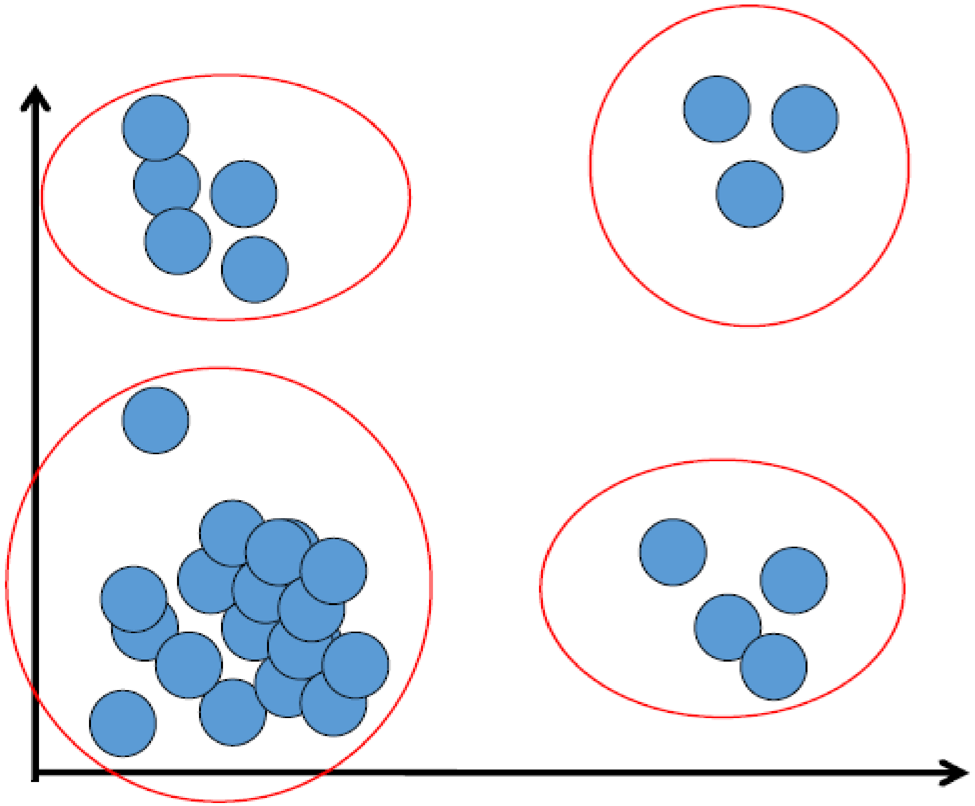
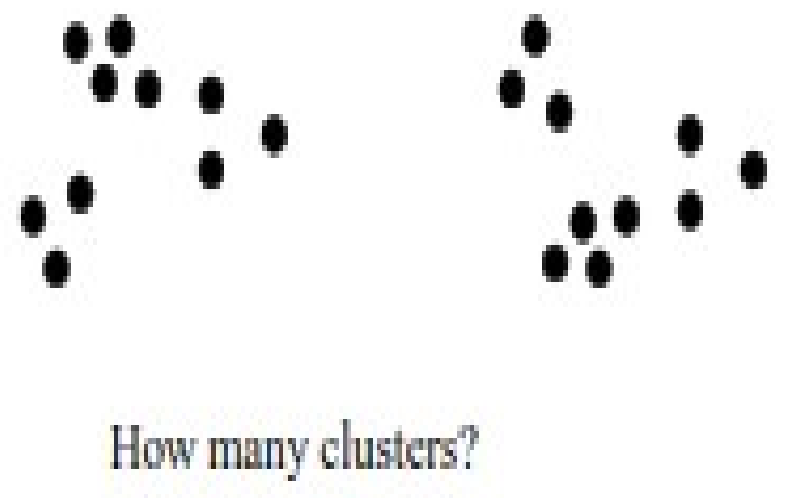
Clustering

Overview of Clustering



- The goal of clustering is to
  - group data points that are close (or **similar**) to each other
  - identify such groupings (or clusters) in an **unsupervised** manner

Observation Exercise on Grouping



- How many groups you could have from this data points

Possible Solutions

- Possible Clusters

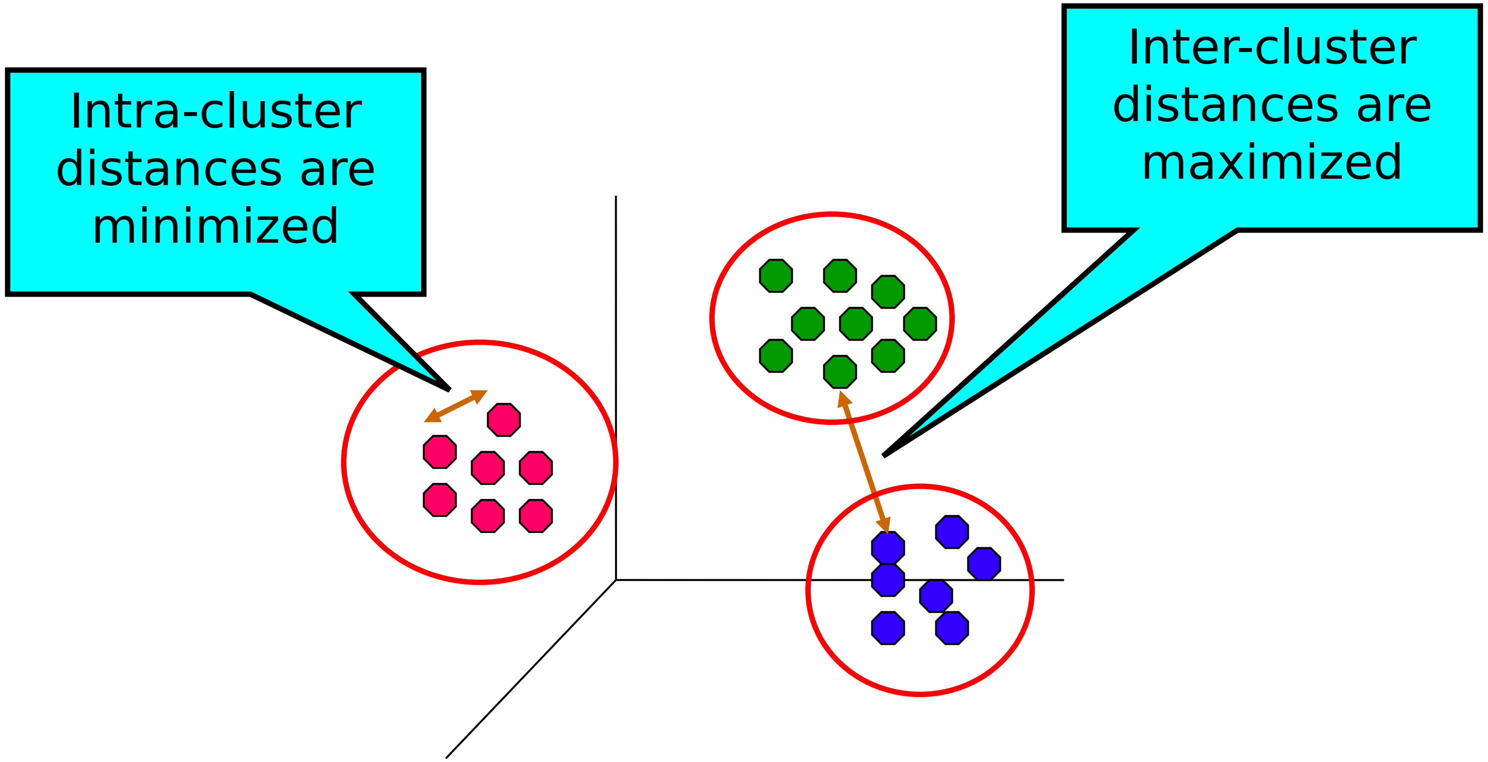
Six Clusters

Two Clusters

Four Clusters

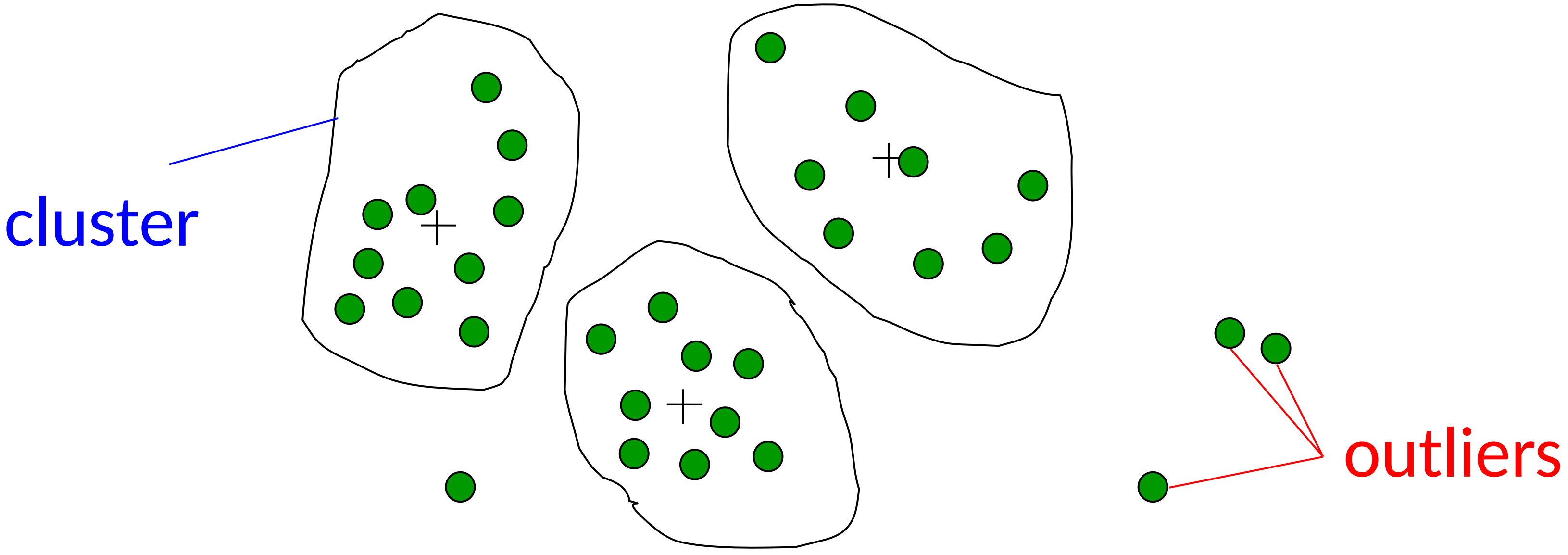
What is clustering?

- A **grouping** of data objects such that the objects **within a group are similar** (or related) to one another **and different from (or unrelated to) the objects in other groups**



Outliers

- **Outliers** are **objects that do not belong to any cluster** or form clusters of very small cardinality



- In some applications we are interested in discovering outliers, not clusters (outlier analysis)

K-means

- Figure 4-1 illustrates three clusters of objects with two attributes. Each object in the dataset is represented by a small dot color-coded to the closest large dot, the mean of the cluster.

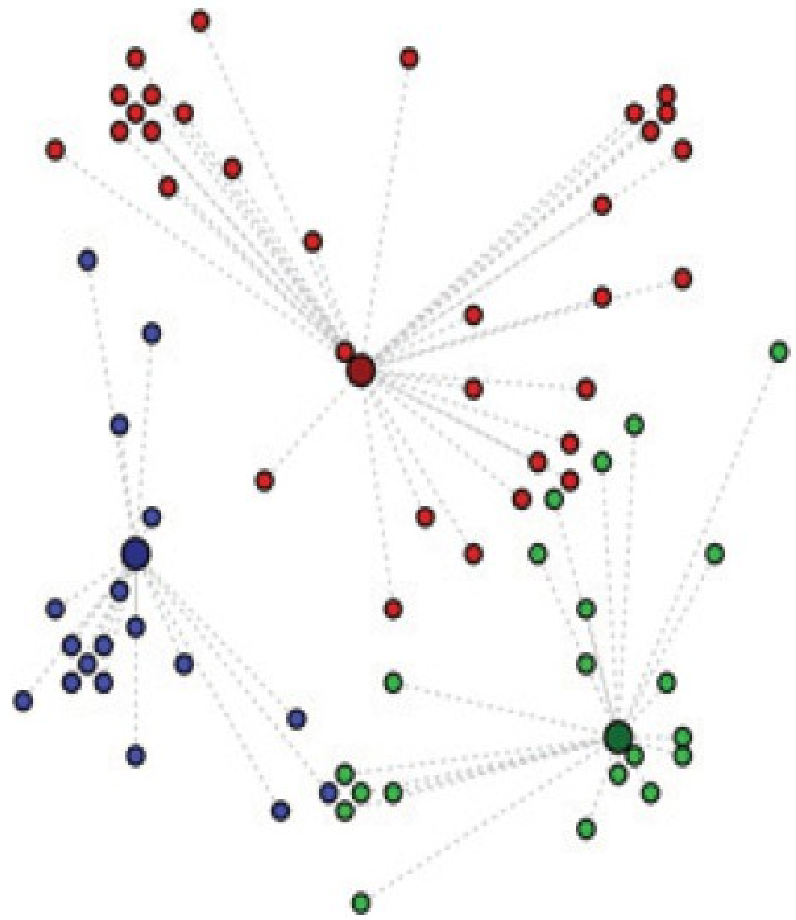


FIGURE 4-1 *Possible k-means clusters for k=3*

Overview of Clustering

- Clustering is often used as a lead-in to classification,
- Clustering is an unsupervised descriptive analytical technique for grouping similar object
  - Unsupervised means that the data scientist does not determine, in advance, the labels to apply to the cluster
  - Once clusters are identified, labels can be applied to each cluster to classify each group based on its characteristics,
- The structure of the data describes the objects of interest and determine how best to group objects,
- For example we can group employees based on their income:
  - Grade 5 earns 15,000 AED a month
  - Grade 4 earns 20,000 AED a month
  - Grade 3 earns 30,000 AED a month, etc.

More on Clustering…

- Clustering is a method often used for exploratory analysis of data
- There are no prediction made in clustering, rather it find similarities between objects according to their attributes and group the similar objects into a cluster.
- Clustering techniques are utilized in marketing, economics, and various branches of science
- A popular clustering method is called: k-means
    - Note: K stands for the number of cluster to create,

Some Definition on K-means Algorithm

- Given a collection of objects each with n measurable attribute, k-means is an analytical technique that, for a chosen k, identify k clusters/groups based on the object proximate to the center (or Centroid) of the K group
- k-means Algorithm aims to partition *n* observations into k clusters (groups) in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster

Use-Cases of Clustering

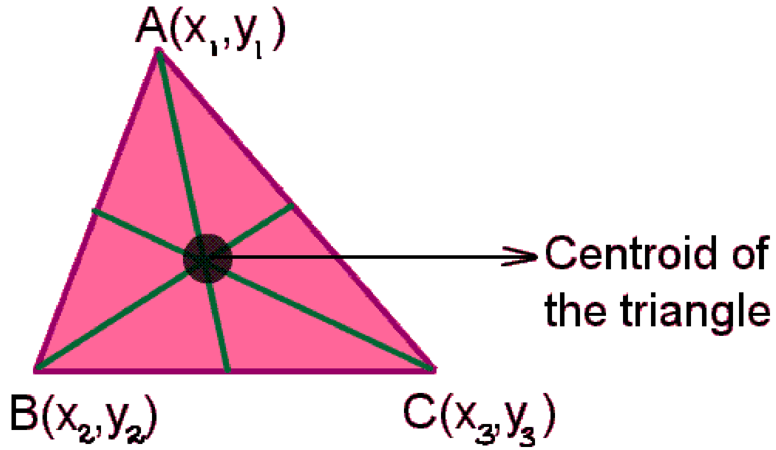
- Use cases of K-means Clustering
  - ***Image Processing***: K-means can identify images that may indicate unauthorized access to facility,
  - ***Medical:*** K-means clustering use patients' profile to identify group of patients that need preventive measures or clinical trials,
  - ***Customer Segmentation***: Marketing and sales groups use k-means to better understand customers who have similar behaviors and spending patters

Use-Cases of Clustering

- **Example 1**: groups people of similar sizes together to make “small”, “medium” and “large” T-Shirts.
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit all.
- **Example 2**: In marketing, segment customers according to their similarities
  - To do targeted marketing.
- **Example 3**: Given a collection of text documents, we want to organize them according to their content similarities,
  - To produce a topic hierarchy

What is Centroid?

- **Centroid definition**: the center of mass of an object of uniform density
- In mathematics and physics, the **centroid** or geometric center of a plane figure is the arithmetic **mean** position of all the points in the shape.
- In n-dimensional space: its **centroid** is the **mean** position of all the points in all of the coordinate directions.



K-means Algorithm

- Given *k*, the *k-means* algorithm works as follows:
  1. Choose *k* (random) data points (seeds) to be the initial centroids, cluster centers
  2. Assign each data point to the closest centroid
  3. Re-compute the centroids using the current cluster memberships
  4. If a convergence criterion is not met, repeat steps 2 and 3

Step 1 of K-Mean Algorithm

Step 1: Choose the value of k and the K initial guess for the centroid

In this example we are using k=3 and the initial 3 centroids (C_1, C_2, C_3) are indicated by the points shaded in red., green and blue

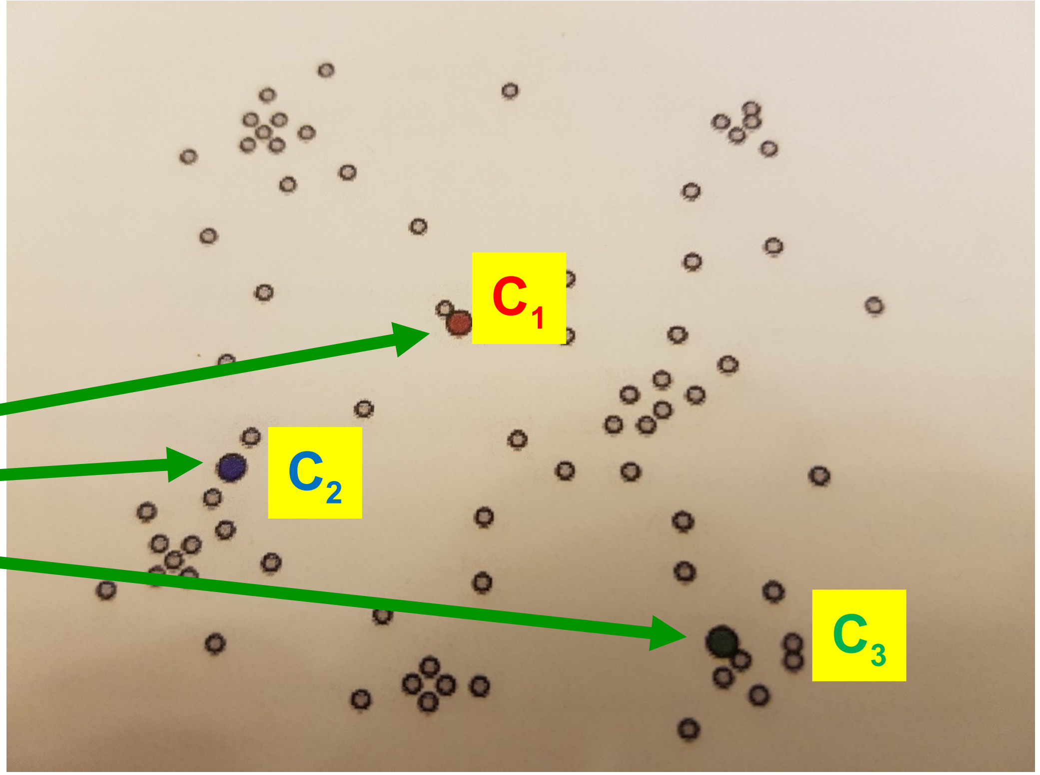


Figure #1: Initial starting points for the 3 centroids (C_1, C_2, C_3)

Step 2 of K-Means Algorithm

Step 2:

- Compute the distance from each data point (x , y) to each centroid.
- Assign each point to the closest centroid.
- This association defines the first K clusters

Note: In two dimension the distance *d* of two points (x_1,y_1) and (x_2, y_2) is

$$d=SQRT((x_1-x_2)^2+(y_1-y_2)^2)$$

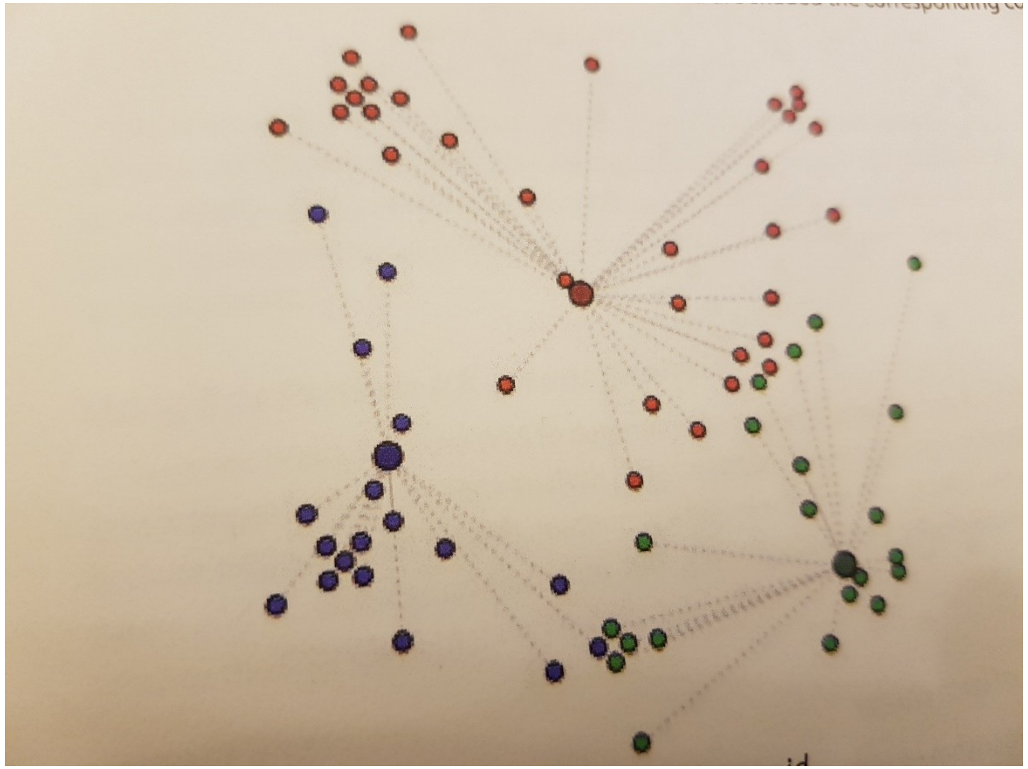


Figure #2: Points assigned the same color of the closest centroid

Step 3 & 4 of K-Means Algorithm

Step 3: Compute the centroid i.e. the center of mass of each newly defined cluster from Step 2

Step 4: Repeat Step 2 & 3 until the algorithm converge to an answer

a) Assign each point to the closest centroid computed in Step 3
b) Compute the centroid of newly defined clusters
c) Repeat the Algorithm until it reaches the final answer

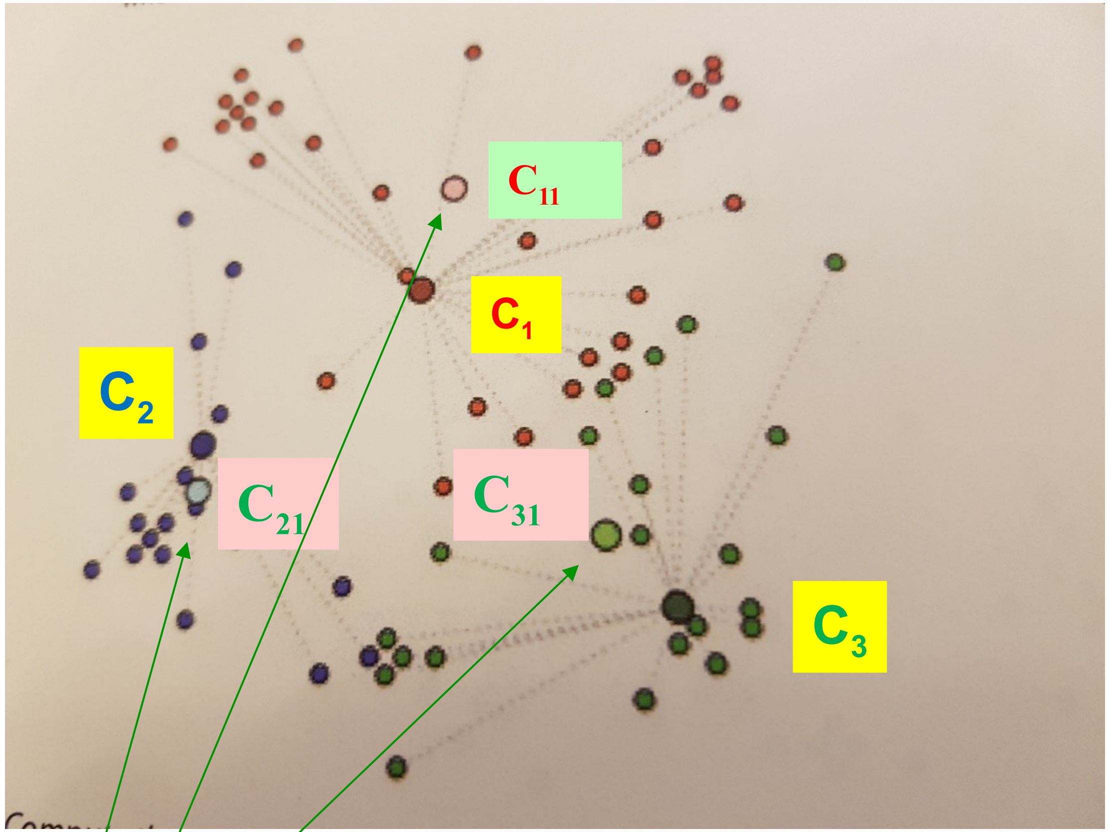
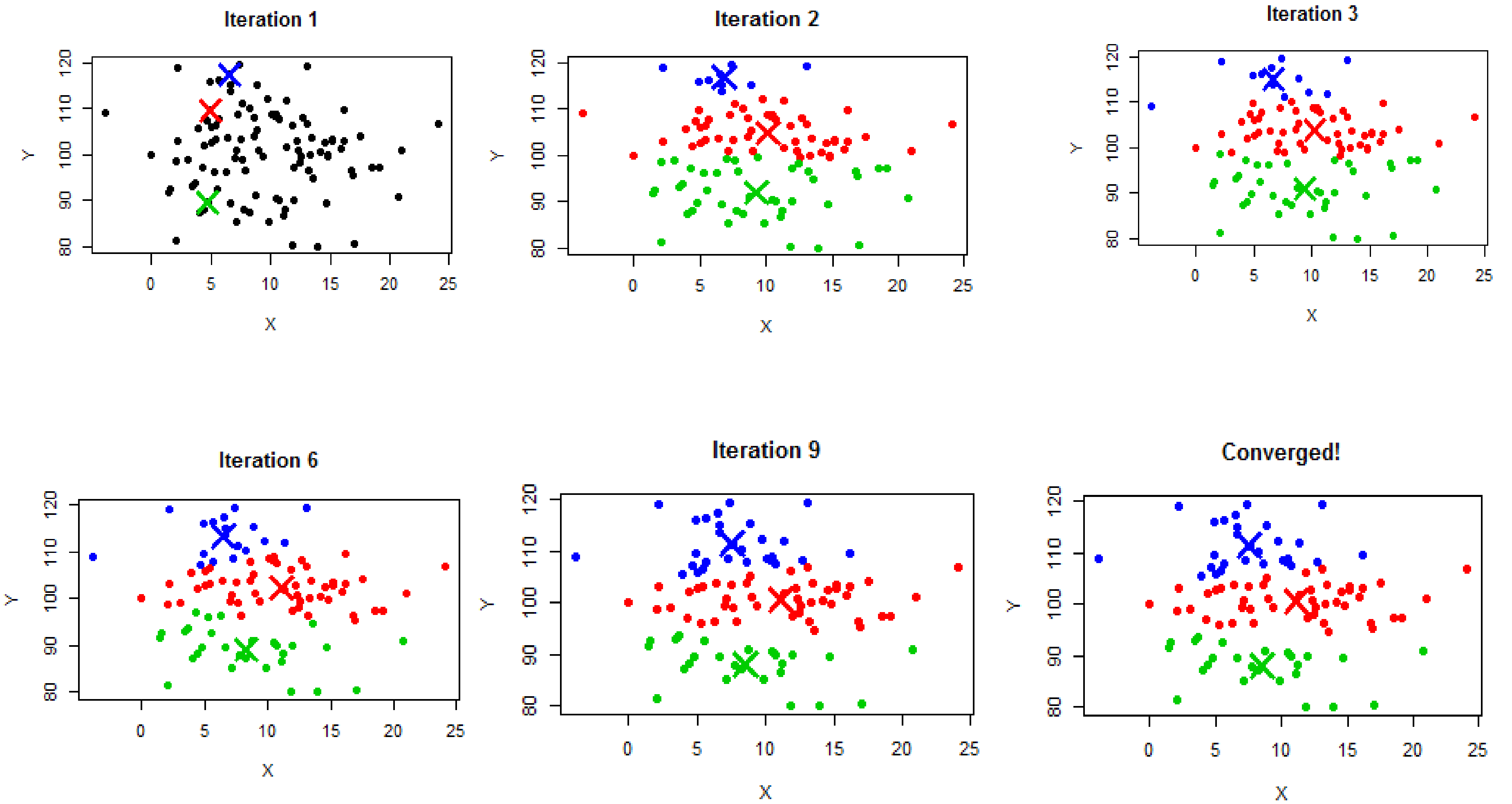


Figure 3: Compute the mean of each new cluster

K-Means Iteration



Elbow Technique for determining K

Determining K using WSS

- How to determine the number of clusters i.e. the K number?
- The value of K can be chosen based on a reasonable guess or some predefined requirements,
  - There is a coefficient that could be computed to determines a reasonable optimal of K which is called Within Sum Square (WSS),
  - WSS is the sum of the squares of the distance between each data point and the closest **centroid**

Elbow Technique

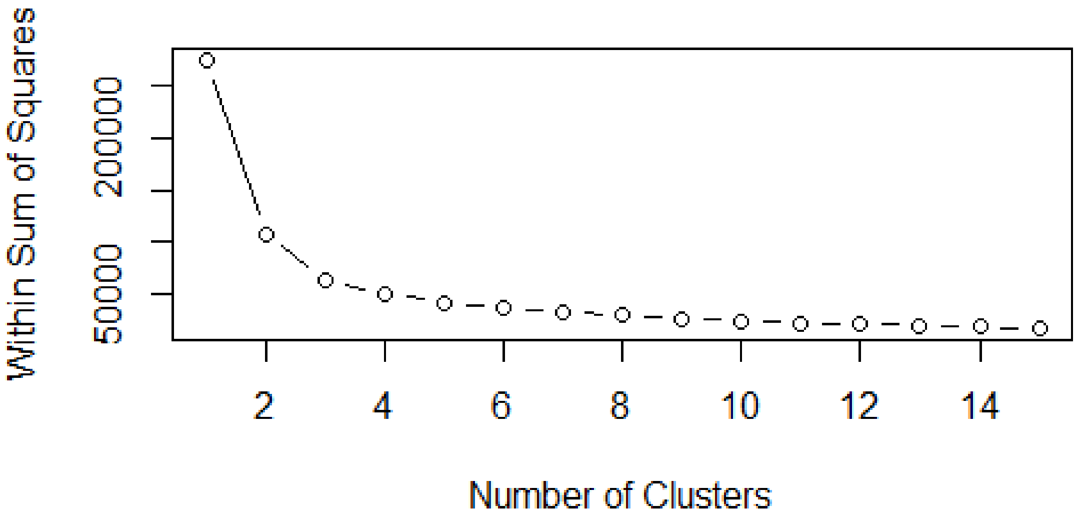
- The elbow method consists in plotting in a graph the WSS(x) value (within-cluster sums of squares) on y-axis according to the number x of clusters considered on the x-axis,
- The WSS(x) value being the sum for all data points of the squared distance between one data point x_i of a cluster j and the centroid of this cluster j (as written in the formula below), after having portioned the dataset in x clusters with the k-means method.

$$\mathrm{WCSS}(k) = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \text{cluster } j} \|\mathbf{x}_i - \bar{\mathbf{x}}_j\|^2,$$

where $\bar{\mathbf{x}}_j$ is the sample mean in cluster j

Elbow Plot

The elbow method consists in plotting in a graph the WSS(x) value as shown below



In this Elbow Plot , the best K value is either 3 or 4

Using R to Perform a K-means Analysis

- The following example uses R to perform a k-means analysis. The task is to group 620 high school seniors based on their grades in three subject areas: English, mathematics, and science.
- The grades are averaged over their high school career and assume values from 0 to 100.
- The following R code establishes the necessary R libraries and imports the CSV file containing the grades.

Using R to Perform a K-means Analysis

- library(factoextra)
- student <- read.csv("c:/data/grades_km_input.csv")

Using R to Perform a K-means Analysis

- The following R code formats the grades for processing. The data file contains four columns.
- The first column holds a student identification (ID) number, and the other three columns are for the grades in the three subject areas.
- Because the student ID is not used in the clustering analysis, it is excluded from the k-means input matrix, *kmdata*.

  - student_data <- student[,2:4]

Using R to Perform a K-means Analysis

- k3 <- kmeans(student_data,3, nstart=25)
- k3
- The displayed contents of the variable *k3* include the following:
  - The location of the cluster means
  - A clustering vector that defines the membership of each student to a corresponding cluster 1, 2, or 3
  - The WSS of each cluster
  - A list of all the available k-means components

Using R to Perform a K-means Analysis

- In the following code, the **<u>factoextra package </u>**is used to visualize the identified student clusters and centroids.
  - library(factoextra)
  - #install.packages("factoextra")
  - fviz_cluster(k3, student_data)

K-Means on IRIS Dataset

- This section shows k-means clustering of iris data.
- First, we remove species from the data to cluster. After that, we apply function **kmeans()** to iris2, and store the clustering result in kmeans.result.

- The cluster number is set to **3** in the code below.

```
iris2 <- iris
iris2
iris2$Species <- NULL
iris2
kmeans.result <- kmeans(iris2, 3)
```

K-Means case study, comparison with real class (Species)

- The clustering result is then compared with the class label (Species) to check whether similar objects are grouped together.

```
table(iris$Species, kmeans.result$cluster)
```

	1	2	3
setosa	0	50	0
versicolor	2	0	48
virginica	36	0	14

- The above result shows that cluster \setosa" can be easily separated from the other clusters, and that clusters \versicolor" and \virginica" are to a small degree overlapped with each other.

K-Means case study, Plotting the results

- Next, the clusters and their centers are plotted
- Note that there are four dimensions in the data and that only the first two dimensions are used to draw the plot below.
- Some black points close to the green center (asterisk) are actually closer to the black center in the four-dimensional space.
- We also need to be aware that the results of k-means clustering may vary from run to run, due to random selection of initial cluster centers.
- plot(iris2[c("Sepal.Length", "Sepal.Width")], col = kmeans.result$cluster)
- # plot cluster centers
- **points(kmeans.result$centers[,c("Sepal.Length", "Sepal.Width")], col = 1:3, pch = 8, cex=2)**

Pch, cex for shapes

- pch=0,square
  pch=1,circle
  pch=2,triangle point up
  pch=3,plus
  pch=4,cross
  pch=5,diamond
  pch=6,triangle point down
  pch=7,square cross
  pch=8,star
  pch=9,diamond plus
  pch=10,circle plus
  pch=11,triangles up and down
  pch=12,square plus
  pch=13,circle
- …….

cex controls the symbol size in the plot,
default is cex=1,

col controls the color of the symbol border,
default is col="black".