CSE/ECE-471: Statistical Methods in AI **Spring 2020**

GMM and Hierarchical Clustering

Prepared by:
Ravikumar Nandigam (20173101)
Tirth Pandit (2019201017)
Kumar Pallav (2019201096)

Contents

1 K-means Revisited

1. Training set : $\{\mathrm{x}^1, ..., x^m\}, x^i \in \mathbb{R}^n$

   The k-means clustering algorithm is as follows:

   1. Initialize cluster centroids $\mu 1, \mu 2, ..., \mu k \in \mathbb{R}^n \, randomly.$
   2. Repeat until convergence:

      For every i, set

      $$\mathrm{c}^i := \arg\min_j \|x^i - \mu^j\|^2$$

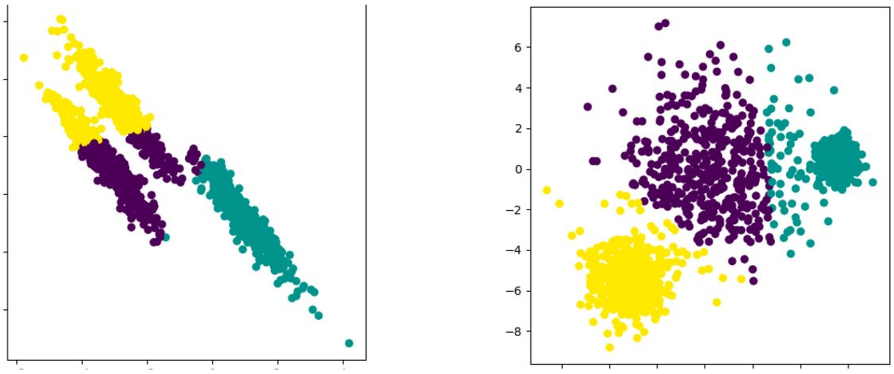
      For each j ,set

      $$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}$$

2. Problems With K-means

   a. Does Hard Assignments

   b. Due to Euclidean distance ,K-means makes spherical clusters



2 GMM and Maximum Likelihood Revisited

- Maximum Likelihood

Data $= x1, x2.., xn.$

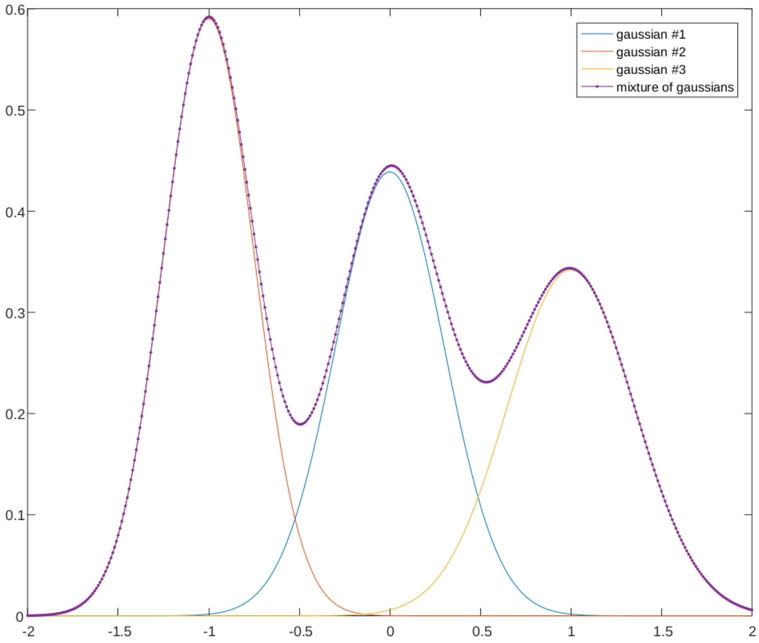
Likelihood Function : $\mathrm{L}(\Theta) = Pr(Data|\Theta)$

$$\mathrm{L}(\Theta) = Pr(Data|\Theta) = \prod_{i=1}^{n} p(x_i|\Theta)$$

$$\Theta^* = \arg\min_{\Theta} Pr(Data|\Theta)$$

- Mixtures of Gaussians

Gaussian mixture model is a simple linear super position of Gaussian components.



$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leqslant \pi_k \leqslant 1$$

Likelihood Function

$$p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \quad |\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

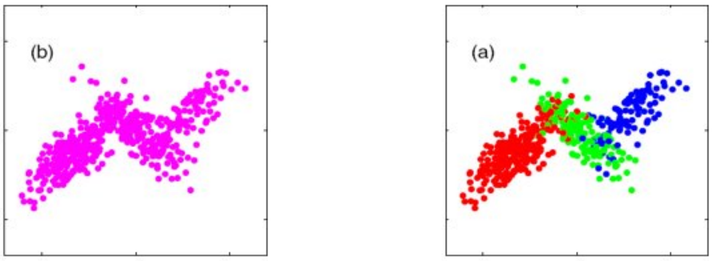
$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

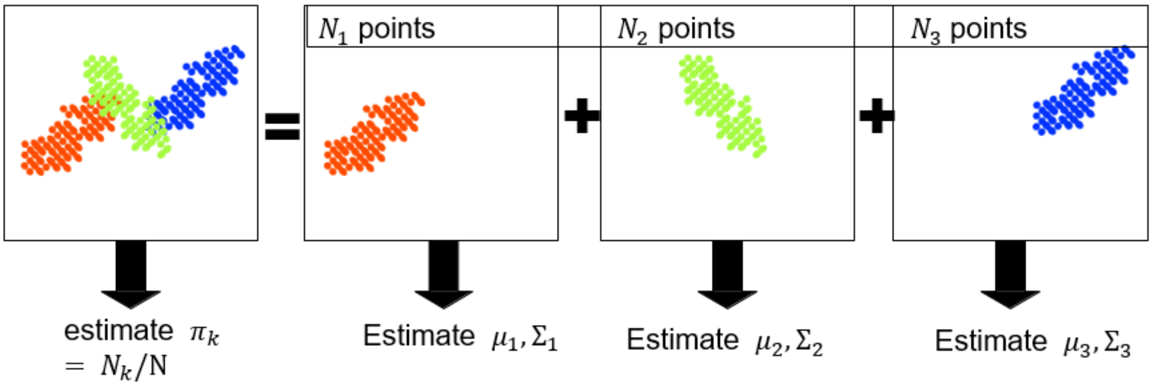
Maximizing this function for a Gaussian mixture model is complex.Difficulty arises due to summation over k that appears inside the logarithm. That why we cant obtain closed form

3 Gaussian Mixture Model Conti.

- We have to device other way to find maximum likelihood.

- Suppose some oracle told us that which point comes from which gaussian. If this happens then we can easily estimate all the parameters for all gaussians.




N_1 points
N_2 points
N_3 points
estimate π_k $= N_k/N$
Estimate μ_1, Σ_1
Estimate μ_2, Σ_2
Estimate μ_3, Σ_3

- This information can be provided by some latent binary variable Z_{nk}. Where $Z_{nk} = 1$ tells that n^{th} point comes from k^{th} gaussian ,otherwise takes value 0

- So ,Now introduce a K-dimensional binary random variable z having a 1-of-K representation .Element z_k is equal to 1 and all other elements are equal to 0.The values of z_k satisfy $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$ .There are K possible states for the vector z according to which element is nonzero.

- Represent Marginal Probability of Z in terms of Mixing Coefficient π_k

$$p(z_k = 1) = \pi_k$$

.

- Parameter π_k Must Satisfy following:

$$0 \leqslant \pi_k \leqslant 1$$

$$\sum_{k=1}^{K} \pi_k = 1$$

.

- Because z uses a 1-of-K representation, Probability of z can be written in terms of the π_k As following .here z_k is indicator variable.

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

.

- Similarly, the conditional distribution of x given a particular value for $z_k = 1$,means probability of point in k^{th} Gaussian

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

.

- Which boils down to

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$$

.

- We can define new likelihood function in terms of joint probability of z and x.

$$p(\text{X,Z}) = \prod_{i=1}^{n} p(x_i, z_i)$$
$$p(\text{X,Z}) = \prod_{i=1}^{n} p(z_i)\, p(x_i|z_i)..$$

- Final equation in terms of z and other parameters π_k ,μ_k and Σ_k is as below

$$p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

.

- Taking logarithm on both side

$$\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \left\{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right\}$$

.

Where π_k = Mixing Coefficient of Kth cluster.
μ_k = Mean of Kth cluster.
Σ_k = Co-variance of Kth cluster.

.

- In Above Likelihood Equation Z_{nk} is Responsible for hard assignment as it is a indicator binary random variable. Z_{nk} can be described as

$$Z_{nk} = \begin{cases} 1, & \text{Point n comes from Kth cluster} \\ 0, & \text{otherwise} \end{cases}$$

.

- Due to Random Variable Z_{nk} Points get hard assignment. So taking expectation of Z_{nk} will map its value to Real value ,which more or less gives notion of soft assignment. So Finding Expectation of Likelihood function as follows.

$$\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})] = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \left\{ \ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}.$$

.

Here $\gamma(z_{nk}) = \mathrm{E}[z_{nk}]$.
$\gamma(z_{nk})$ is expectation of z_{nk} ,its a continues variable between 0 to 1

.

- Expectation of Z_{nk} can be find easily as it is a binary variable

  $\mathrm{E}[Z_{nk}] = \sum_{i=1}^{K} Z_{ni} P(Z_{ni})$.

  since $Z_{ik} \in \{0, 1\}$.

  $\mathrm{E}[Z_{nk}] = \mathrm{P}(Z_{nk} = 1) = \gamma(Z_{nk})$

.

- Interpretation of the $\gamma(Z_{nk})$

  Its shows the extent to which given point N belongs to kth cluster.
  In other words it shows soft membership of point in Kth cluster.
  It can be interpreted as conditional probability of z given x.
  I.e $\mathrm{P}(z_k = 1|\mathrm{x})$,whose value can be obtained by Bayes theorem.

$$\begin{aligned} \gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \end{aligned}$$

  π_k is prior probability of $z_k = 1$
  $\gamma(z_k)$ is corresponding posterior probability once we have observed x

.

4 Expectation Maximization Algorithm for GMM

- Expectation maximization is an iterative algorithm for using maximum likelihood to estimate the parameters of a statistical model with unobserved (hidden)variables. It has two main steps. First is the E-step.We compute some probability distribution of the model so we can use it for expectations.Second comes the M-step, which stands for maximization. In this step, we maximize the lower bound of the log-likelihood function by generating a new set of parameters with respect to the expectations.

- First Find the Conditions for which maximum likelihood can be reached. So for that recall the older maximum likelihood function which is in form of sum of logs and finding optimum parameters from that function is hard. So try to reduce derivative of that function in terms of our new introduced variables

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}.$$

1) First Taking Derivative w.r.t to μ_k.

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k(\mathbf{x}_n - \boldsymbol{\mu}_k).$$

As we defined $\gamma(z_{nk})$ before, it can be used to reduce the term as shown
And after simplifying it, μ_k can be estimated.

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n.$$

where .

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})..$$

N_k can be interpreted as the effective number of points assigned to cluster k

2) Taking Derivative w.r.t to Σ_k we will get.

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}.$$

3) Taking Derivative w.r.t to π_k

Here we must take account of the constraint $\sum \pi_k = 1$
Can be achieved using a Lagrange multiplier

Now maximize following quantity

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

$$0 = \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda$$

Which Gives

$$\pi_k = \frac{N_k}{N}$$

.

- ***EM Algorithm For GMM***

Given a Gaussian mixture model,maximize the likelihood function w.r.t parameters

1) Initialize μ_k , σ_k and π_k and find the initial value of log likelihood.

2) **E step** : Evaluate the $\gamma(Z_{nk})$ using the current parameter values.

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\displaystyle\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

3) **M step** : Re-estimate the parameters from $\gamma(Z_{nk})$

$$\begin{aligned} \boldsymbol{\mu}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \\ \boldsymbol{\Sigma}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\text{T}} \\ \pi_k^{\text{new}} &= \frac{N_k}{N} \end{aligned}$$

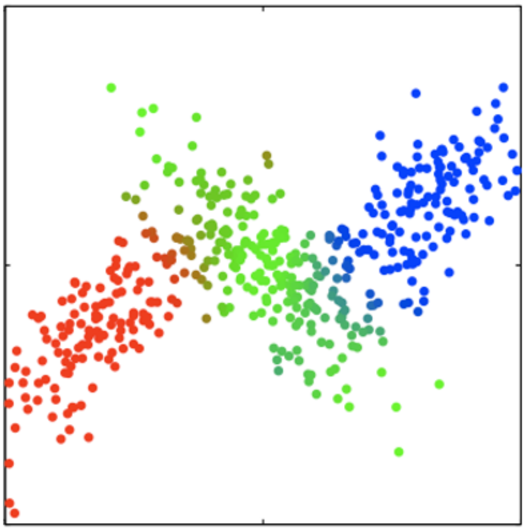
4) Evaluate the log likelihood .

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Check for convergence of either the parameters or the log likelihood.the convergence criterion is not satisfied return to step 2.

- ***Results of EM***

1) EM Gives Soft Assignments



2) All points contribute to estimate all components

3) Each point has unit weight to contribute, but splits it across the K components

4) Weight contributed by point to component is proportional to the likelihood that point was generated by that component.

5 EM and K-means

- K-means can be thought as the special case of EM algorithm

- In K-means we only take cares about mean ,so try to drop out other parameters π_k and Σ_k and try to reduce likelihood function into simple form having μ_k only.

- Considerations for K-means

  1) Mixing Coefficient
     To Drop Mixing Coefficient π_k ,Fix all π_k to 1/K

  2) Fix all co-variances to Σ_k to constant ϵI ,Where ϵ is Variance and I is identity matrix of d*d

  3) Take limit as ϵ goes to 0 ,which behaves as binary assignments .

- So ,now likelihood function for single gaussian changes as below

$$p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi\epsilon)^{1/2}} \exp\left\{-\frac{1}{2\epsilon}\|\mathbf{x} - \boldsymbol{\mu}_k\|^2\right\}$$

- Now ,consider mixture of gaussians. Here all Σ is converted to constant ϵ and Mixing Coefficients are also considered constant. So now we don't have to estimate these parameters in EM algorithm

- So now $\gamma(Z_nk)$ for a particular data point x_n can be written as below.

$$\gamma(z_{nk}) = \frac{\pi_k \exp\left\{-\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2/2\epsilon\right\}}{\sum_j \pi_j \exp\left\{-\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2/2\epsilon\right\}}.$$

- Now consider limit $\epsilon \to 0$,in denominator the term for which $\|x_n - \mu_j\|^2$ is smallest will go to zero most slowly, and hence the $\gamma(z_{nk})$ for the data point x_n all go to zero except for term j ,for which the it will go to unity.Thus, in this limit, we obtain a hard assignment of data points to clusters, just as in the K-means algorithm, so that $\gamma(z_{nk}) \to$z$_{nk}$

$$\gamma(z_{nj}) = z_{nj} = \begin{cases} 1 & \text{if } \mu_j \text{ is closest mean to } x_n \\ 0 & \text{otherwise} \end{cases}$$

hard labels, as in the K–means algorithm

- So from the limit it can be seen that maximizing the expected log likelihood is equivalent to minimizing the distortion measure J for the K-means algorithm

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

$$\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})] \to -\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 + \text{const}$$

Where $r_{nk} = z_{nk}$

- ***K-means Algorithm***

  1) Given N data points $x_1, x_2, ..., x_N$

  2) Find K cluster centers $\mu_1, \mu_2, ..., \mu_N$ to minimize

  $$\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \left\|x_n - \mu_j\right\|^2$$

  3) Algorithm

     a) initialize K cluster centers $\mu_1, \mu_2, ..., \mu_N$

     b) **E-step** : set z_{nk} labels to assign each point to closest cluster center

     c) **M-step** : revise each cluster center μ_k to be center of points in that cluster

     $$\mu_j = \frac{\sum_{n=1}^{N} z_{nj}\, x_n}{\sum_{n=1}^{N} z_{nj}}$$

     Repeat step b if algorithm not converged

- ***Choosing Value of K***

  1) Randomly Choose some Hold out set of samples

  2) Find log likelihood value for different values of K

  3) Pick K which generates maximum log likelihood for hold out set

- ***Additional Features of GMM***

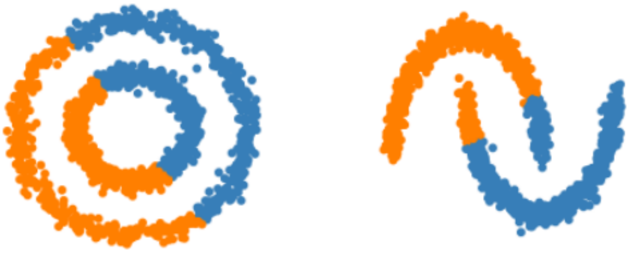
  1) GMM allows clustering with missing feature data
  One can make gaussian model from the data of feature having missing values.
  Now it can predict the missing values from the model

  2) GMM lets us generate new data with statistical properties of given data
  One can find underlying gaussian distribution from the data.
  Which can be used to generate new data having same properties.

- ***Issues with GMM***

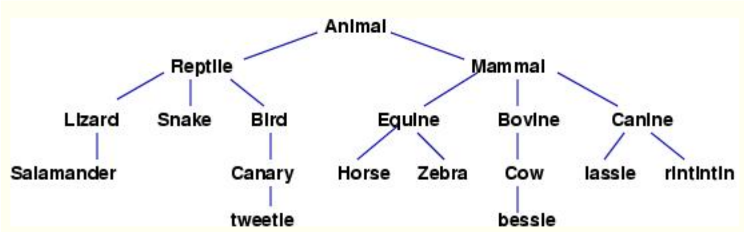
  1) In GMM we assumes data follows gaussian distributions,it may be different distributions other then gaussian.

  2) GMM dosen't work for non-convex clusters.



6 Hierarchical Clustering

- Real World data often have many features and multiple hierarchical structures. K-means and GMM can be used optimized and captures only one or few layers of it so it will not possible for these algorithm to capture these types of complex structures.Hierarchical clustring can be used in these type of scenarios.
- Also as dimension increases ,probability of points to be close decreases due to curse of dimensionality
- Idea is to Build a tree-based hierarchical structure from data.



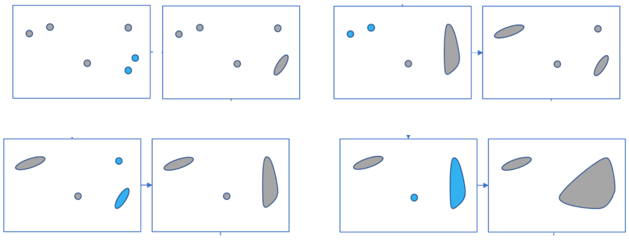
- Hierarchical clustering : 2 Types

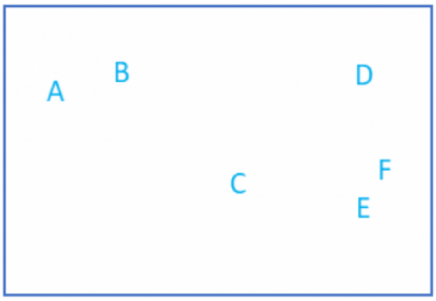
  1) Agglomerative clustering(Bottom up)
     Start with each point being a single cluster.
     Eventually all points belong to the same cluster.

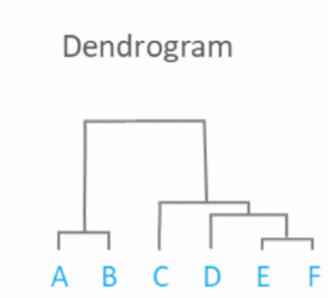
  1) Divisive (top-down)(Top Down)
     Start with all point belong to the same cluster.
     Eventually each point forms a cluster on its own

- ***Agglomerative clustering***

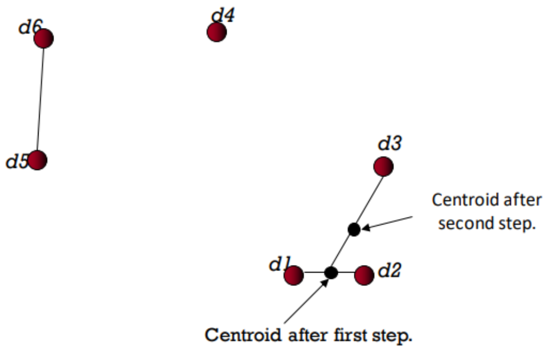
  1) Start with each points as own clusters and Merge them greedily







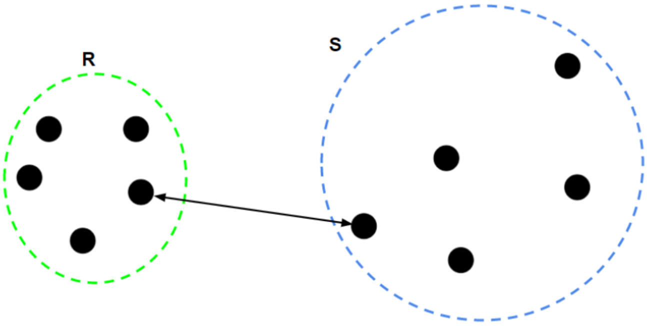
  2) Cluster Representative
     Representative should be some sort of central point in the cluster
     Ex : Centroid or center



- ***Linkage in Clusters***

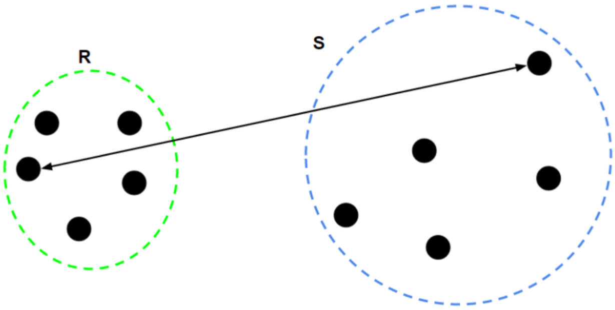
1. **Single Linkage**
   - Minimum distance between two points i and j such they are in different clusters

$$\mathrm{L(R,\ S)} = \min(\mathrm{D(i,\ j)}),\ \mathrm{i}\in R, j \in S$$



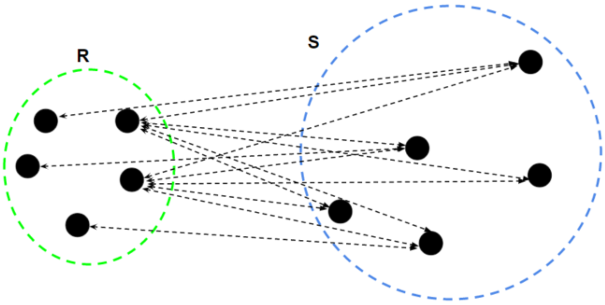
2. **Complete Linkage**
   - Maximum distance between two points i and j such that i and j belongs to different clusters.

$$\mathrm{L(R,\ S)} = \max(\mathrm{D(i,\ j)}),\ \mathrm{i}\epsilon R, j \epsilon S$$



3. **Average Linkage**
   - Arithmetic mean of all distances between all possible pairs (i,j) where i and j belongs to different clusters.

$$L(R,S) = \frac{1}{n_R+n_S} \sum_{i=1}^{n_R} \sum_{j=1}^{n_S} D(i,j), i\epsilon R, j\epsilon S$$



7 References

- Andrew NG's machine learning course. Lectures on Unsupervised Learning, k-means clustering, Mixture of Gaussians and The EM Algorithm
  http://cs229.stanford.edu/materials.html

- Haim Permuter's Machine leaning course, lecture 1
  http://www.ee.bgu.ac.il/~haimp/ml/lectures/lec2/lec2.pdf.

- Arthur Dempster, Nan Laird, and Donald Rubin (1977) Maximum Likelihood from Incomplete Data via the EM
  https://www.jstor.org/stable/2984875.

- Stephan Boyd's Convex Optimization
  https://web.stanford.edu/ boyd/cvxbook/bv_cvxbook.pdf

- Haim Permuter, Joseph Francos, Ian Jermyn, A study of Gaussian mixture models of color and texture features for image classification and segmentation
  http://www.ee.bgu.ac.il/~francos/gmm_seg_f.pdf.

- Ramesh Sridharan's Gaussian mixture models and the EM algorithm
  https://people.csail.mit.edu/rameshvs/content/gmm-em.pdf.