Gaussian Mixture Models

A Mixture model is a probabilistic Model which assumes the underlying data to belong to a certain distribution

Gaussian Mixture Models assume a Gaussian distribution.

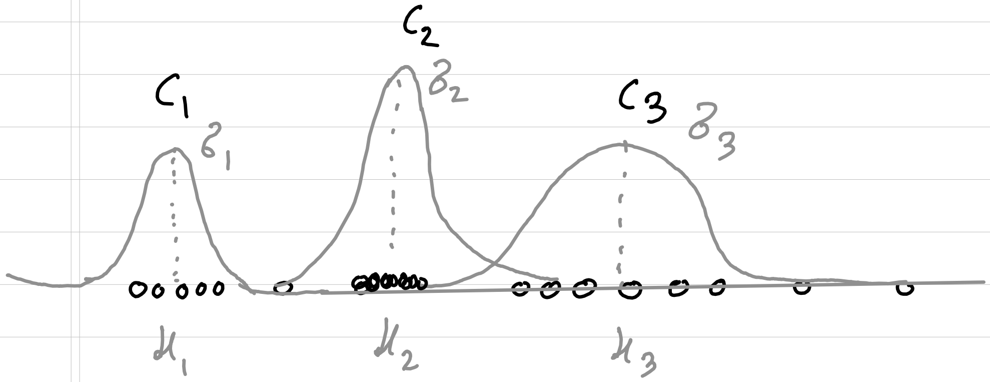
K-Means classify the points based on the nearest neighbour while GMMs use a probabilistic approach

K-Means is hard clustering telling which datapoint should belong to which cluster and only one cluster.

However, GMM tells us probability of the data being in a cluster, so the points can be in two clusters as well

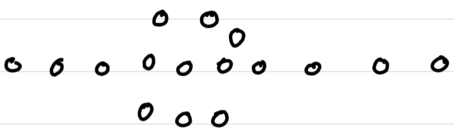
GMM assumes that every cluster has points around it in a gaussian distribution.

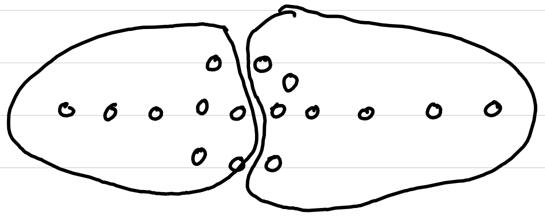
eg for 1D data like this



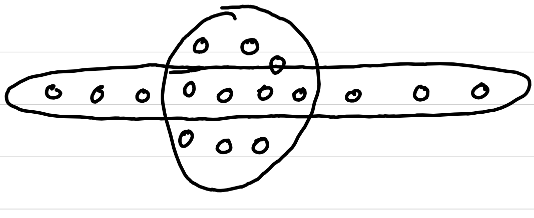
every cluster has its own mean μ and standard deviation σ

GMM can handle overlapping clusters very well





K-Means



GMM

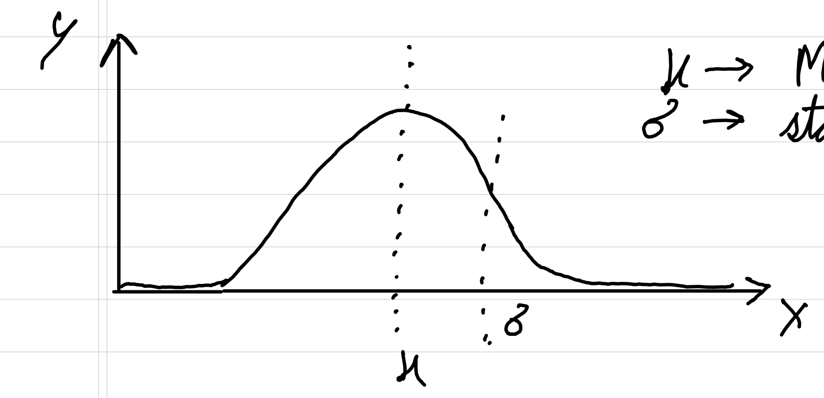
Points can belong to both clusters at same time

Multivariate Normal Distribution

for a 1D case, we have

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

which gives us the graph of a Normal distribution



$\mu \rightarrow$ Mean
$\sigma \rightarrow$ standard deviation

when we extend this to nD, we get

$$f(X) = \frac{1}{(2\pi)^{d/2}\,|\Sigma|^{1/2}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)}$$

3.14159...

d = no of dimentions

Covariance Matrix

determinant

Covariance Matrix

Mean

x data point

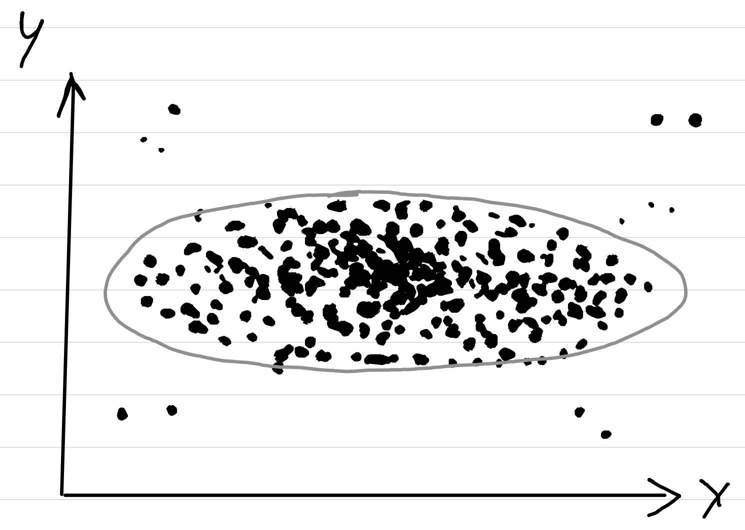
eg

$$\Sigma_{2D} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$

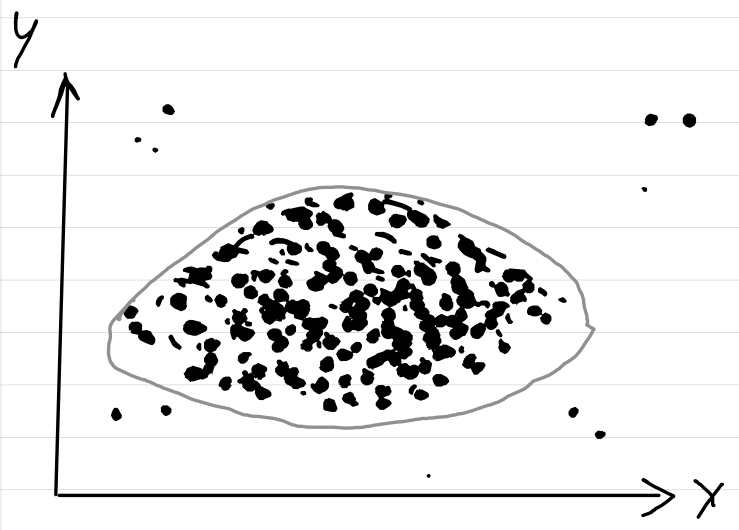
By controlling the values of Σ & μ vectors, we can change the shape of the distribution

eg $\rightarrow$

If $\sigma_{11} = +6.00$, $\sigma_{12} = 0$, $\sigma_{21} = 0$, $\sigma_{22} = +1.00$
$\mu_1 = 5$ $\mu_2 = 5$
We get distribution as follows $\rightarrow$

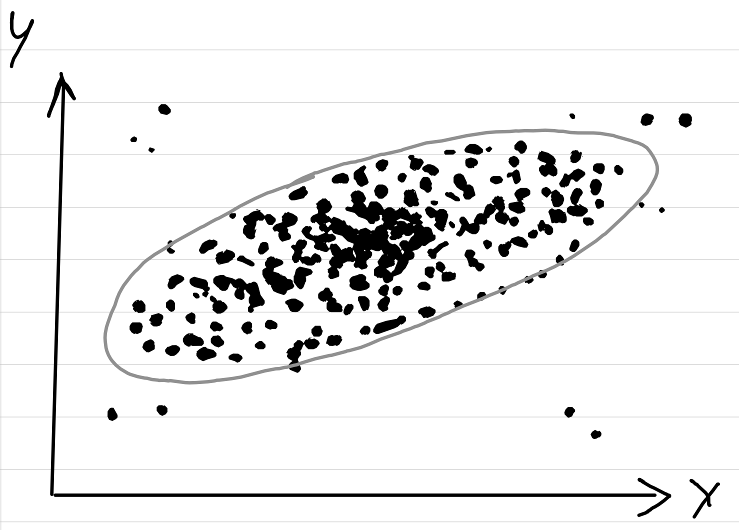


If $\sigma_{11} = +1.00$, $\sigma_{12} = 0$, $\sigma_{21} = 0$, $\sigma_{22} = \cdot 3$
$\mu_1 = 5$ $\mu_2 = 5$
We get distribution as follows $\rightarrow$



The graph buldged because σ_{22} was increased

If $\sigma_{11} = +6.00$, $\sigma_{12} \neq 0$, $\sigma_{21} \neq 0$, $\sigma_{22} = +1.00$
$\mu_1 = 5$ $\mu_2 = 5$
We get distribution as follows $\rightarrow$



The graph rotated cus $\sigma_{12} \neq 0$
$\sigma_{21} \neq 0$

If σ_{12}, σ_{12} are 0 then the graph will bulge along the horizontal or vertical axis only.

σ_{12} $\sigma_{21} = 0$ means that there is no linear corellation in between the axes

If σ_{12} or $\sigma_{21} \neq 0$ then the graph is tilted

This indicates that changes in one dimention are associated with changes in other dimention

Mixing Coeficients

Mixing coeficient is used to decide how small or big the gaussian will be

They are denoted by π_k (density)

More π_k, more important is the gaussian.

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

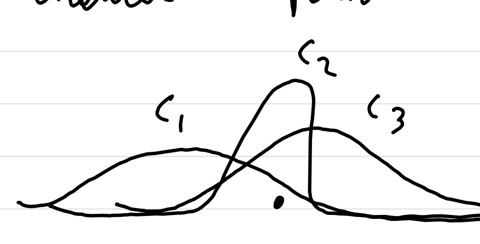
ie. π_k is the probability that a point is assigned a cluster z_k

$$\pi = \frac{\text{No of Points assigned to cluster}}{\text{Total No of Points}}$$

Hence $\sum \pi_k = 1$

Responsibilities

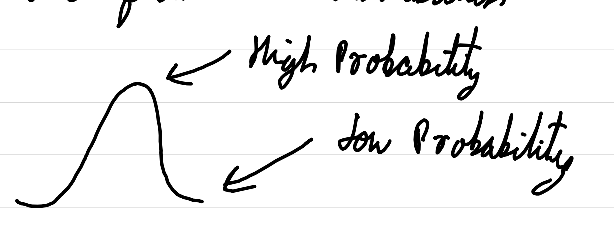
Consider a point P



To which cluster will it belong?

Now consider from perspective of c_1

from gaussian distribution, we can calculate the probability that a point comes from a distribution



from c_1 Probability that P belongs to c_1 is

$$= \Pi_1 N(x_i ; \mu_c \Sigma_c) = \Pi_1 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

(or for nd ....)

We have Π_1 as the importance of the cluster

More important clusters will have higher Π_x & thus higher probability.

Think of this value as amount of which c_1 pulls P_i towards itself

from perspective of c_2

Probability P in c_2 = $\Pi_2 N(x_i ; u_c \Sigma_c)$

Similarly from c_3 ....

But from perspective of P the probability that it will be in c_1 depends not only on c_1's perspective but also c_1, c_2, c_3 etc perspective

That is a weighted sum of all pulls.

That is for a point P, the probability that it belongs to a cluster is

$$\gamma_{ic} = \frac{\text{Probability } P_i \text{ belongs to } c_1}{\text{Sum of Probability that } x_i \text{ belongs to } c_1, c_2 \,\&\, c_3}$$

Example

$$\left.\begin{aligned} P(x_i = c_1) &= 0.8 \\ P(x_i = c_2) &= 0.6 \\ P(x_i = c_3) &= 0.4 \end{aligned}\right\} \text{Pull} \qquad \begin{aligned} \Pi_1 &= 0.4 \\ \Pi_2 &= 0.3 \\ \Pi_3 &= 0.2 \end{aligned}$$

Then $\gamma_{ic} = \dfrac{0.8 \times 0.4}{0.8\times0.4 + 0.6\times0.3 + 0.4\times0.2}$ = ☺

The initial values are just the pulls while γ_{ic} is the total probability

γ_{ic} is used for classification

This can be proved by the bayes theorem

$$P(A_j | B) = \frac{P(B|A_j)P(A_j)}{\sum P(B|A_j)\cdot P(A_j)} = \frac{P(A_j \cap B)}{P(B)}$$

$P(\text{class } j \mid \text{Point } P)$ ie what is the probability that class is j when point P is given

$P(\text{Point } P \mid \text{class } j)$ cluster's perspective

that is probability of point being in cluster j when cluster j is given

$P(\text{class } j)$ = Probability a point lies in class j

$= \Pi_x$

$$P(\text{class } j \mid \text{Point } P) = \frac{P(\text{Point } P \mid \text{class } j)\,\Pi_x}{\sum P(\text{Point } P \mid \text{class } j)\,\Pi_j}$$

γ will be higher when assigned correct cluster & lower otherwise

γ is called responsibility

EM based learning

Once we have π, μ, Σ, we can calculate clusters

Once we have clusters, we can calculate the values of π, μ & Σ

To solve the chicken & egg problem, EM based learning is used.

$$\begin{tikzcd}
\pi, \mu, \Sigma \arrow[rr, bend left, "\text{Bayes theorem}"] & & \gamma_{ij} \arrow[ll, bend left, "\text{Maximum likelihood estimation}"]
\end{tikzcd}$$

Initialization of clusters can be done using k-means algorithm

E step: Responsibility γ_{ij} is calculated

Gives the probability of each cluster for each data point

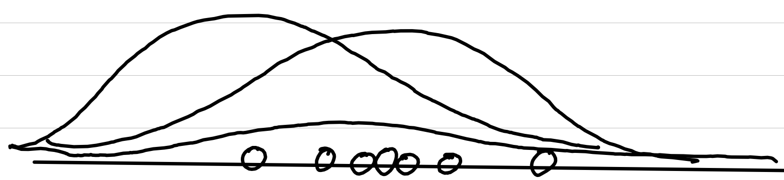
M step: Parameters (π, μ, Σ) are updated each step based on the responsibilities calculated in E step.

These two steps are repeated until convergence is reached

Maximum Likelyhood Estimation

Suppose we have datapoints like these

Then which Normal distribution will be the best fit?



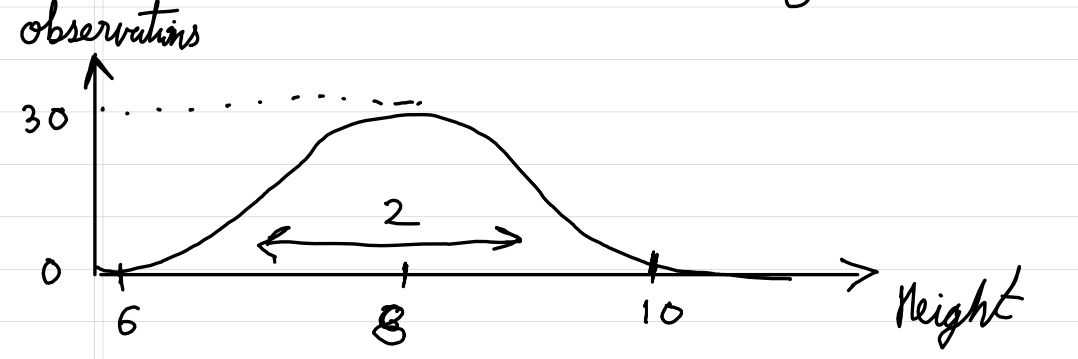
There are ∞ such distributions possible

But we want the distribution with the maximum likelyhood.

That is choose a distribution such that the points have maximum probability of comming from it.

Probability vs likelyhood

Consider counting height of 1000 people
Most people have height around 8
we get a normal distribution with
$\mu = 8$
$\sigma = 2$

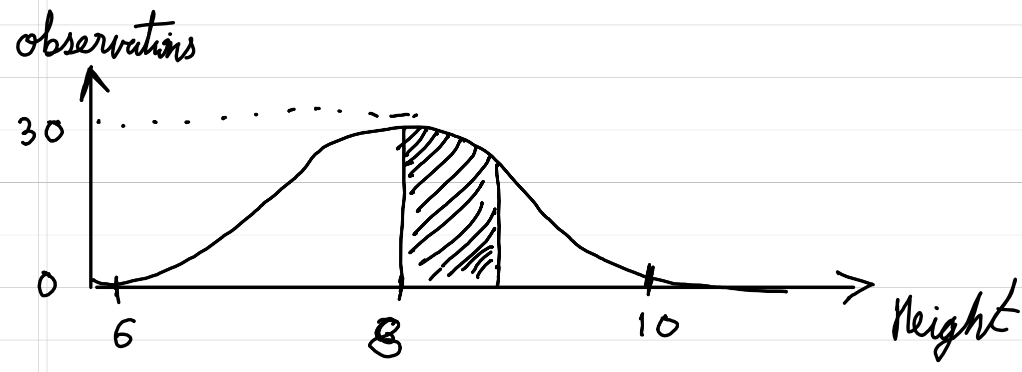


μ is the mean of the samples
σ is the calculated standard deviation

Probability that a person weighs between 8-9 is

$$P(\text{weight between } 8\text{-}9 \mid \overset{\text{given}}{\mu = 8,\ \sigma = 2}) =$$

Area under the curve



Likelyhood is when a data point is known and μ & ε are not known.
We want to find the probability that the point comes from that distribution

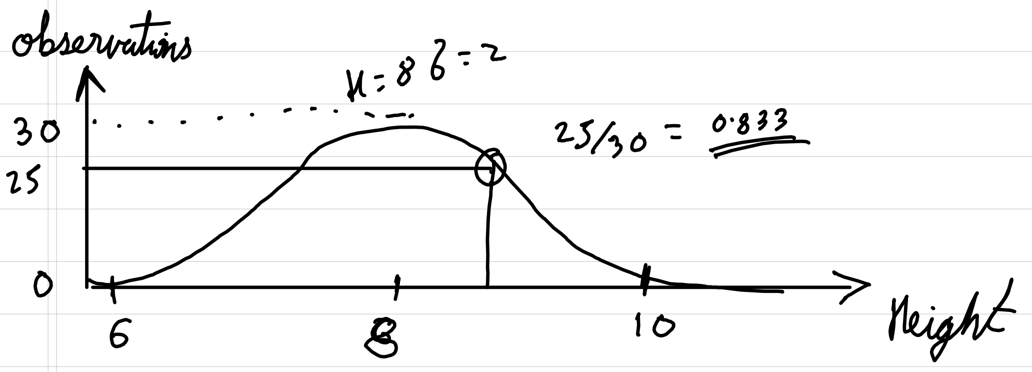
eg we measured a random person and found Height as 8.5

what is the probability that the person has come from this distribution with mean = 8 & ε = 2

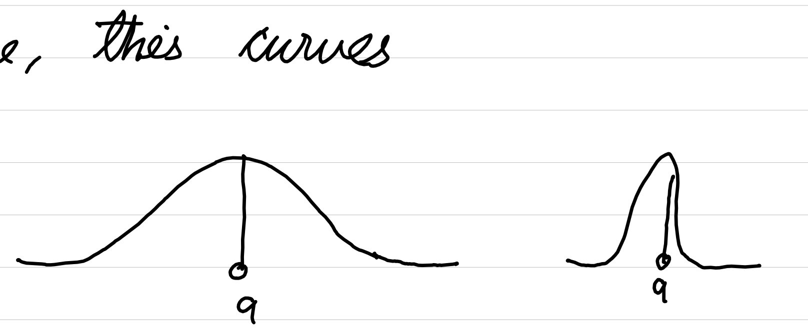
ie.

$$P(\mu = 8, \sigma = 2 \mid \text{height} = 8.5) =$$

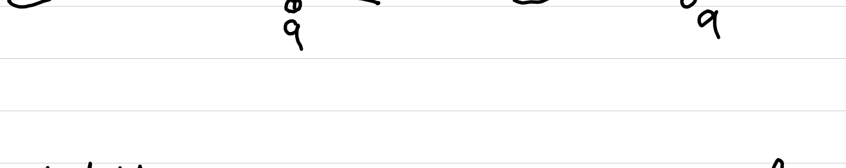
y coordinate of the point



Hence, this curves



Has more likelyhood than these



Probabilities are the areas under a fixed distribution

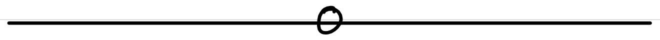
$$P(\text{data} \mid \text{distribution})$$

Likelyhoods are y axis values for fixed data points with distributions that can be moved.

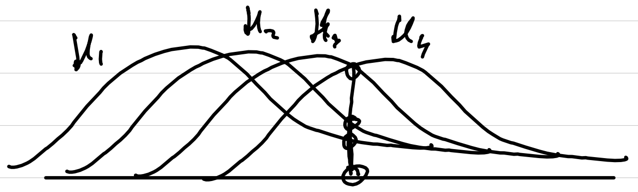
$$P(\text{distribution} \mid \text{data})$$

MLE

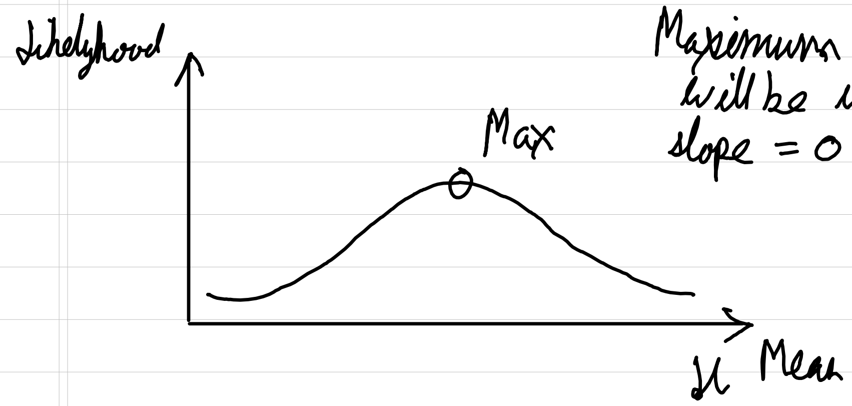
Consider a point to be fitted in a Normal distribution



Keep a constant value of σ & change μ



We calculate the likelyhood everytime & plot a graph.



Maximum value will be when slope = 0

Similarly we keep μ constant at its maximum value $\hat{\mu}$
We can find the maximum by taking the derivative

μ & σ are not independant, but we can find $\hat{\mu}$ by fixing value of σ & once we find $\hat{\mu}$ we can use it to find $\hat{\sigma}$

This is because $\mu = \frac{1}{n} \Sigma x_i$

while $\sigma = \frac{1}{n} \Sigma (x - \mu)$

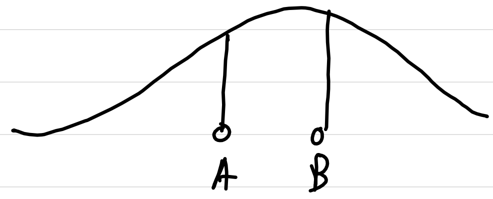
Hence σ depends on μ while μ does not depend on σ

Finally choose the μ & Σ with the maximum values.

We actually are taking partial derivatives of μ and σ. When taking partial derivative of σ, put $\mu = \hat{\mu}$

MLE

for two points



The likelyhood is $P(\mu, \sigma \mid A)$ for A
$P(\mu, \sigma \mid B)$ for B

Since A & B are independant

$$P(\mu, \sigma \mid A, B) = P(\mu, \sigma \mid A) \times P(\mu, \sigma \mid B)$$

ie we can write it as likelyhood

$$L(\mu, \sigma \mid x_1, x_2 \ldots x_n) = \prod_i L(\mu, \sigma \mid x_i)$$

We can now calculate μ & σ in similar way as one point

We need two different derivatives

first: Partial derivative treating σ as constant and setting to 0 to find $\hat{\mu}$

Second: Partial derivative treating $\mu = \hat{\mu}$ & setting value to 0 to find $\hat{\sigma}$

Goal is to Maximize $L(\mu, \sigma \mid x_1, x_2 \ldots x_n)$ by finding $\hat{\mu}$ & $\hat{\sigma}$ variables

In order to take derivative we take the log on both sides

$$\log_e(L(\mu, \sigma \mid x_1, x_2 \ldots x_n)) = \sum_i \log_e(L(\mu, \sigma, x_i))$$

wherever the likelyhood function peeks, the log function peaks too.

Hence instead of Maximizing L, we maximize $\log(L)$

The values of $\hat{\mu}$ & $\hat{\sigma}$ wont change

Mathematics of MLE

$$L(\mu,\sigma) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

$$l(\mu,\sigma) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

$$\frac{\partial l}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu)$$

at $\frac{\partial l}{\partial \mu} = 0$, we get $\boxed{\mu = \frac{1}{n}\sum_{i=1}^{n} x_i}$

$$\frac{\partial l}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}(x_i-\mu)^2$$

at $\frac{\partial l}{\partial \sigma} = 0$

$$\boxed{\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2}$$

In the end we get μ = mean
σ = standard deviation

That is this is proof that in a normal distribution μ is the mean while σ is the standard deviation indeed.

Maximization step

π = Propability of cluster = $\frac{\text{No of points assigned to cluster}}{\text{Total no of points}}$

Since we are in soft clustering scenario Points are not assigned to cluster.

$$\pi_c = \frac{\sum_{i=1}^{m} \gamma_{ic}}{m}$$

$\sum_{i=1}^{m} \gamma_{ic}$ ← Points assigned weighted

m ← Total no of points

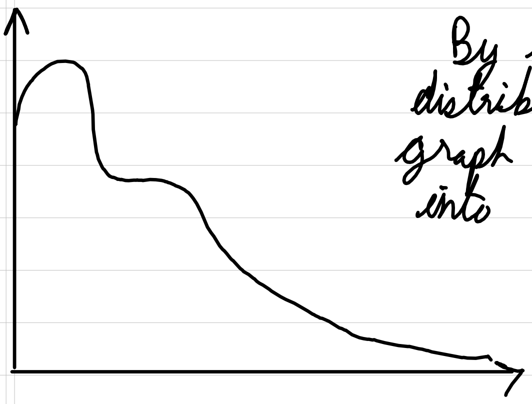
$$\mu = \frac{\sum \gamma_{ic} \, x_i}{\sum_{i=1}^{m} \gamma_{ic}}$$

Mean of cluster

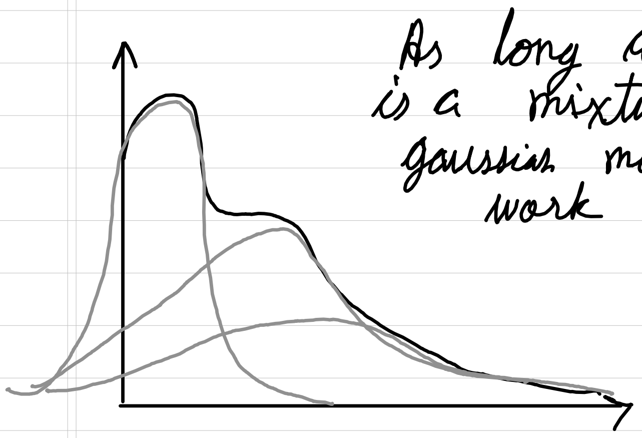
$$\Sigma_c = \frac{\sum \gamma_{ic} \, (x_i - \mu_c)^T (x_i - \mu_c)}{\sum_{i=1}^{m} \gamma_{ic}}$$

Non Gaussian Data

GMM can classify data, as long as it is a weighted sum of gaussian distributions

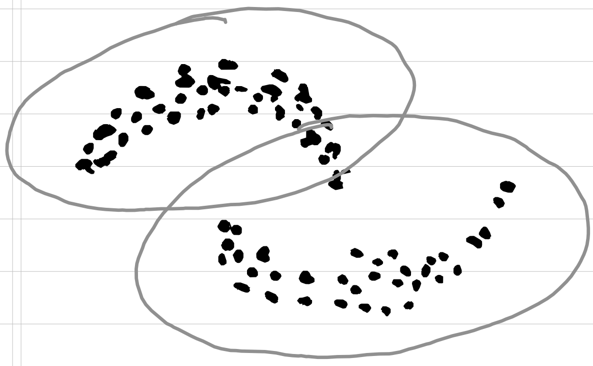


By increasing the No of distributions, the graph can be decomposed into gaussian data

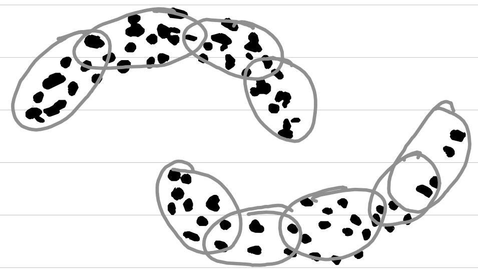


As long as the data is a mixture of several gaussian models, GMM can work

Non Gaussian Data



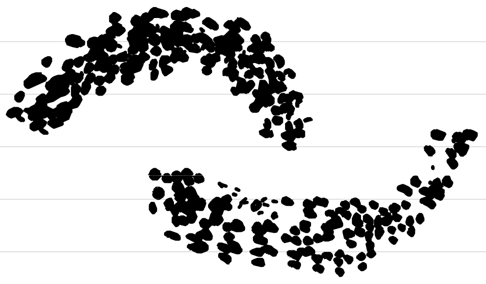
$n = 2$



n = large

This may not be useful for clustering but is useful for generation

Once we know the distribution GMM can augment more datapoints



Model has learnt the overall distribution of the data which it can reproduce

GMM as generative Models

GMM can create new datapoints that are similar to the sample

for a cluster, we just have to generate a random point that follows the Normal distribution of the cluster

This can be used for data augmentation

GMM can be used to generate Images as well

MNIST

↓ 64 dim

PCA

↓ 41 dim

GMM Train

↓

GMM Generate

↓ 41 dim

IPCA

↓ 64 dim

Image

Types of Covariance

① Full Covariance

The ideal scenerio, Σ is

$$\Sigma_k = \begin{pmatrix} \sigma_{k11} & \sigma_{k12} & \cdots & \sigma_{k1d} \\ \sigma_{k21} & \sigma_{k22} & \cdots & \sigma_{k2d} \\ \vdots & & & \\ \sigma_{kd1} & \sigma_{kd2} & \cdots & \sigma_{kdd} \end{pmatrix}$$

for every cluster k

This is the actual defination of the covariance Σ, the ideal case.

Full covariance is computationally expensive but gives best results.

Time complexity : $d \times d \times k$

② Diagonal Covariance

To reduce the complexity we assume that the Non diagonal elements are 0

ie $\Sigma_k = \begin{pmatrix} \sigma_{k11} & 0 & \cdots & 0 \\ 0 & \sigma_{k22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \sigma_{kdd} \end{pmatrix}$

$$\Sigma_k = \text{diag}\left[\sigma_{k11} \; \sigma_{k22} \cdots \sigma_{kdd}\right]$$

σ_{kij} is the covariance between i^{th} & j^{th} dimention

if $i=j$, it becomes standard deviation

$$\Sigma_k = \text{diag}\left[\sigma_1^2 \; \sigma_2^2 \; \sigma_3^2 \cdots\right]$$

$\sigma_i^2 \rightarrow$ variance

This means that we are assuming

$\sigma_{ij} = 0$ if $i \neq j$

that is Uncorrelated dimentions

Hence the datapoints will form clusters that are ellipses aligned with the coordinate axis

Time Complexity : $d \times k$

Suitable when clusters have different spreads along each dimention but there are no correlations between dimentions

③ Spherical Covariance

All the σ values accross every dimention are the same.

Results in spherical clusters with same spread

$$\Sigma_k = \sigma^2 I$$

Time complexity : k

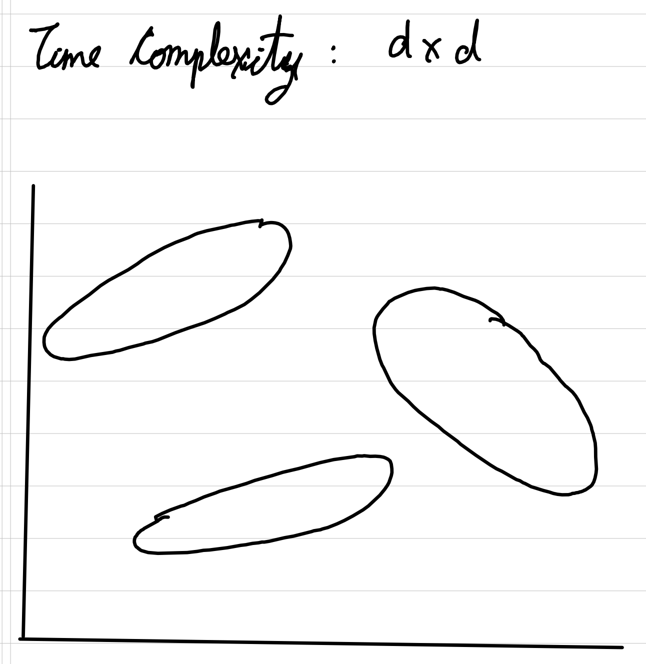
④ Tied Covariance

All the clusters have same covariance

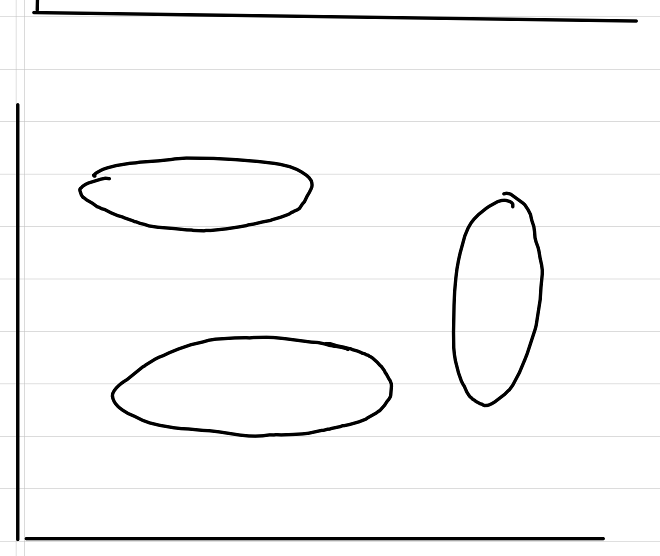
Calculate only 1 covariance matrix and use it for all clusters

Same shapes for all clusters., means may be different

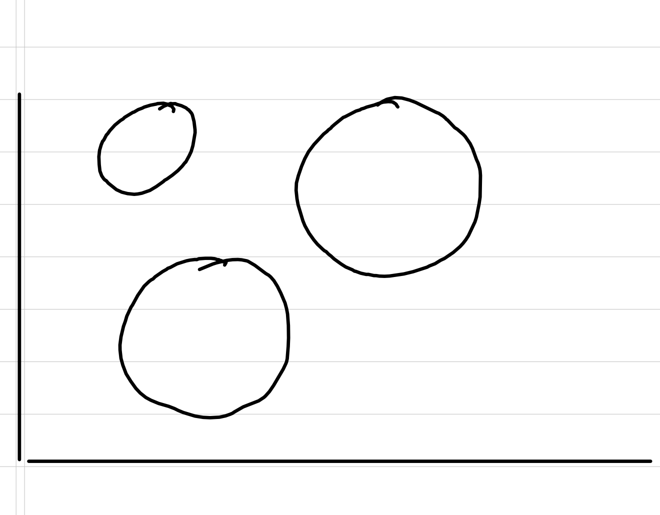
Time Complexity : $d \times d$



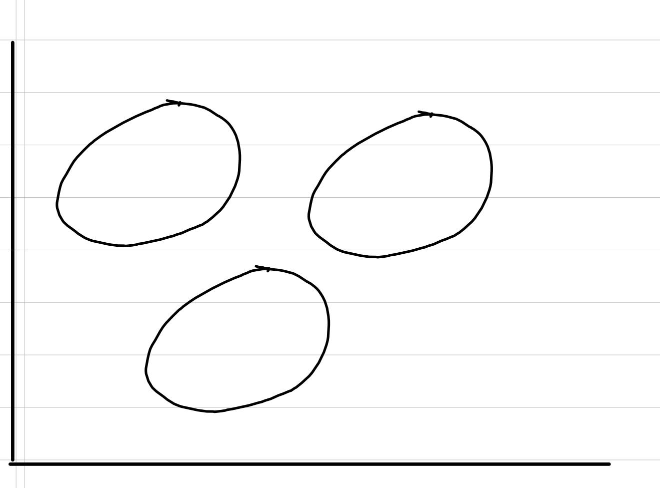
Full



Diagonal



Spherical



Tied

Prevention of overfitting

① AIC (Akaike Information Criterion)

$$AIC = 2K - 2\ln(L)$$

No of Parameters

likelyhood

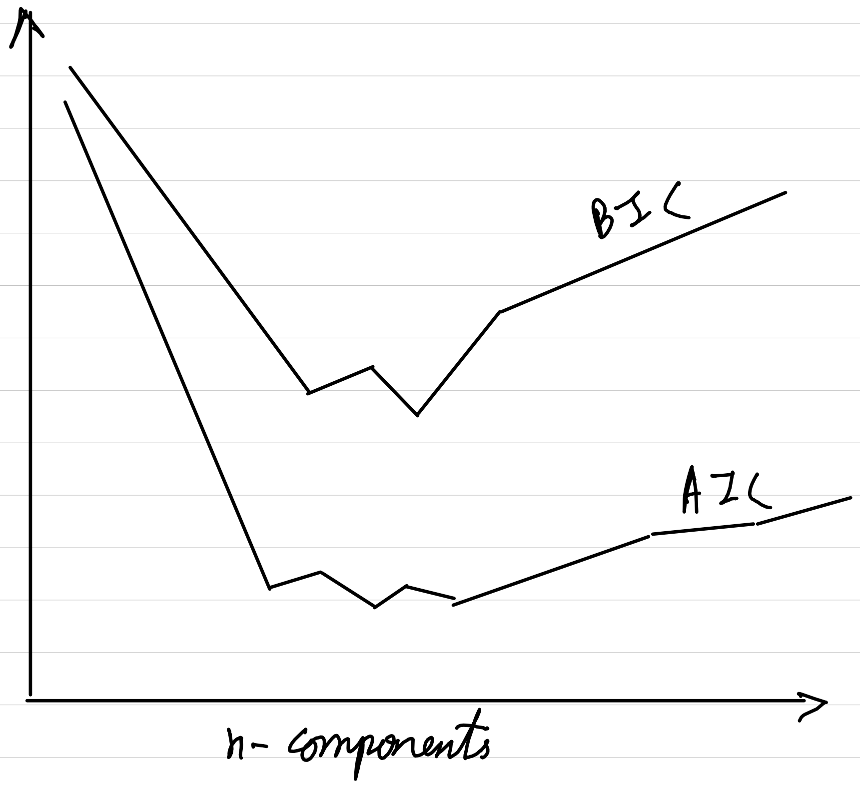
② BIC (Bayesian Information Criterion)

$$BIC = \ln(n)K - 2\ln(L)$$

No of datapoints

More Penalty for complex Models

for GMM, K means total no of parameters (means, covariances, mixing coeficiants)



Helpful to determine when to stop

Advantages of GMM →

Flexibility → Can Model complex distributions
Wide range of shapes apart from spherical & elliptical

Overlapping clusters handled

Works good for Nested clusters

Soft clustering

Robust to outliers

Scalability to large datasets

Disadvantages →

Gaussian assumption

Determining No of components can be challenging

Computationally intensive for large or higher dimentional data

Sensative to initial values

Convergence issues for E-M

Overfitting of data

Applications →

Speech recognition

Image segmentation

Anamoly detection

Clustering

Medical Image analysis

Text clustering

Recommendation Networks