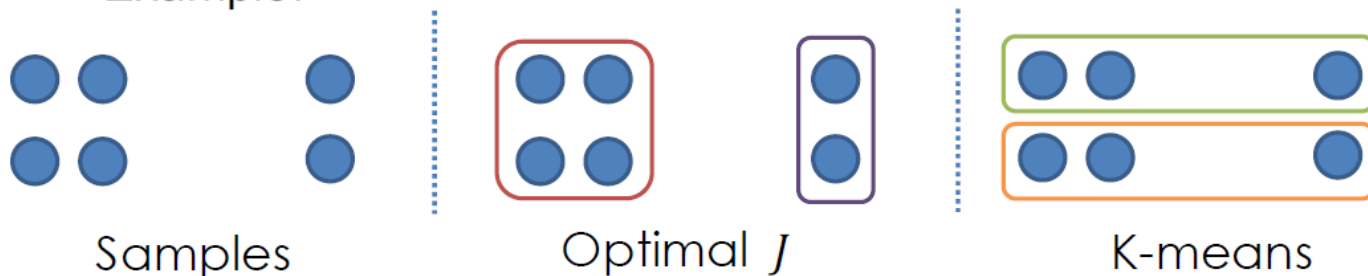
The total cost function is

$$J' = 0^2 + (4/3)^2 + 2(2 - 4/3)^2 = 24/9$$

which is suboptimal and there is no way of improving the cost anymore, i.e., we are stuck. The moral of the story is that initialization is important.

# K-means Clustering

- Unfortunately, k-means is not guaranteed to find a global minimum, it finds only local minimum.
- Example:



- Finding the optimal  $J$  is NP-hard\*
- In practice, k-means clustering usually performs well
- It can be very efficient, and its solution can be used as a starting point for other clustering algorithms

# K-means Clustering

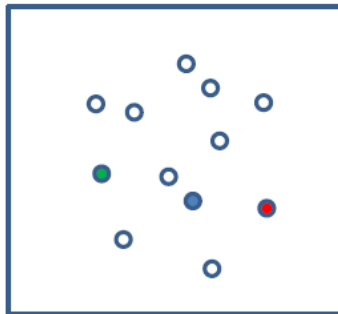
- Initialization

Initialization by centroid

**Forgy method:**

- Randomly chooses  $k$  observations from the dataset and uses these as the initial means.

$k=3$

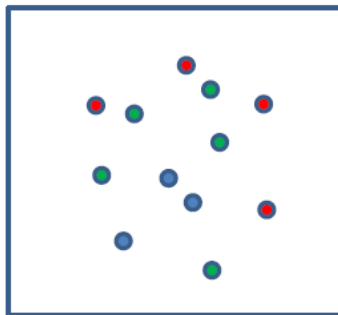


**Random partition:**

- First randomly assigns a cluster to each observation and then proceeds to the update step, thus computing the initial mean to be the centroid of the cluster's randomly assigned points

Initialization by grouping

$k=3$



# Hard vs Soft Clustering

## Hard clustering:

Each data point can belong only one cluster, e.g. K-means

- For example, an apple can be red **OR** green (hard clustering)

## Soft clustering (also known as **Fuzzy clustering**):

Each data point can belong to more than one cluster.

- For example, an apple can be red **AND** green (fuzzy clustering)
- Here, the apple can be red to a certain degree as well as green to a certain degree.
- Instead of the apple belonging to green [green = 1] and not red [red = 0], the apple can belong to green [green = 0.3] and red [red = 0.5]. These value are normalized between 0 and 1; however, they do not represent probabilities, so the two values **do not need to add up to 1**.

# Hard vs Soft Clustering

## Objective Function for Fuzzy C-means

Minimize  $J$

$$J = \sum_{i=1}^m \sum_{k=1}^C (w_{ik})^r \|\mathbf{x}_i - \mathbf{c}_k\|^2$$

$$\text{where } w_{ik} = \frac{1}{\sum_{j=1}^C \left( \frac{\|\mathbf{x}_i - \mathbf{c}_k\|}{\|\mathbf{x}_i - \mathbf{c}_j\|} \right)^{\frac{2}{r-1}}}$$

**No need to memorize  
the equation!**

Each element,  $w_{ik} \in [0,1]$ , tells the degree to which element,  $\mathbf{x}_i$ , belongs to cluster  $\mathbf{c}_k$ .

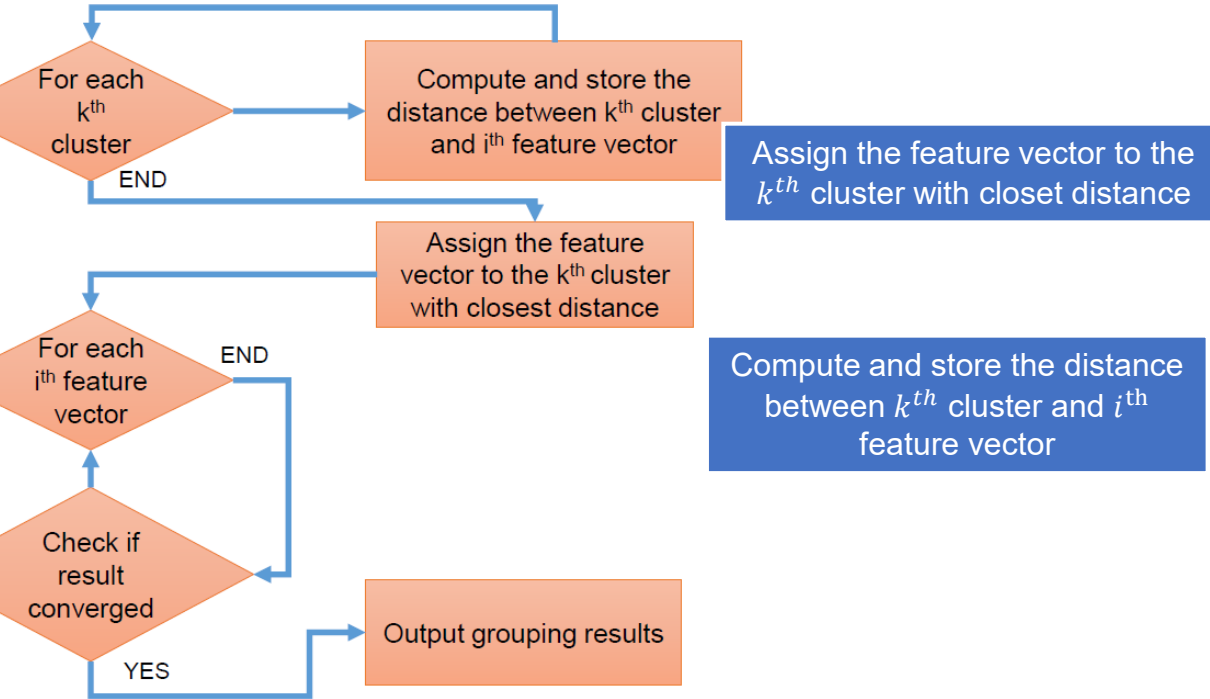
The fuzzifier  $r > 1$  determines the level of cluster fuzziness; usually  $1.25 \leq r \leq 2$ .





**THANK YOU**

# K-means Clustering



How many clusters?

Initialize cluster's centroid

For each  $i^{\text{th}}$   
feature vector

For each  $k^{\text{th}}$   
cluster

Check if result  
converged