Unsupervised Learning: Clustering

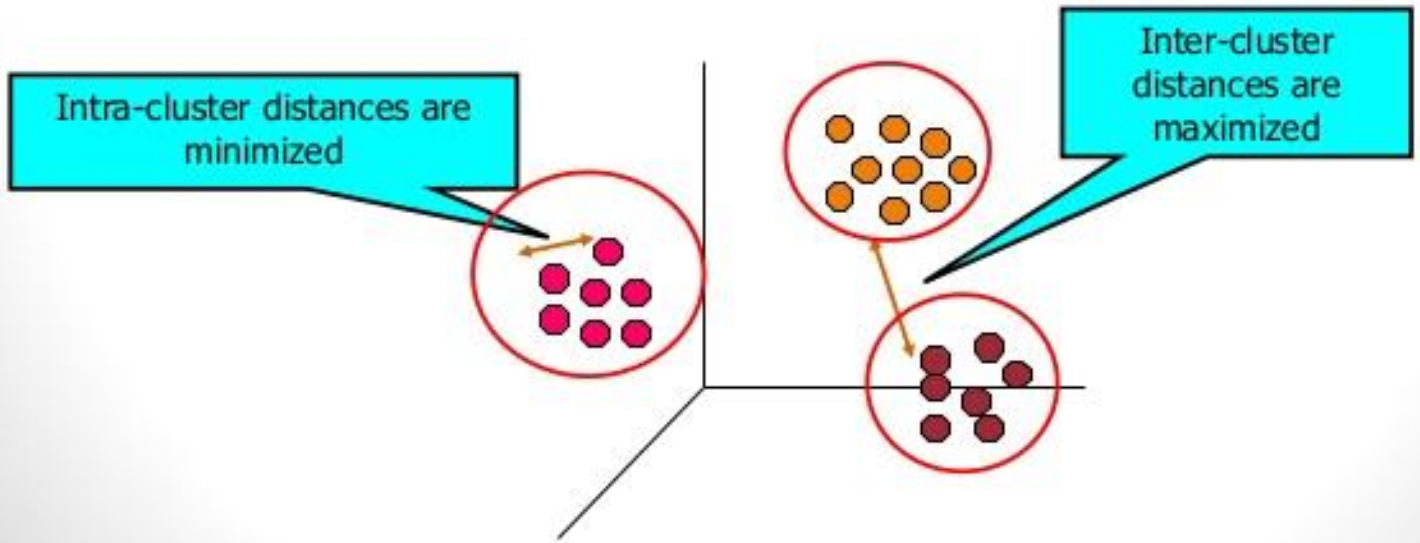
Presented by-
Anup Singh
4th year, DMS student
IISER Kolkata

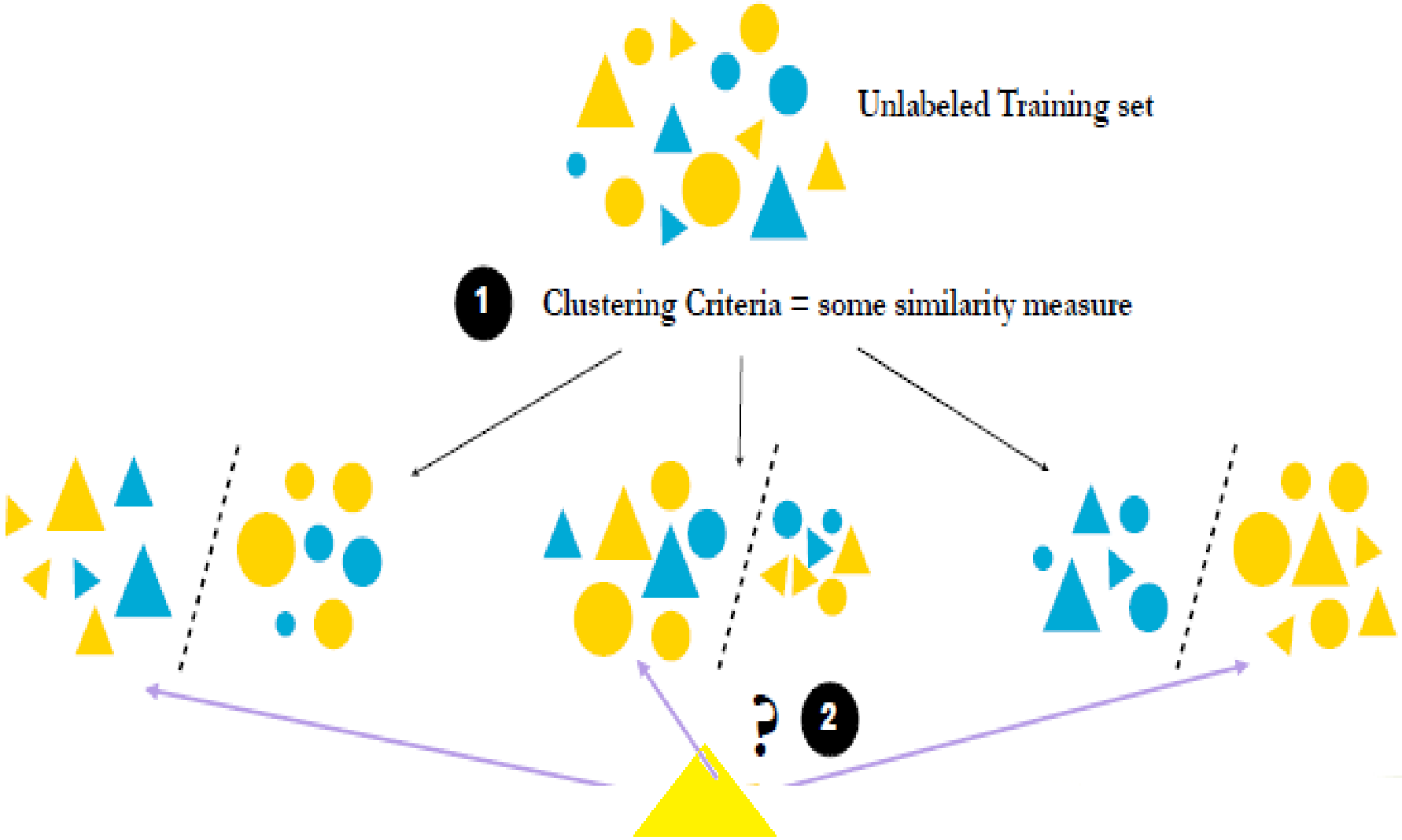
Clustering

- Clustering means grouping the objects based on the information found in the data describing the objects or their relationships.
- The goal is that the objects in a group will be similar (or related) to one other and different from (or unrelated to) the objects in other groups.
- grouped according to logical relationships or consumer preferences.
- unsupervised learning no target field,
- bottom-up approach.
- originated in anthropology by Driver and Kroeber in 1932 and introduced to psychology by Zubin in 1938 and Robert tryon in 1939 and famously used by Cattell beginning in 1943 for trait theory classification in personality psychology.

Segmentation and Cluster Analysis

- Cluster is a group of similar objects (cases, points, observations, examples, members, customers, patients, locations, etc)
- Finding the groups of cases/observations/ objects in the population such that the objects are
  - Homogeneous within the group (high intra-class similarity)
  - Heterogeneous between the groups(low inter-class similarity)




Unlabeled Training set
1
Clustering Criteria = some similarity measure
?
2

What is the need of segmentation?

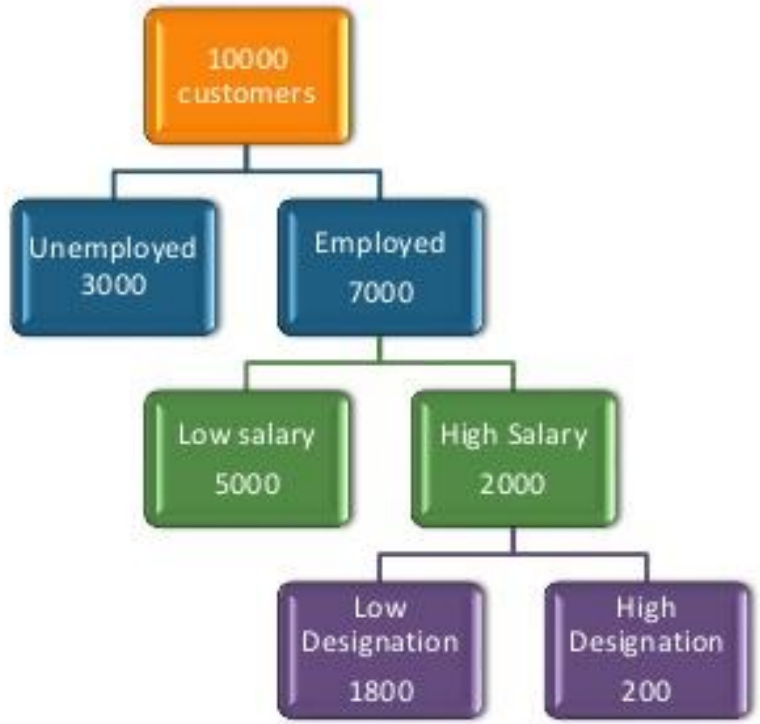
Problem:

- 10,000 Customers - we know their age, city name, income, employment status, designation
- You have to sell 100 Blackberry phones(each costs $1000) to the people in this group. You have maximum of 7 days
- If you start giving demos to each individual, 10,000 demos will take more than one year. How will you sell maximum number of phones by giving minimum number of demos?

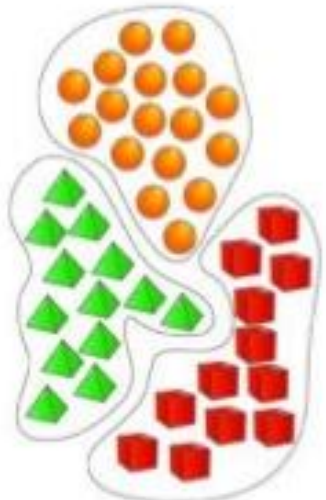
What is the need of segmentation?

Solution

- Divide the whole population into two groups employed / unemployed
- Further divide the employed population into two groups high/low salary
- Further divide that group into high /low designation

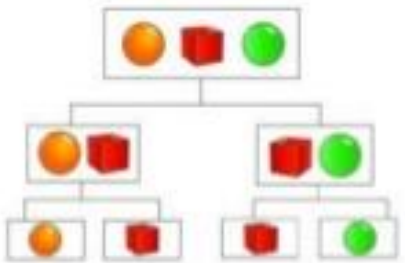


Types of Clusters



- ***Partitional*** **clustering or non-hierarchical :** A division of objects into non-overlapping subsets (clusters) such that each object is in exactly one cluster
- The non-hierarchical methods divide a dataset of N objects into M clusters.
- **K-means clustering**, a non-hierarchical technique, is the most commonly used one in business analytics

- ***Hierarchical*** **clustering:** A set of nested clusters organized as a hierarchical tree
- The hierarchical methods produce a set of nested clusters in which each pair of objects or clusters is progressively nested in a larger cluster until only one cluster remains



K-means clustering

Algorithm *k-means*

1. Randomly choose K data items from X as initial centroids.

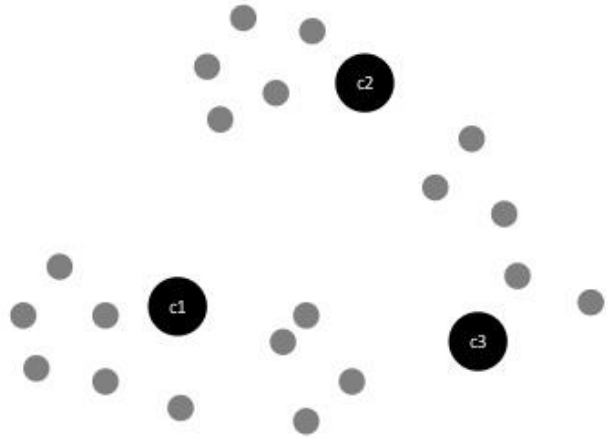
2. Repeat

   - Assign each data point to the cluster which has the closest centroid.

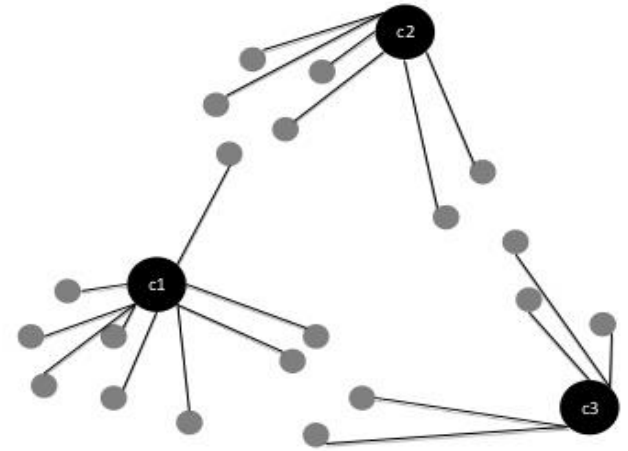
   - Calculate new cluster centroids.

   Until the convergence criteria is met.

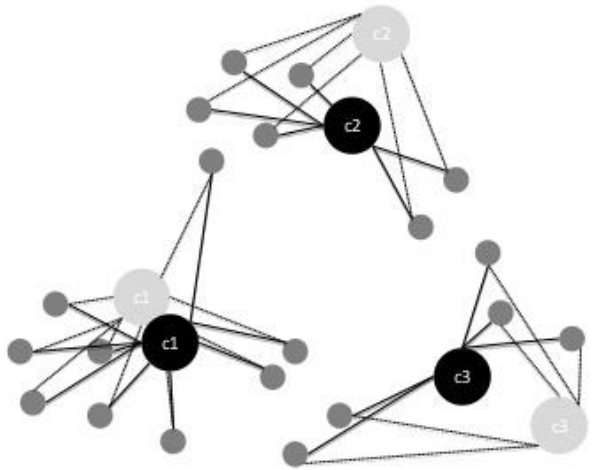
K-Means clustering



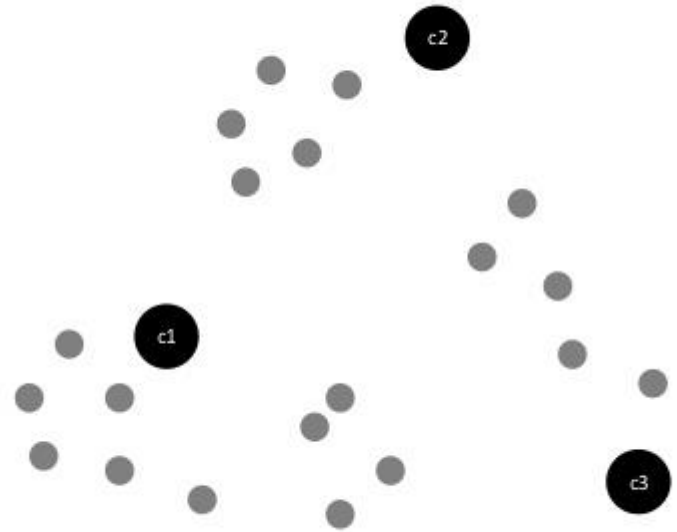
K-Means clustering



K-Means clustering



K-Means clustering

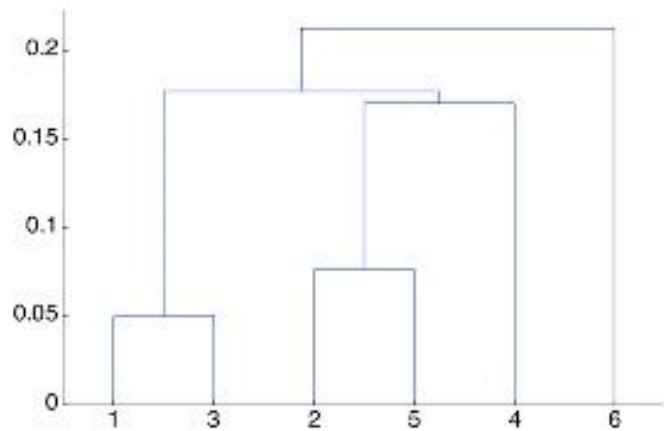


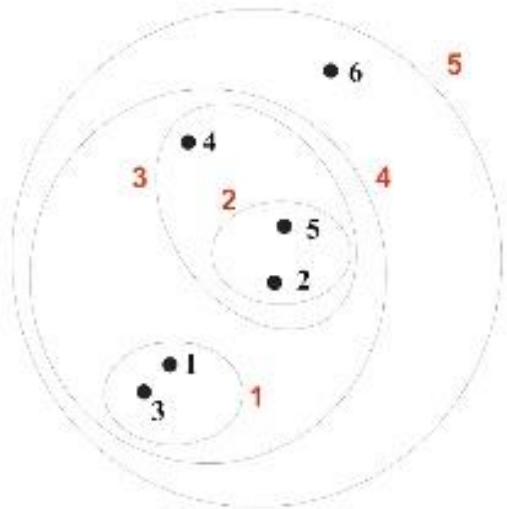
Comments on the *K-Means* Method

- Strength
  - *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Weakness
  - Applicable only when *mean* is defined, then what about categorical data?
  - Need to specify k, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Hierarchical Clustering

- Produces a set of **nested clusters** organized as a hierarchical tree
- Can be visualized as a **dendrogram**
  - A tree-like diagram that records the sequences of merges or splits





Hierarchical Clustering Algorithms

- Two main types of hierarchical clustering
  - **Agglomerative:**
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or **k** clusters) left
  - **Divisive:**
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are **k** clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative clustering algorithm

- Most popular hierarchical clustering technique
- Basic algorithm:
  - Compute the distance matrix between the input data points
  - Let each data point be a cluster
  - **Repeat**
    - Merge the two closest clusters
    - Update the distance matrix
  - **Until** only a single cluster remains

  Key operation is the computation of the distance between two clusters
  - Different definitions of the distance between clusters lead to different algorithms

Distance between two clusters

- **Single-link distance** between clusters C_i and C_j is the **minimum distance** between any object in C_i and any object in C_j

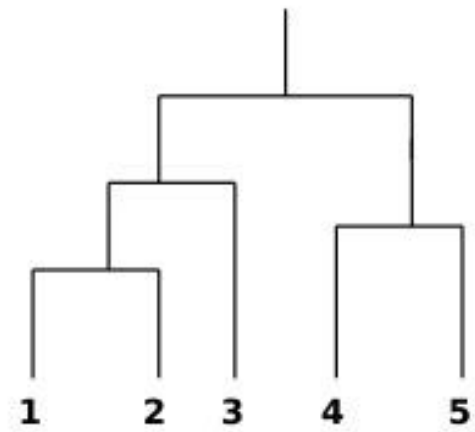
- The distance is **defined by the two most similar objects**

$$D_{\text{single}} = \min_{x,y}\{d(x,y) \mid x \in C_i, y \in C_j\}$$

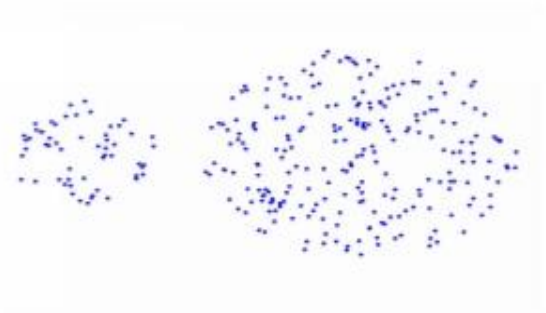
Single-link clustering: example

- Determined by one pair of points, i.e., by one link in the proximity graph.

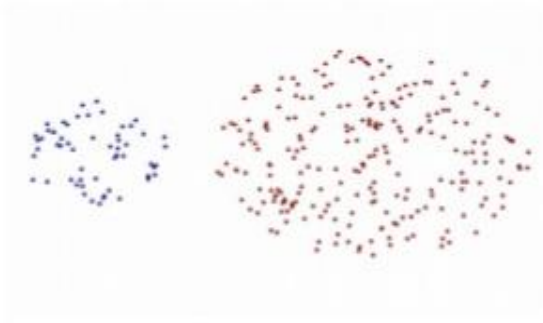
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Strengths of single-link clustering



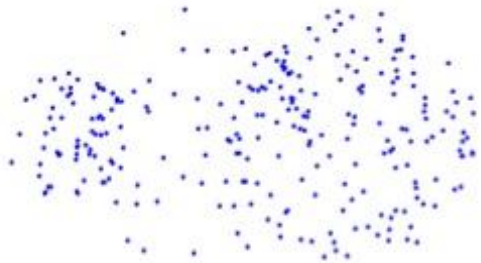
Original Points



Two Clusters

- Can handle non-elliptical shapes

Limitations of single-link clustering



Original Points

Two Clusters

- Sensitive to noise and outliers
- It produces long, elongated clusters

Distance between two clusters

- **Complete-link distance** between clusters C_i and C_j is the **maximum distance** between any object in C_i and any object in C_j

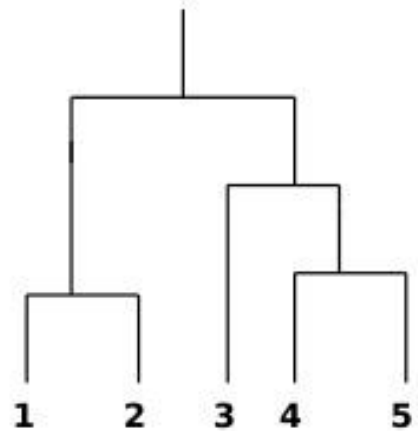
- The distance is **defined by the two most dissimilar objects**

$$D_{\text{complete}} = \max_{x,y}\{d(x,y) \mid x \in C_i, y \in C_j\}$$

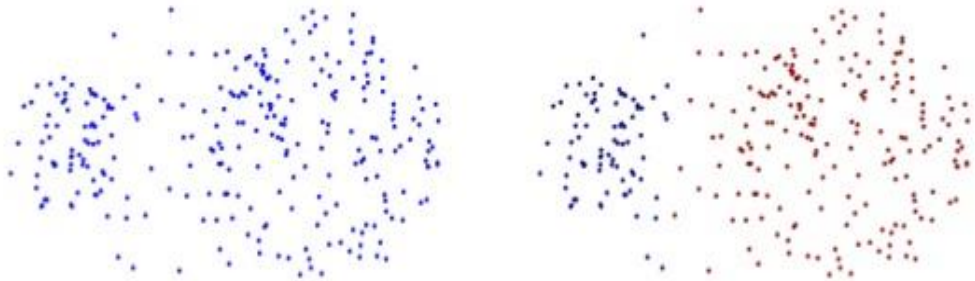
Complete-link clustering: example

- Distance between clusters is determined by the two most distant points in the different clusters

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



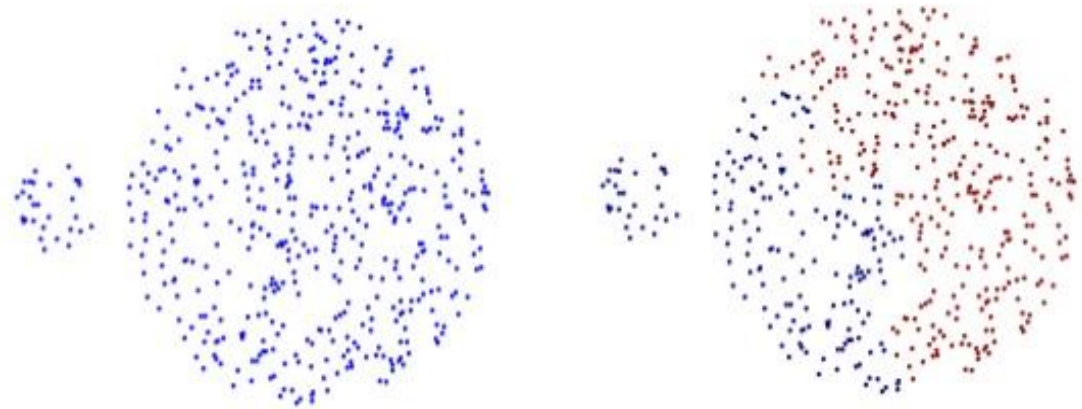
Strengths of complete-link clustering



Original Points Two Clusters

- More balanced clusters (with equal diameter)
- Less susceptible to noise

Limitations of complete-link clustering



Original Points Two Clusters

- Tends to break large clusters
- All clusters tend to have the same diameter – small clusters are merged with larger ones

Distance between two clusters

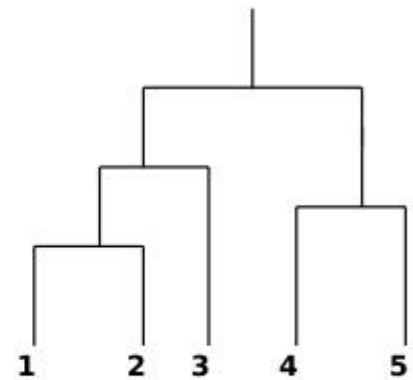
- **Group average distance** between clusters C_i and C_j is the **average distance** between any object in C_i and any object in C_j

$$D_{\text{average}} = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$

Average-link clustering: example

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Average-link clustering: discussion

- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

E.M.

Expectation Maximization

- A general-purpose algorithm for MLE in a wide range of situations.
- First formally stated by Dempster, Laird and Rubin in 1977 [1]
  - We even have several books discussing only on EM and its variations!
- An excellent way of doing our unsupervised learning problem, as we will see
  - EM is also used widely in other domains.

EM: a solution for MLE

- Given a statistical model with:
  - a set $\mathbf{X}$ of observed data,
  - a set $\mathbf{Z}$ of unobserved latent data,
  - a vector of unknown parameters $\boldsymbol{\theta}$,
  - a likelihood function $L\left(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}\right) = p\left(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}\right)$

- Roughly speaking, the aim of **MLE** is to determine $\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L\left(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}\right)$
  - We known the old trick: *partial derivatives* of the log likelihood...
  - But it is not always tractable [*e.g.*]
  - Other solutions are available.

EM: General Case

$$L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) = p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})$$

- EM is just an iterative procedure for finding the MLE
- **Expectation step:** keep the current estimate $\boldsymbol{\theta}^{(t)}$ fixed, calculate the expected value of the log likelihood function

$$Q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}\right) = E\left[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})\right] = E\left[\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})\right]$$

- **Maximization step**: Find the parameter that maximizes this quantity

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}\right)$$

GMM

Gaussian Mixture Model

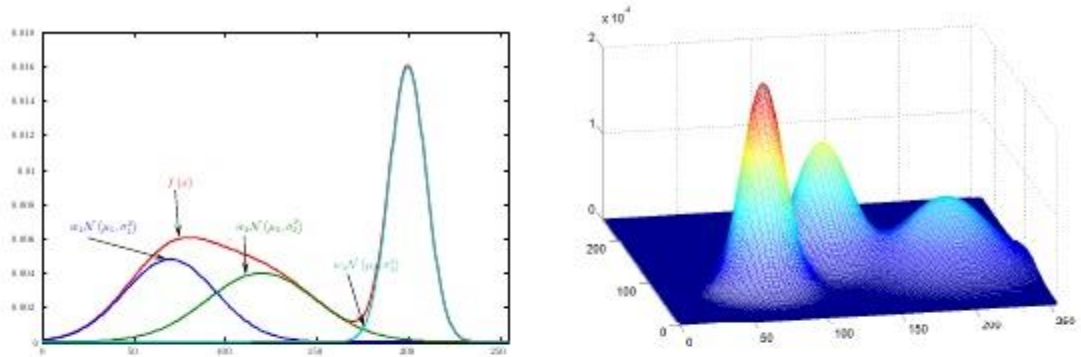
The Problem

- You have data that you believe is drawn from **n**populations
- You want to identify parameters for each population
- You don't know anything about the populations *a priori*
  - Except you believe that they're gaussian…

Probability density function of GMM

"Linear combination" of Gaussians:

$$f(\mathbf{x}) = \sum_{i=1}^{k} w_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), \quad \text{where} \sum_{i=1}^{k} w_i = 1$$



(a) The *pdf* of an 1D GMM with 3 components.

(b) The *pdf* of an 2D GMM with 3 components.

Figure 2: Probability density function of some GMMs.

GMM: Problem definition

$$f(\mathbf{x}) = \sum_{i=1}^{k} w_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), \quad \text{where} \sum_{i=1}^{k} w_i = 1$$

Given a training set, how to model these data point using GMM?

- **Given:**
  - The trainning set: $\{\mathbf{x_i}\}_{i=1}^{N}$
  - Number of clusters: k
- **Goal:** model this data using a mixture of Gaussians
  - Weights: $w_1, w_2, ..., w_k$
  - Means and covariances: $\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, ..., \boldsymbol{\mu_k}; \boldsymbol{\Sigma_1}, \boldsymbol{\Sigma_2}, ..., \boldsymbol{\Sigma_k}$

Computing likelihoods in unsupervised case

$$f(\mathbf{x}) = \sum_{i=1}^{k} w_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), \quad \text{where } \sum_{i=1}^{k} w_i = 1$$

- Given a mixture of Gaussians, denoted by G. For any $\mathbf{x}$, we can define the likelihood:

$$\begin{aligned} \mathcal{P}(\mathbf{x} \mid G) &= \mathcal{P}(\mathbf{x} \mid w_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, ..., w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \sum_{i=1}^{k} \mathcal{P}(\mathbf{x} \mid c_i) \mathcal{P}(c_i) \\ &= \sum_{i=1}^{k} w_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) \end{aligned}$$

- So we can define likelihood for the whole training set [Why?]

$$\begin{aligned} \mathcal{P}(\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N} \mid G) &= \prod_{i=1}^{N} \mathcal{P}(\mathbf{x_i} \mid G) \\ &= \prod_{i=1}^{N} \sum_{j=1}^{k} w_j \mathcal{N}(\mathbf{x_i}; \boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}) \end{aligned}$$

Estimating GMM parameters

- We known this: **M**aximum **L**ikelihood **E**stimation

$$\ln \mathcal{P}\left(\mathbf{X} \mid G\right) = \sum_{i=1}^{N} \ln \left[\sum_{j=1}^{k} w_j \mathcal{N}\left(\mathbf{x_i}; \boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}\right) \right]$$

  - For the max likelihood:

$$0 = \frac{\partial \ln \mathcal{P}\left(\mathbf{X} \mid G\right)}{\partial \boldsymbol{\mu_j}}$$

  - This leads to non-linear non-analytically-solvable equations!

- Use gradient descent
  - Slow but doable

- A much cuter and recently popular method...

E.M. for General GMM: E-step

- On the t'th iteration, let our estimates be

$$\boldsymbol{\lambda}_t = \{\boldsymbol{\mu}_1(t), \boldsymbol{\mu}_2(t), ..., \boldsymbol{\mu}_k(t), \boldsymbol{\Sigma}_1(t), \boldsymbol{\Sigma}_2(t), ..., \boldsymbol{\Sigma}_k(t), w_1(t), w_2(t), ..., w_k(t)\}$$

- E-step: compute the *expected* classes of all data points for each class

$$\begin{aligned}\tau_{ij}(t) \equiv p(c_j \mid \mathbf{x}_i, \boldsymbol{\lambda}_t) &= \frac{p(\mathbf{x}_i \mid c_j, \boldsymbol{\lambda}_t)\, p(c_j \mid \boldsymbol{\lambda}_t)}{p(\mathbf{x}_i \mid \boldsymbol{\lambda}_t)} \\ &= \frac{p\left(\mathbf{x}_i \mid c_j, \boldsymbol{\mu}_j(t), \boldsymbol{\Sigma}_j(t)\right) w_j(t)}{\sum_{m=1}^{k} p(\mathbf{x}_i \mid c_m, \boldsymbol{\mu}_m(t), \boldsymbol{\Sigma}_j(t))\, w_m(t)}\end{aligned}$$

E.M. for General GMM: M-step

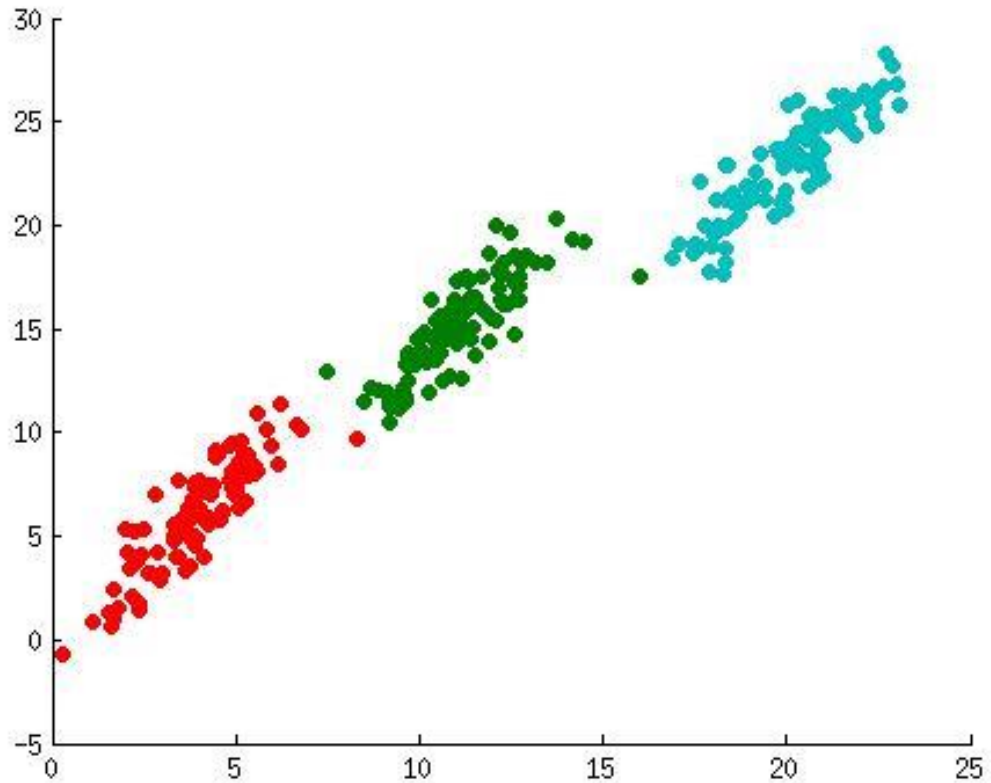
- M-step: compute $\boldsymbol{\mu}$ given our data's class membership distributions

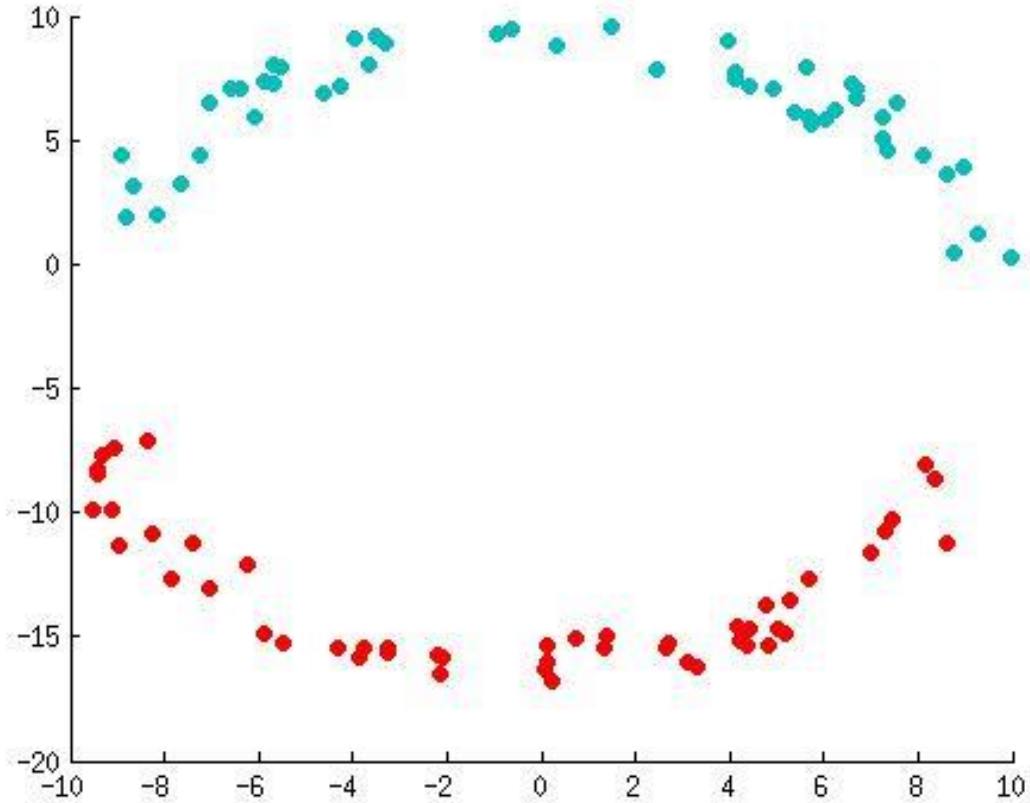
$$
\begin{aligned}
w_j(t+1) &= \frac{\sum_{i=1}^{N} p(c_j \mid \mathbf{x}_i, \boldsymbol{\lambda}_t)}{N} \\
&= \frac{1}{N} \sum_{i=1}^{N} \tau_{ij}(t)
\end{aligned}
\qquad
\begin{aligned}
\boldsymbol{\mu}_j(t+1) &= \frac{\sum_{i=1}^{N} p(c_j \mid \mathbf{x}_i, \boldsymbol{\lambda}_t)\, \mathbf{x}_i}{\sum_{i=1}^{N} p(c_j \mid \mathbf{x}_i, \boldsymbol{\lambda}_t)} \\
&= \frac{1}{N w_j(t+1)} \sum_{i=1}^{N} \tau_{ij}(t)\, \mathbf{x}_i
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{\Sigma}_j(t+1) &= \frac{\sum_{i=1}^{N} p(c_j \mid \mathbf{x}_i, \boldsymbol{\lambda}_t) \left[\mathbf{x}_i - \boldsymbol{\mu}_j(t+1)\right] \left[\mathbf{x}_i - \boldsymbol{\mu}_j(t+1)\right]^{\mathrm{T}}}{\sum_{i=1}^{N} p(c_j \mid \mathbf{x}_i, \boldsymbol{\lambda}_t)} \\
&= \frac{1}{N w_j(t+1)} \sum_{i=1}^{N} \tau_{ij}(t) \left[\mathbf{x}_i - \boldsymbol{\mu}_j(t+1)\right] \left[\mathbf{x}_i - \boldsymbol{\mu}_j(t+1)\right]^{\mathrm{T}}
\end{aligned}
$$

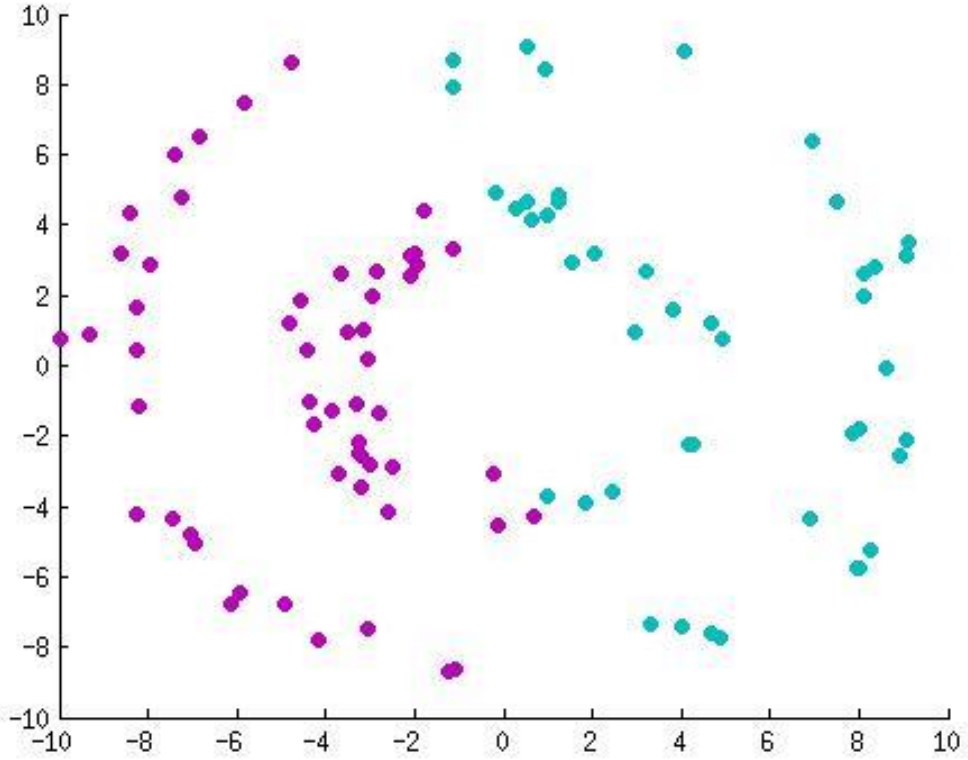
APPLICATIONS

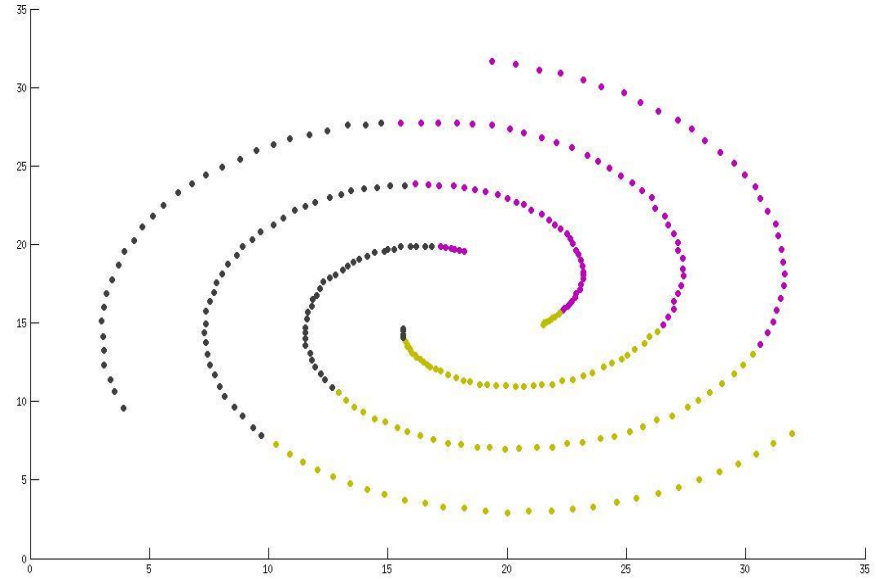
K-Means

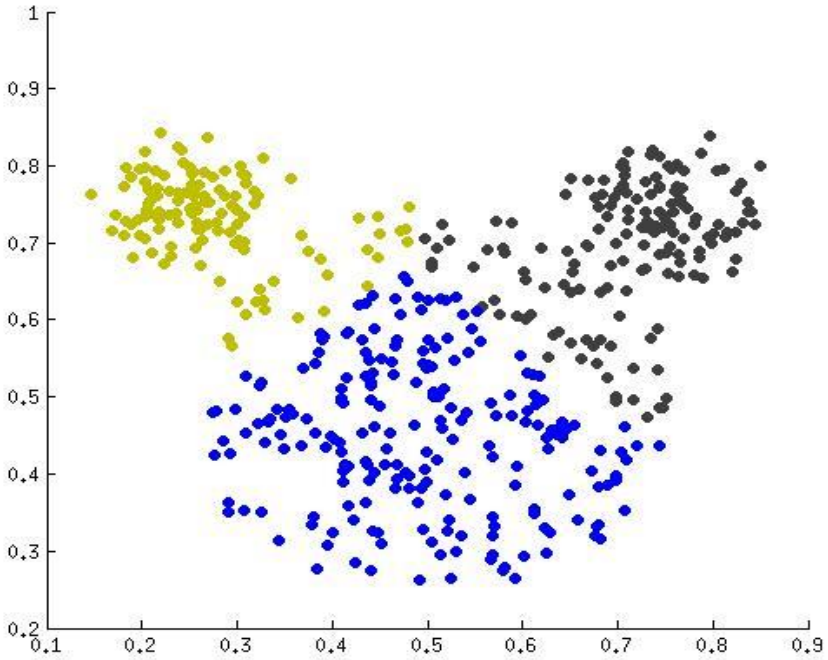




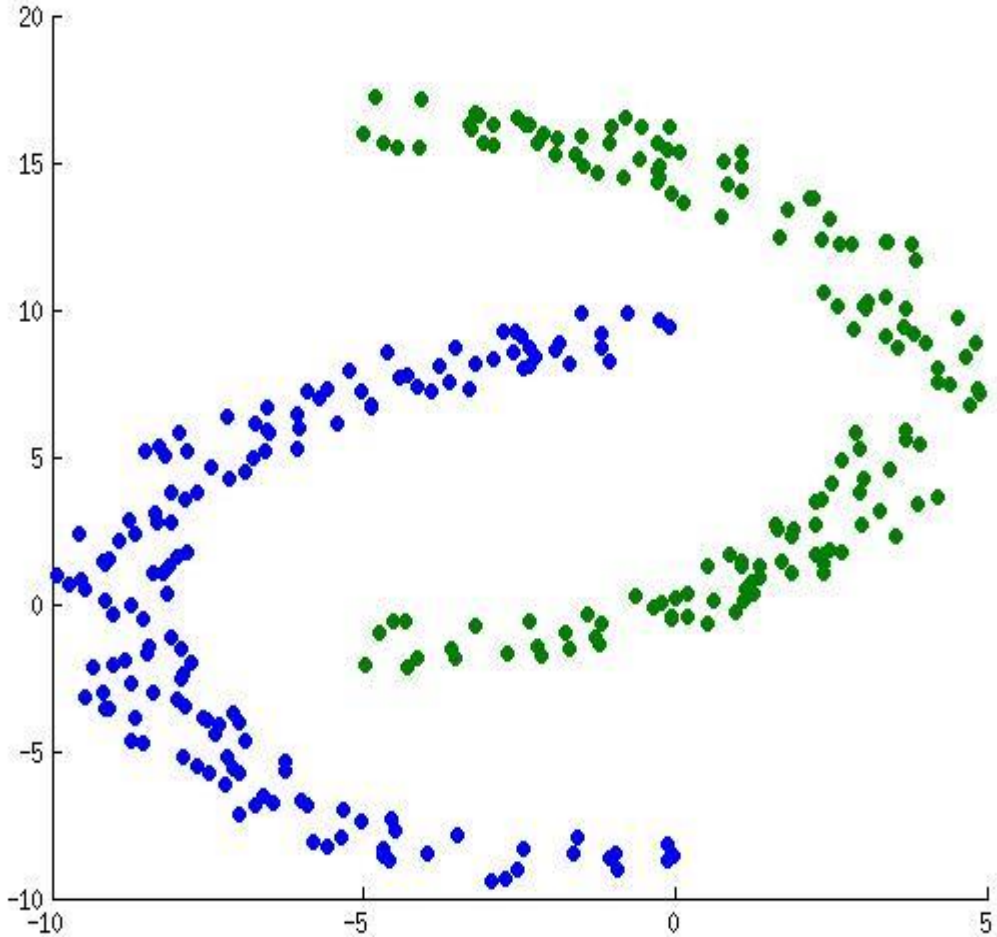
K-Means

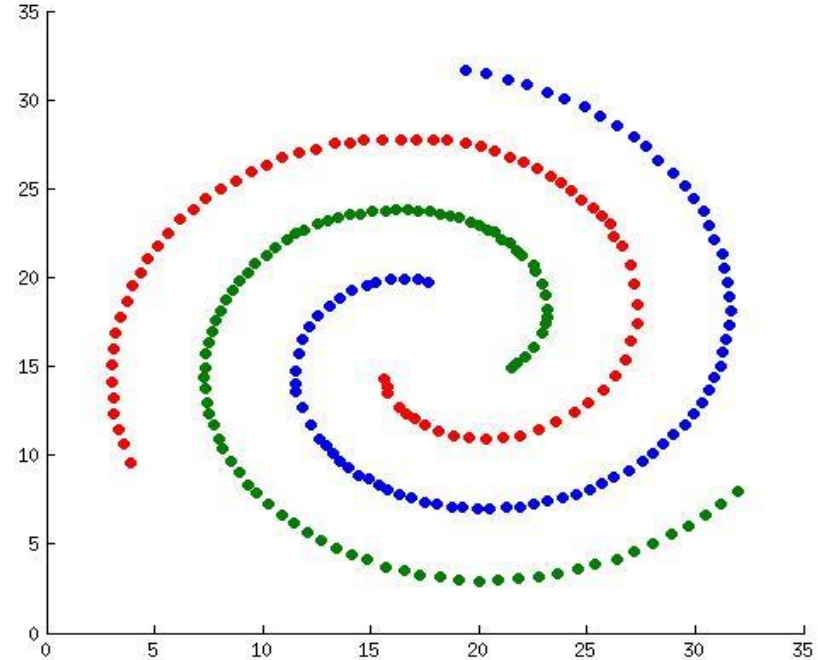

10
8
6
4
2
0
−2
−4
−6
−8
−10
−10
−8
−6
−4
−2
0
2
4
6
8
10

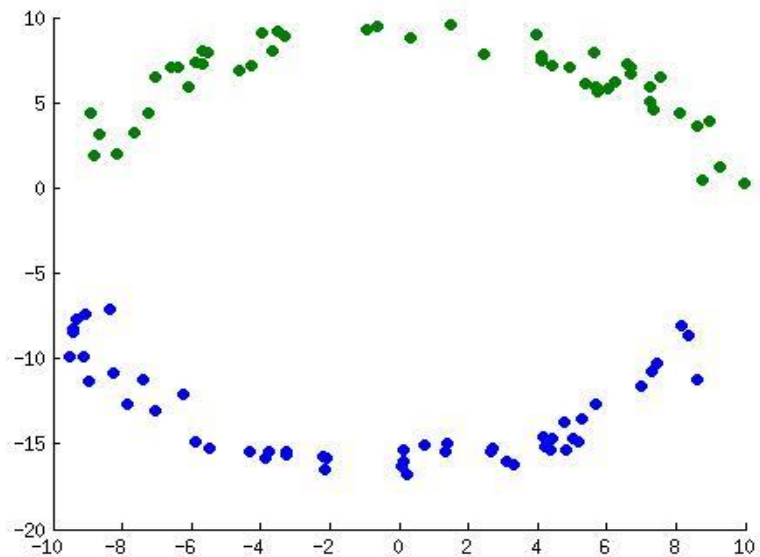

35
30
25
20
15
10
5
0
0
5
10
15
20
25
30
35


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

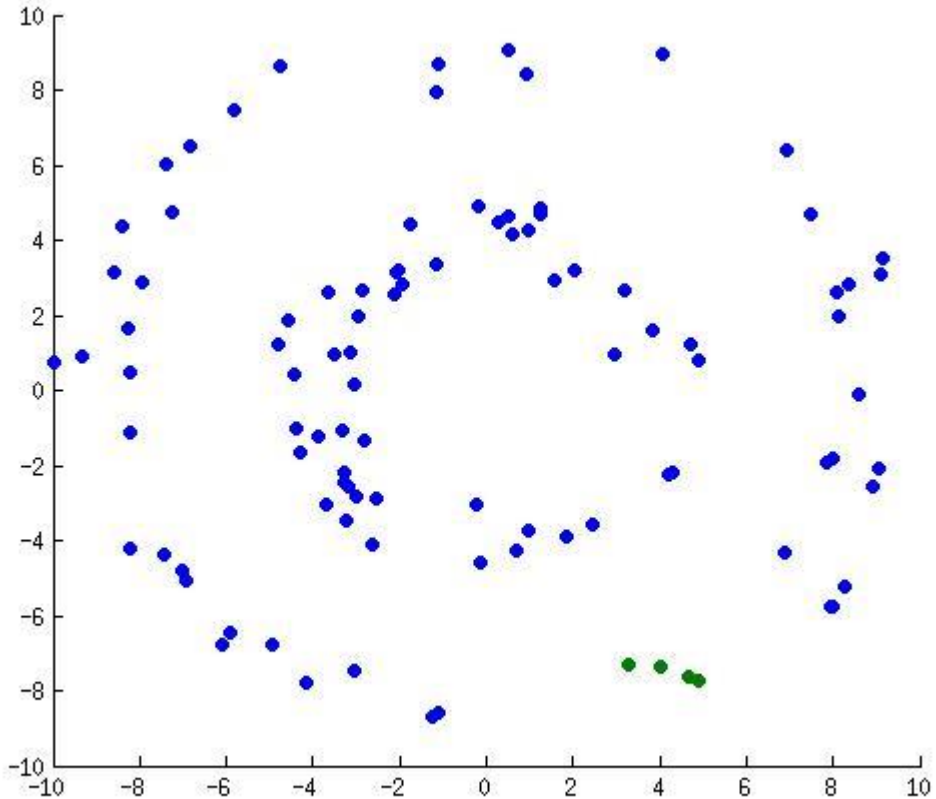
Single – Linkage

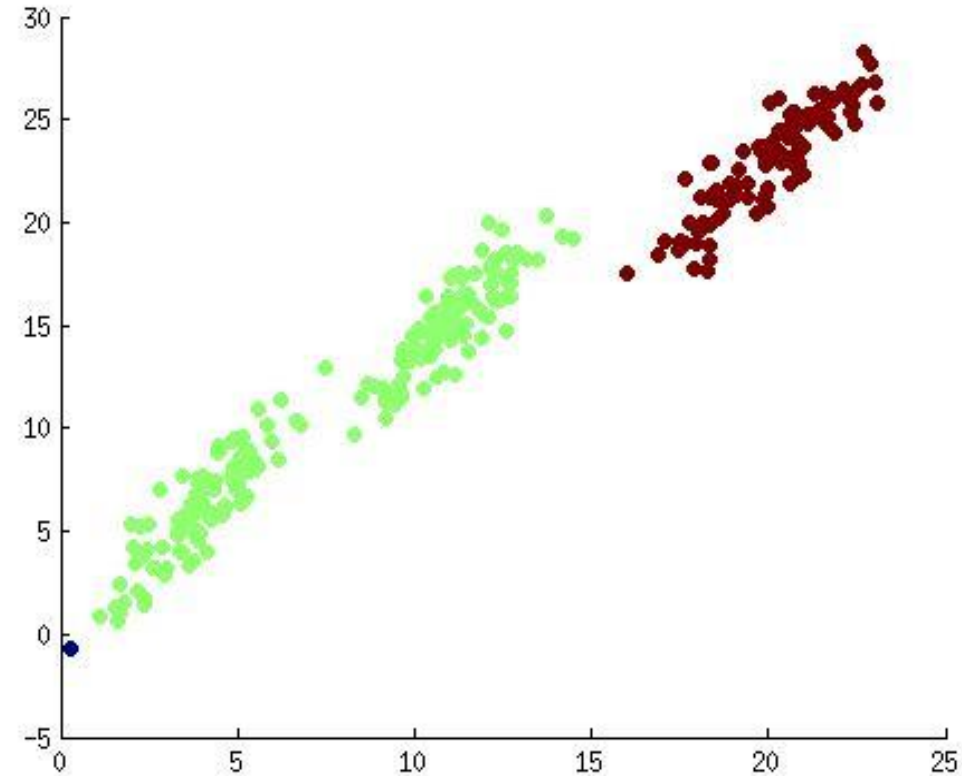




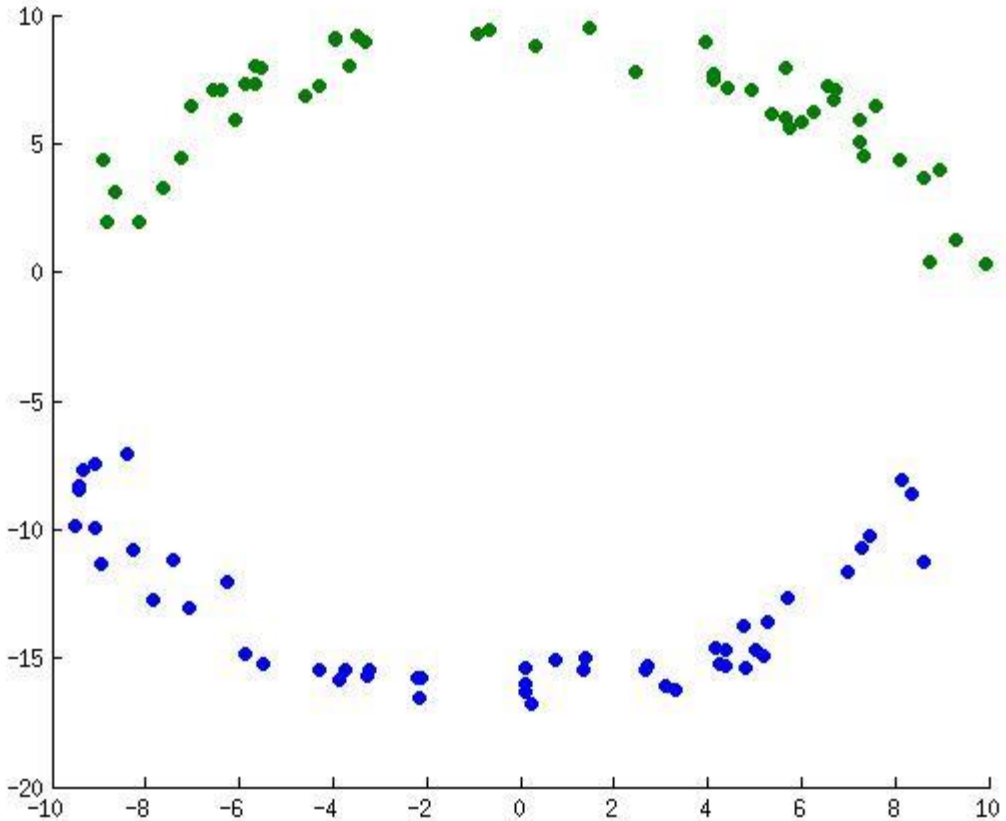


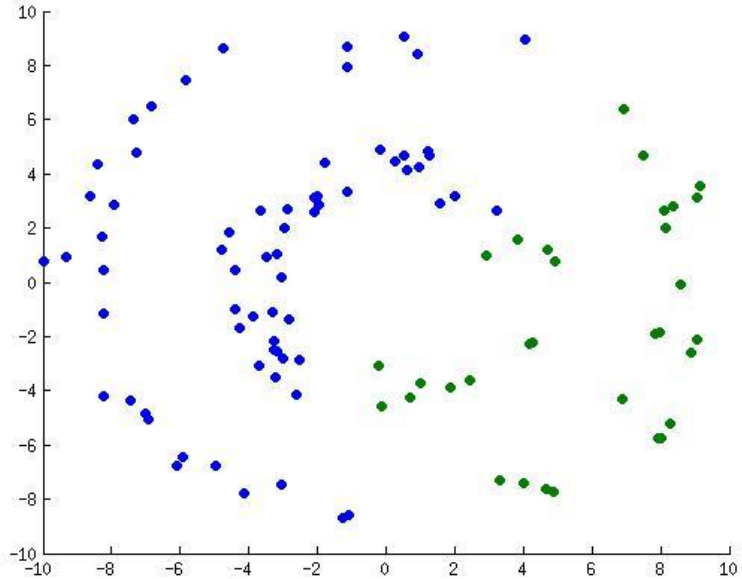
Single – Linkage

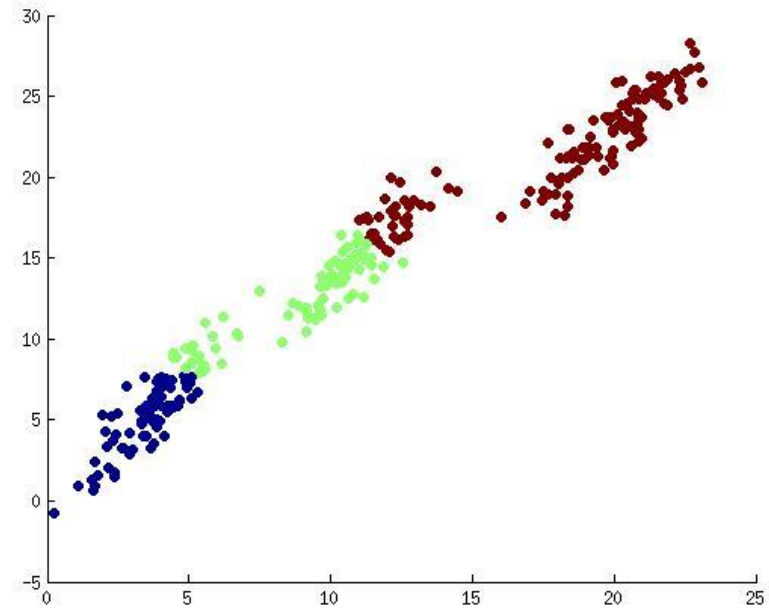




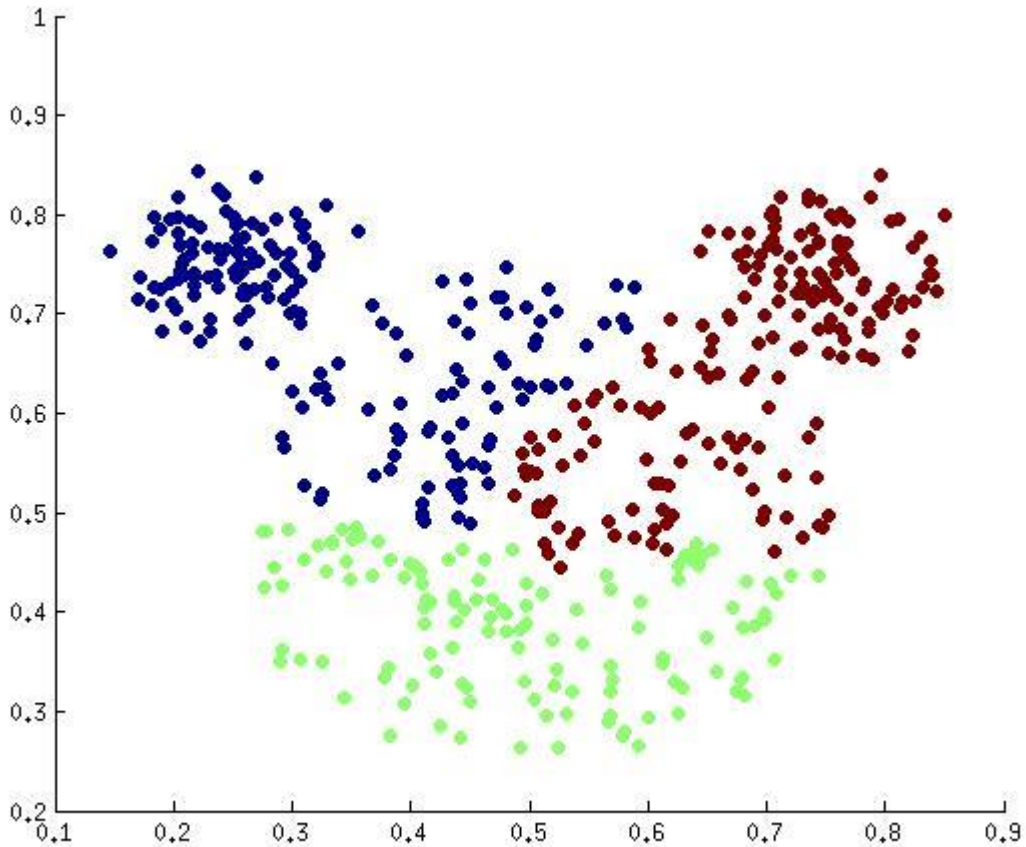
Complete - Linkage

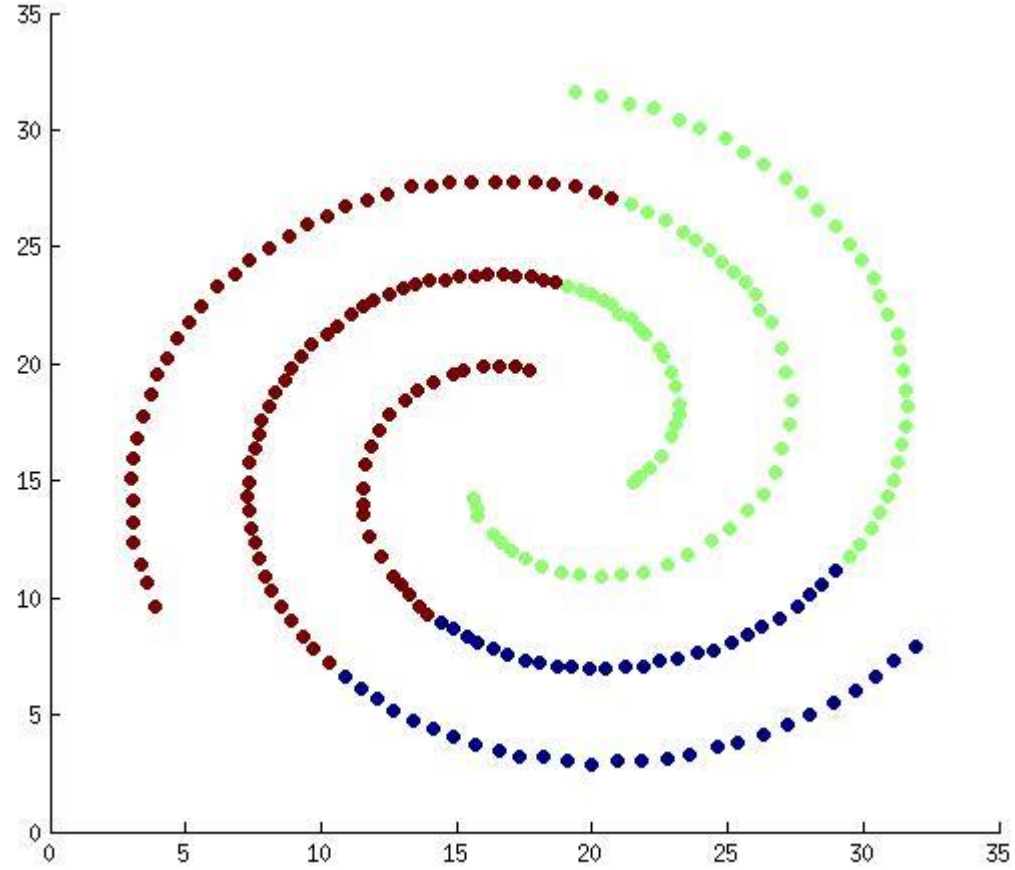




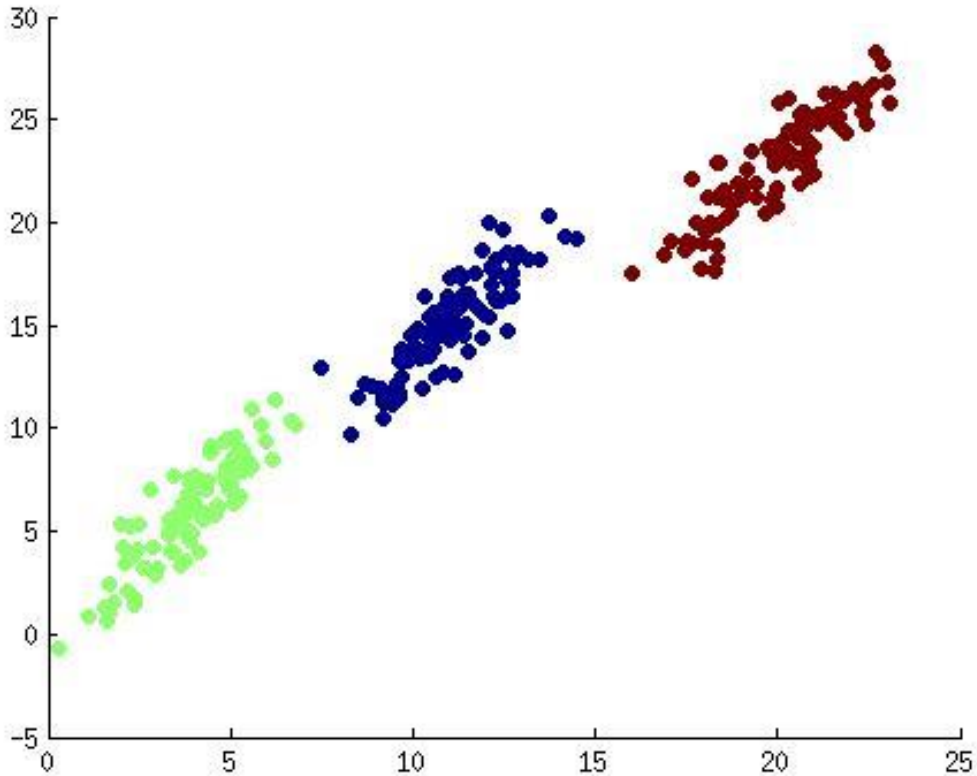


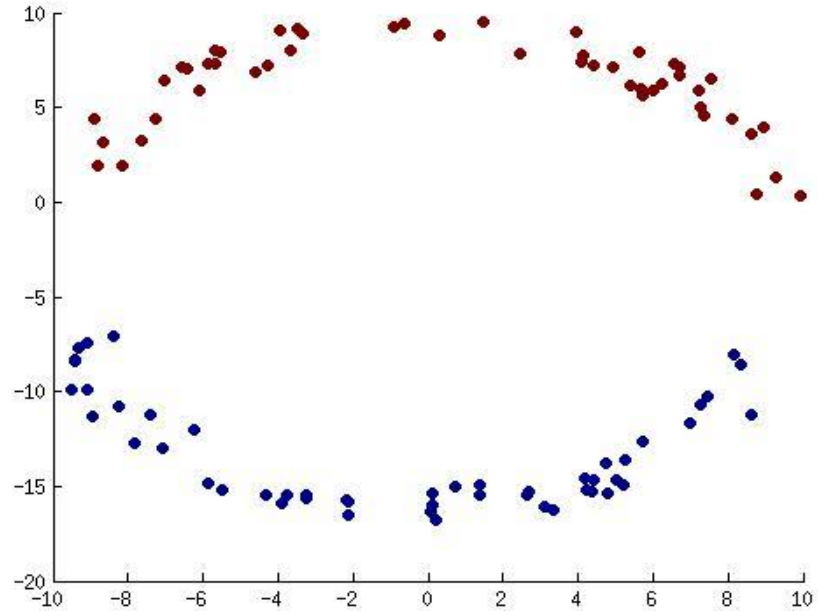
Complete - Linkage

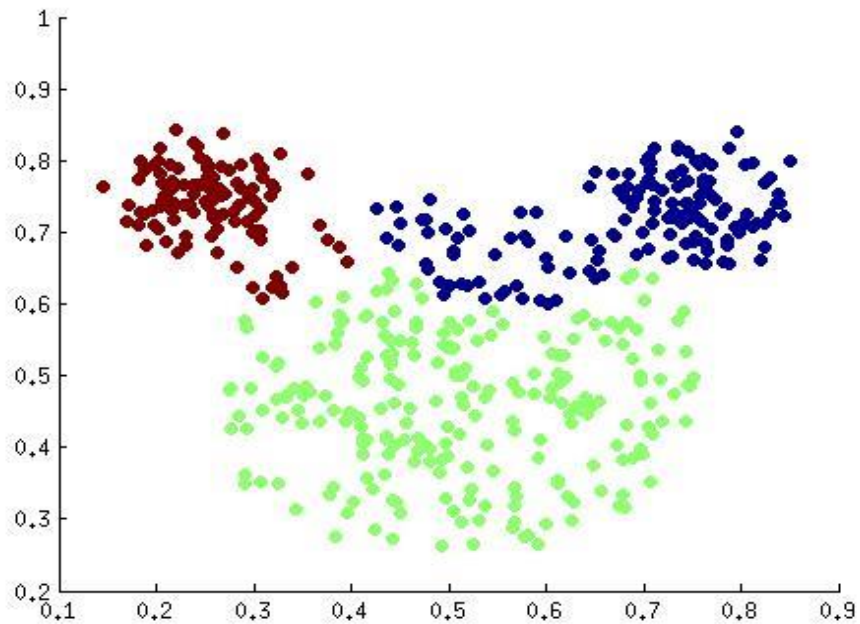




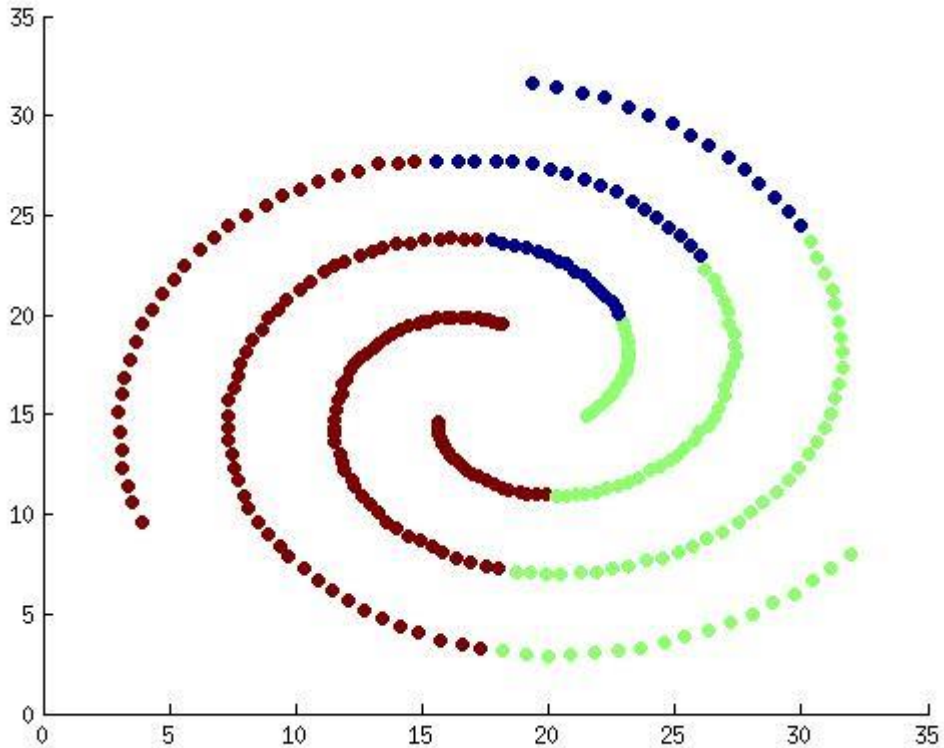
Average - Linkage

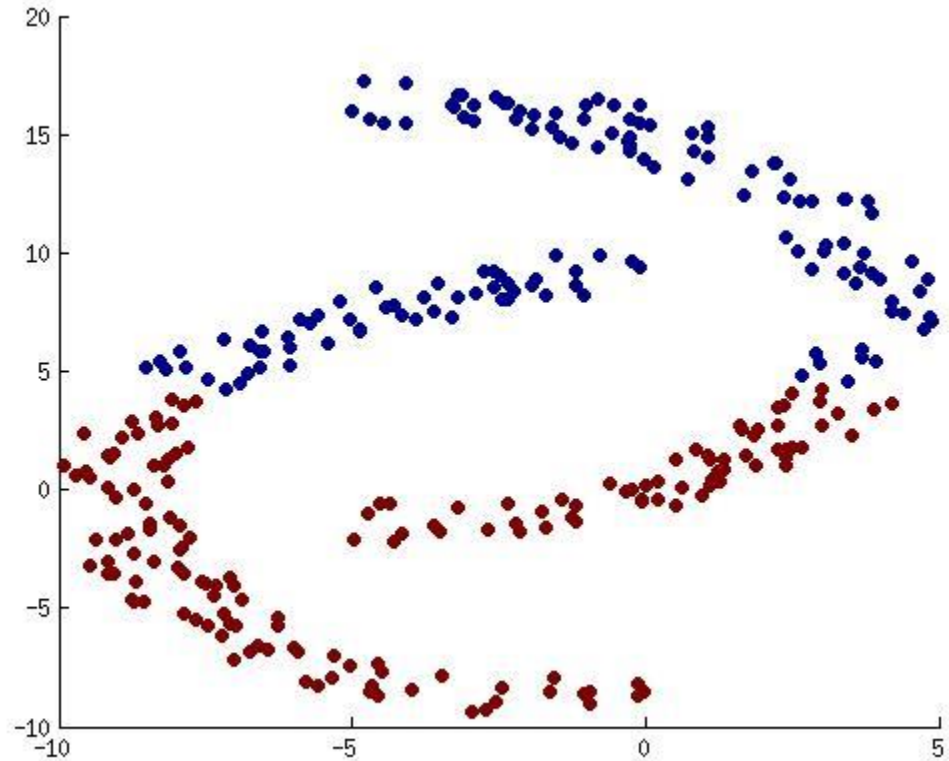




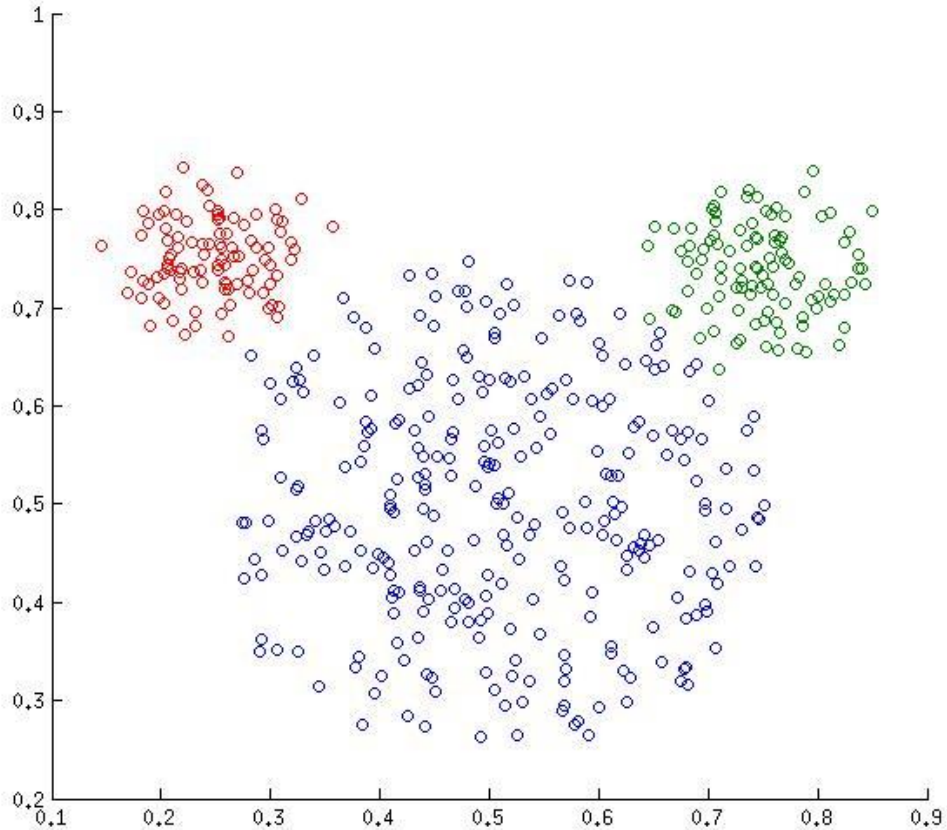


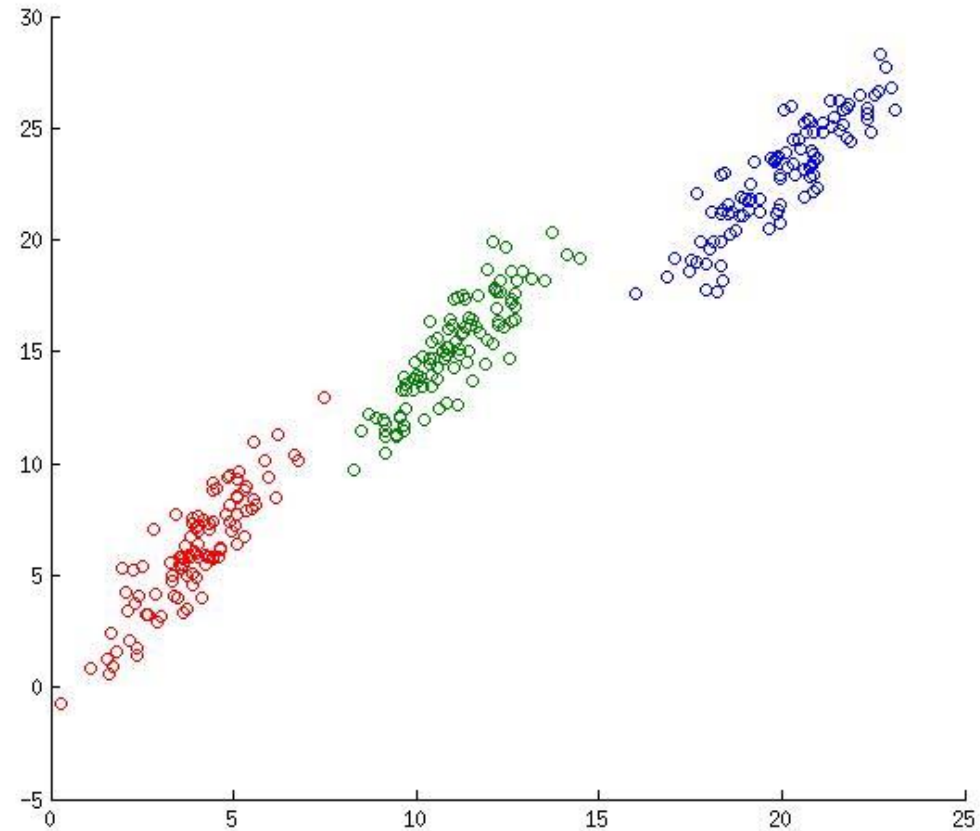
Average - Linkage



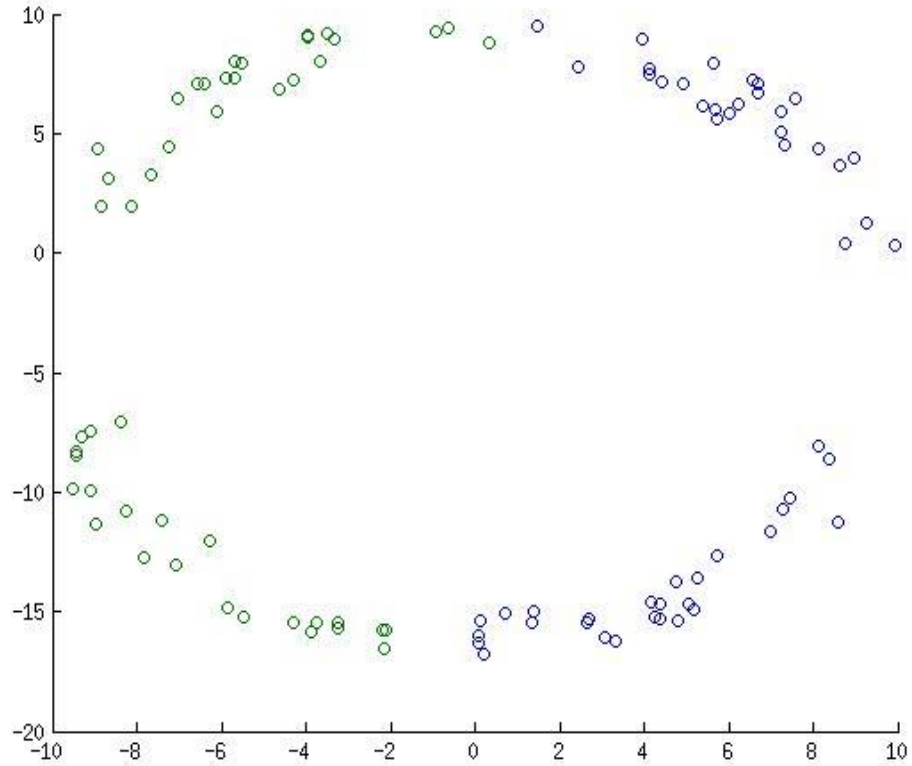


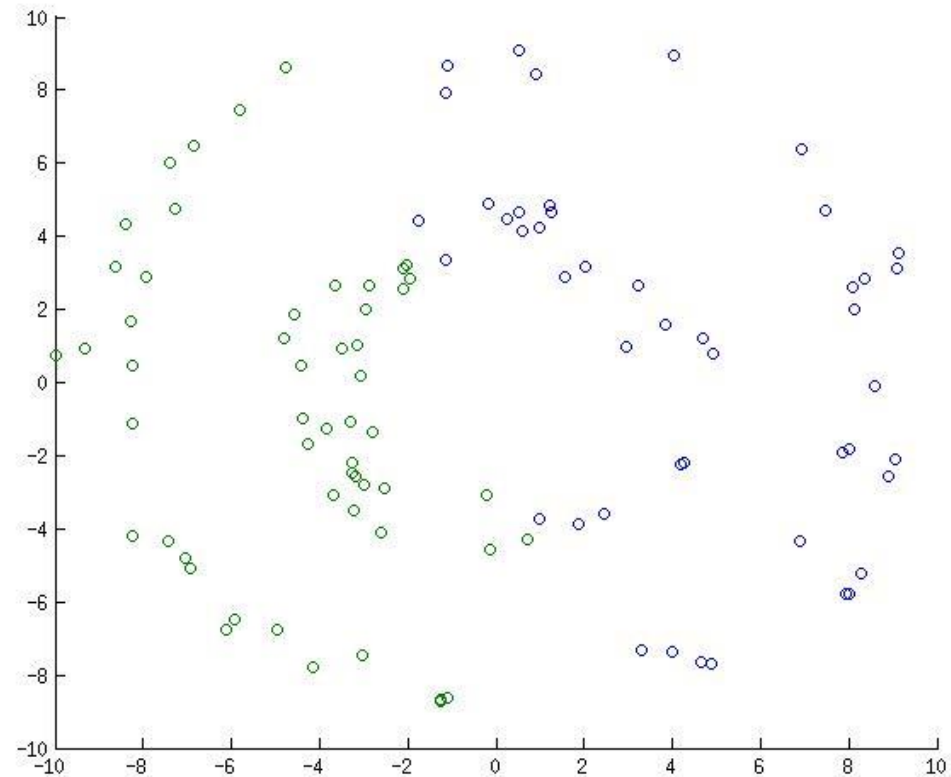
Clustering using EM





Clustering using EM





Thank You