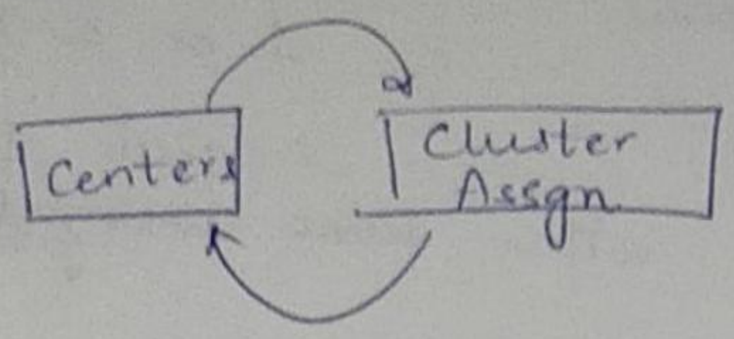
5/11/18

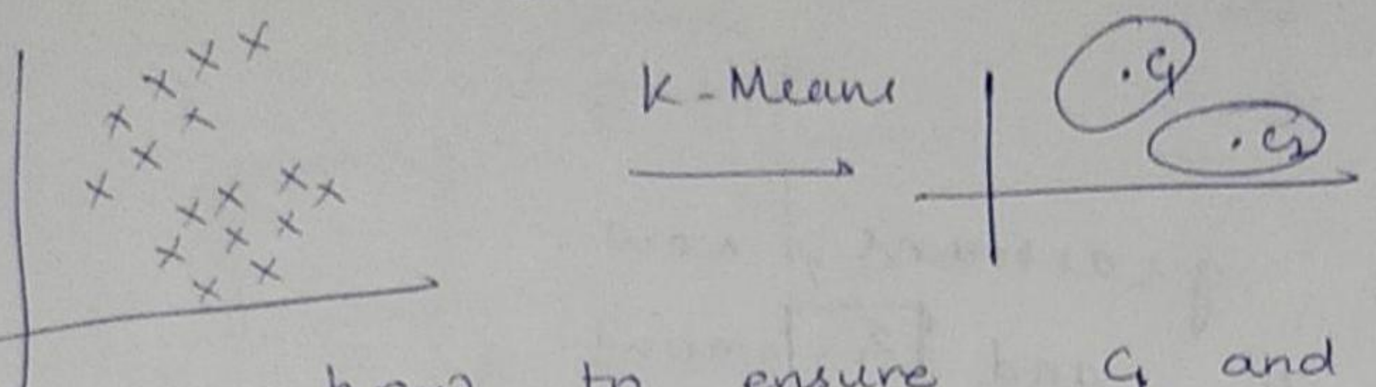
Lec-23

- **K-Means algorithm**

- we didn't know the clusters and the centers

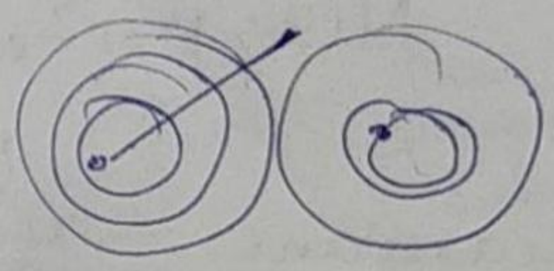


K-Means

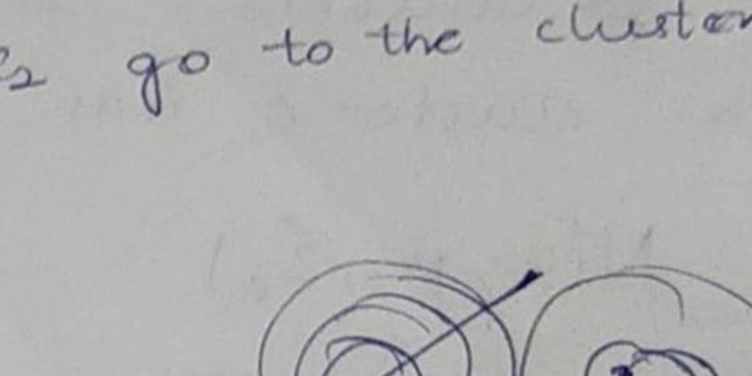


K-Means, how to ensure C_1 and C_2 go to the cluster center

- Circles/Spheres are produced, if we consider the euclidean distance b/w means



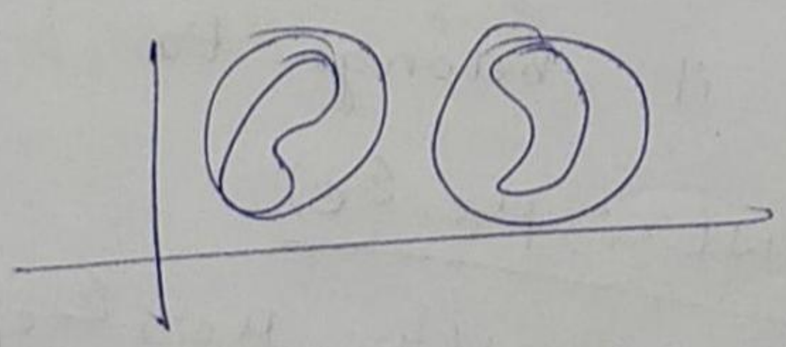
- So, K-means, won't be good, if points deviate from the cluster // shape's are non spherical.

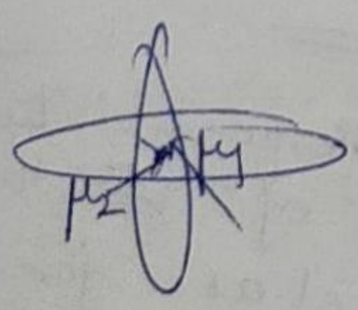


Doesn't think about these gaps //

↓

Shape = Spherical is not the exact condition, // Boundary should be seperable by spheres





→ K-means can't handle it.

Soln ⇒ Make 20 clusters and then combine 'em

⇒ Mahalonobis distance

$\sqrt{\sum_i (x_i - y_i)^2}$ → Euclidean

$\sqrt{(x-\mu)^T (x-\mu)}$ → weighted distances (using covariance matrix)

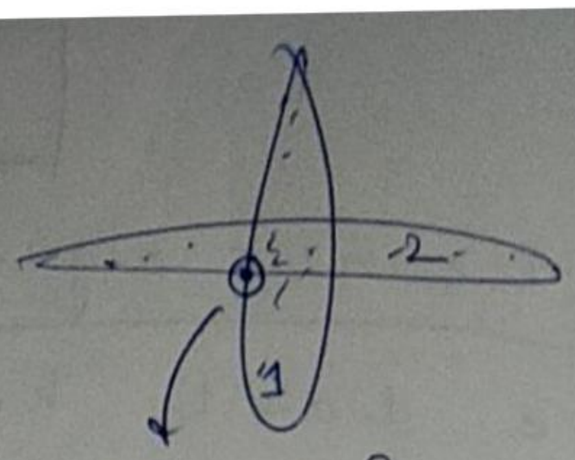
$(x-\mu)^T \Sigma^{-1} (x-\mu)$

↓

Covariance matrix of a cluster.

- Euclidean dist → $\Sigma = \sigma^2 I$

But, now: Σ = generic, Bad
↳ Ellipsoidal / Guassian density



- Ambiguity — at intersection?

Soft assignment.

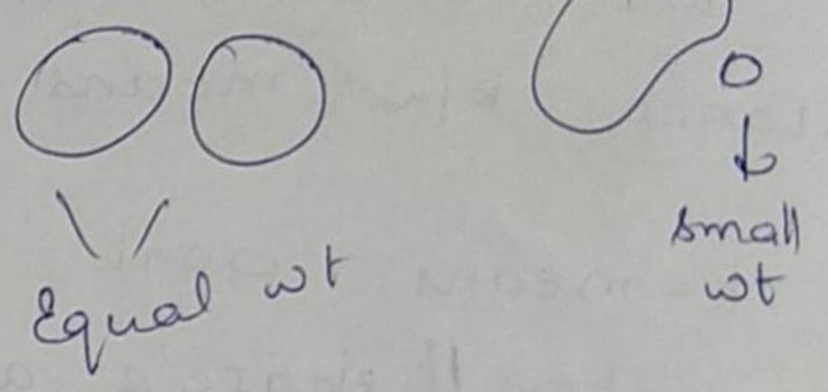
Every point is associated with some ~~weight~~ prob of belonging to cluster i, using

So, Clusters are guassians, now.

Each cluster c, has μ_c and Σ_c now

$N(x; \mu_c, \Sigma_c)$

- Guassians: weighted (π_c) α Number of points Belonging to the guassian



$N(x, \mu_c, \Sigma_c, \pi_c)$

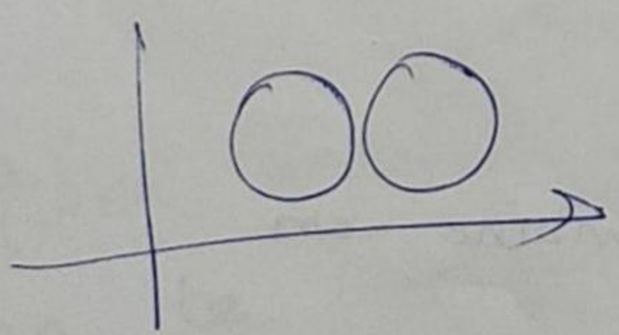
↓

(Cluster center)

(Cluster info)

To find which cluster it belongs to

~~N_1, N_2~~ Likelihood ~~Prob~~ of class → $\frac{\pi_c N(x_i, \mu_c, \Sigma_c)}{\sum_j \pi_j N(x_i, \mu_j, \Sigma_{cj})} = \gamma_{ic}$ =

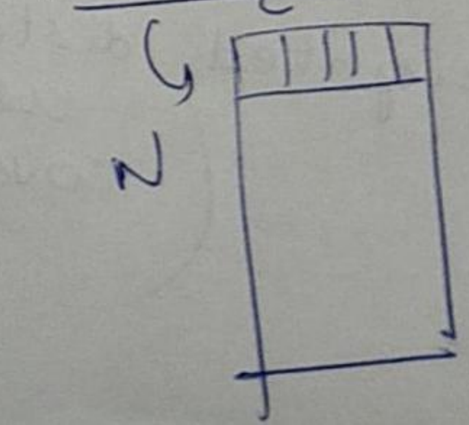


Responsibility of ~~i~~ c^{th} class for point i.

The cluster assignment (percentage)

Nxc matrix

1 Matrix



i^{th} row has all the likelihoods

Sum of partial assignments to a cluster = total assignments

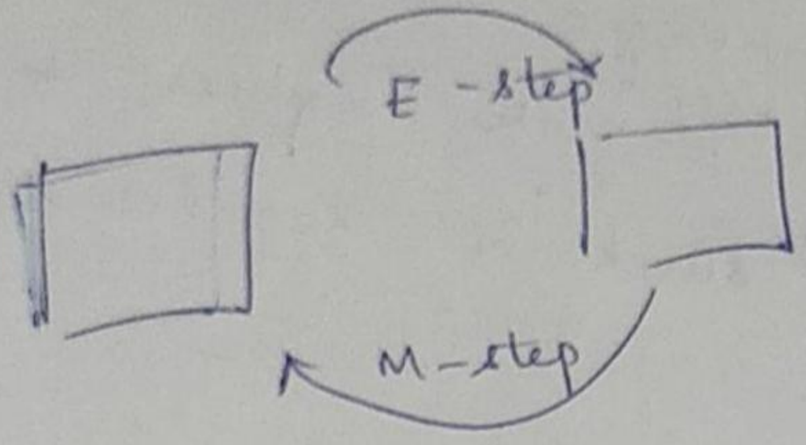
$m_c = \sum_i r_{ic}$

$\pi_c = \frac{m_c}{m} = \sum m_c$

$\mu_c' = \frac{\sum r_{ic} \times x_i}{\text{weights}} = \frac{\sum r_{ic} \times x_i}{\sum r_{ic}} = \frac{\sum r_{ic} x_i}{m_c}$

$\Sigma_c' = \frac{1}{m_c} \sum (x_i - \mu_c')^T (x_i - \mu_i)$

$\Sigma', \mu' \rightarrow$ To denote new μ, Σ

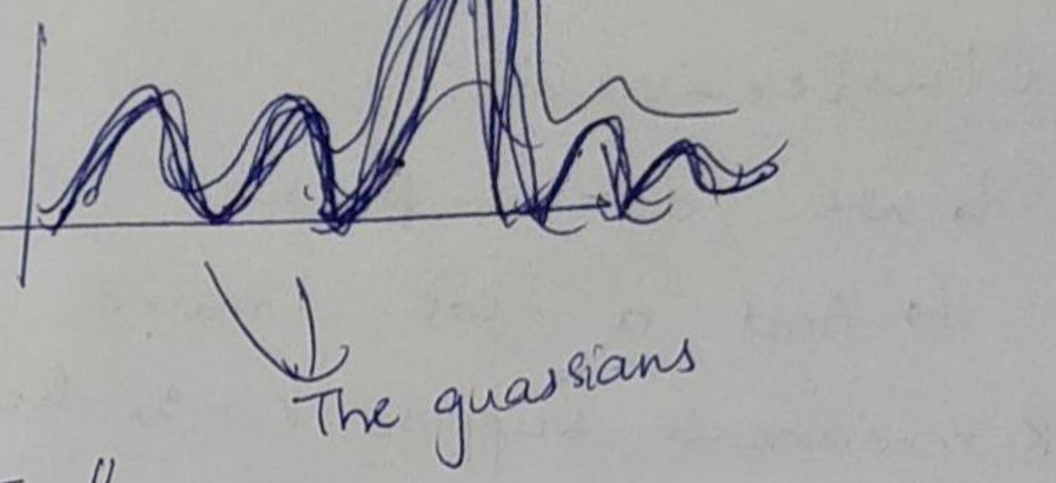


EM - algorithm

K-means is an EM-algorithm

- Guassian mixture model

It is a parameter estimation method MLE



Now π_c $\mu_c' = \frac{1}{m_c} \sum r_{ic} x_i$, μ (mean of data)

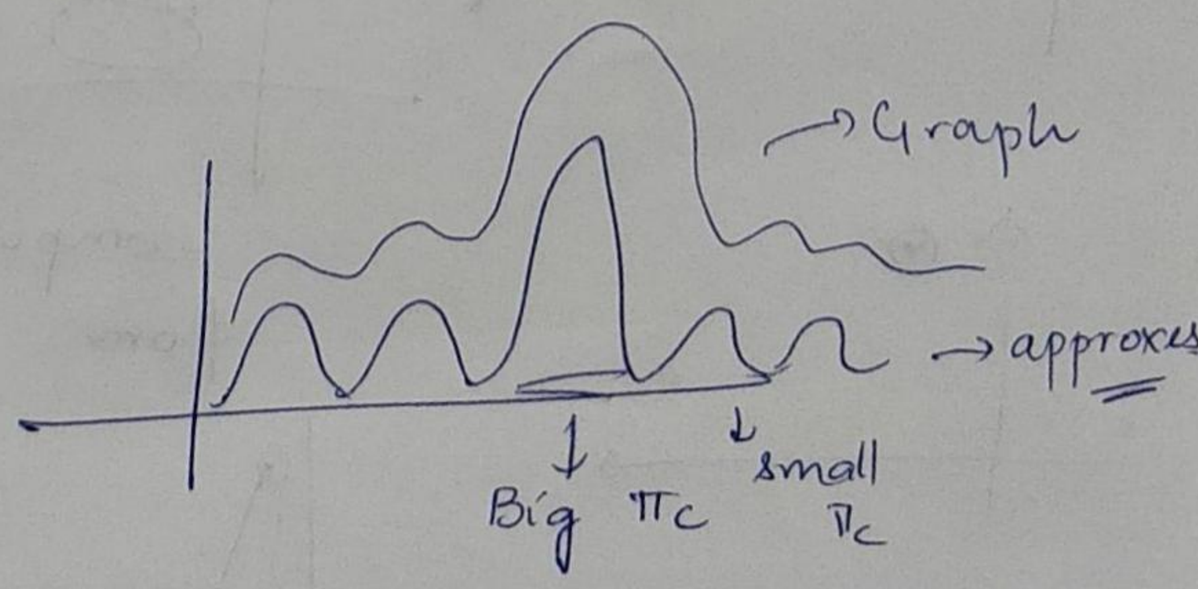
Σ_c' (var of data)

That is why E → π is estimation and

M →

EM - Problem: Previously, we talked abt a class of probs, A ⟳ B, A, B unknown, now a more general case

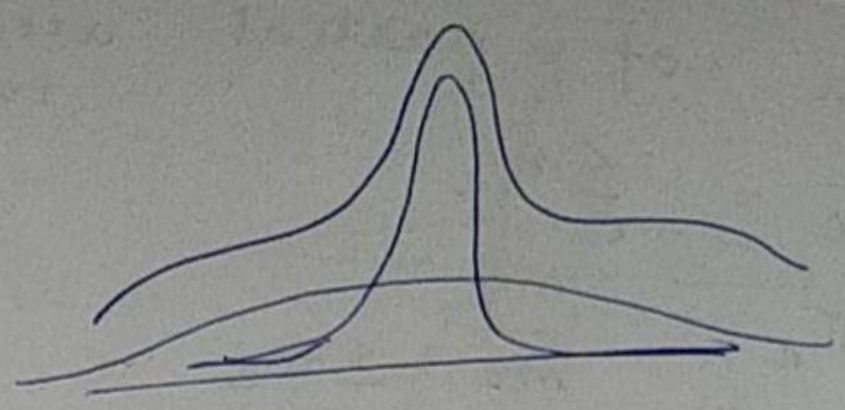
z is ~~unknown~~ the latent variable (r_{ic}), so that knowing latent variable can help us solve the prob & vice-versa



also
Π is a the number of points described by the guassian.

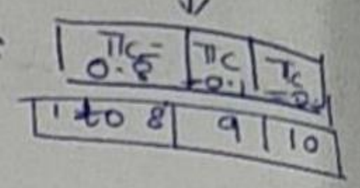
Thinly spread $\rightarrow \Pi_c$ less

Conc $\longrightarrow$ Huge Π_c



I can generate samples now, using, first 1 random no for the cluster number. (proportional to Π_c)

and then random number from the guassian.



eg:

- EM is guaranteed to converge, it won't stop, but it will reach very small values, so u
- K means generally doesn't get stuck in local-minima, but E-M can get stuck

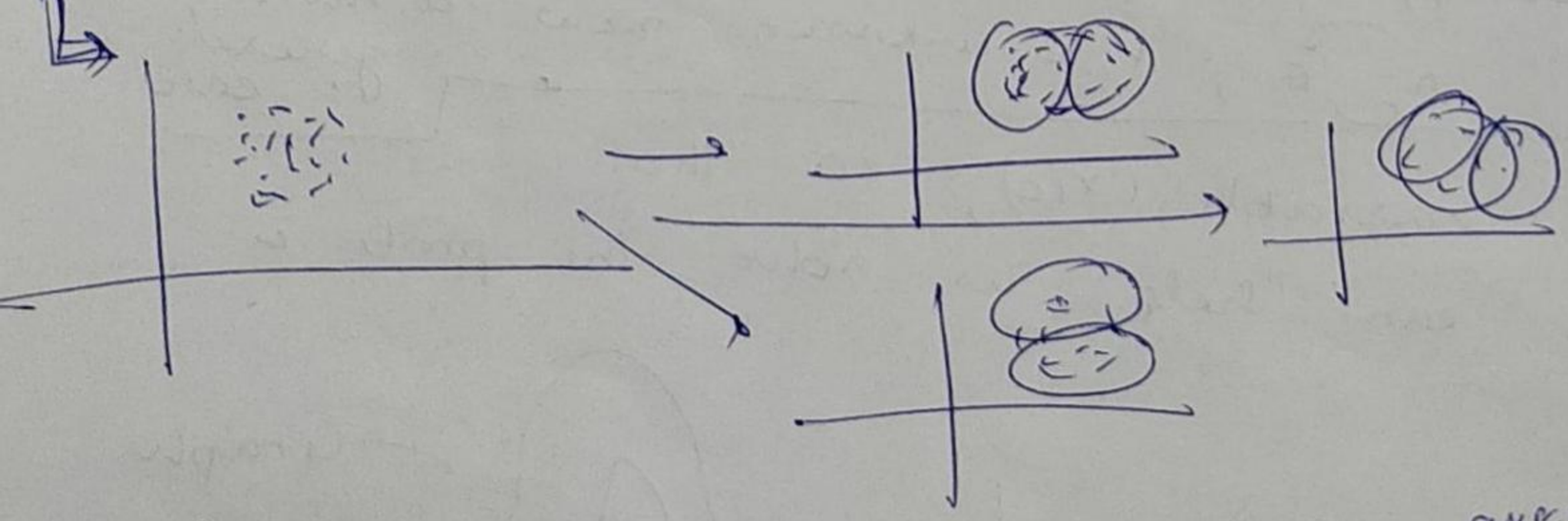
// EM is like dynamic programming an idea → Just a class of algos //

- Cluster-validity:-
  - ↳ we don't know, how many clusters to choose?
  - ↳ And a lot more (means, Σ, —)

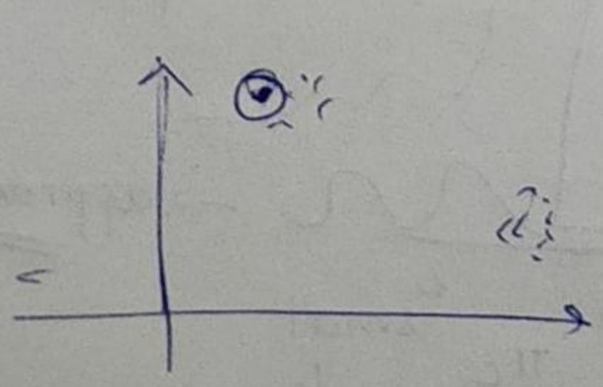
K-means → suppose $k = 3$, how to we know the right number of clusters.

① Stability
② } Ways of measuring validity.

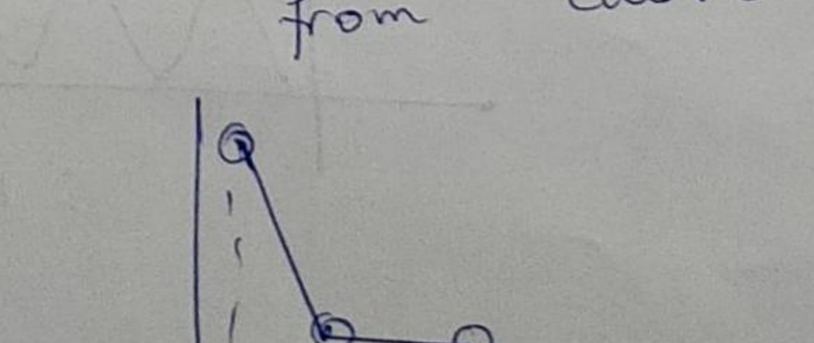
→ Initialize with different values, do all, lead to the same cluster, How much % give same result



Unstable along different initializations.



Compute averaged distance from cluter centres



- ② Mean - squared distance to cluster center

↳ other similar metrix too

↳ For GMM, you can use ric, related stuff

- Some give good graphs, to find minima