DBSCAN & GMM

[Clustering]

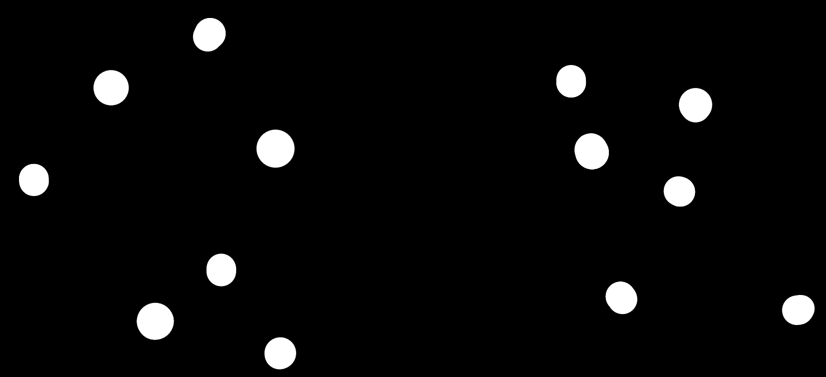
→ 2 new ideas !!

→ Comparision of algorithms

ML-2 is a good place to see how we think of algorithms. For the same problem we have seen 2 ideas:

Kmeans: Distance from centriods

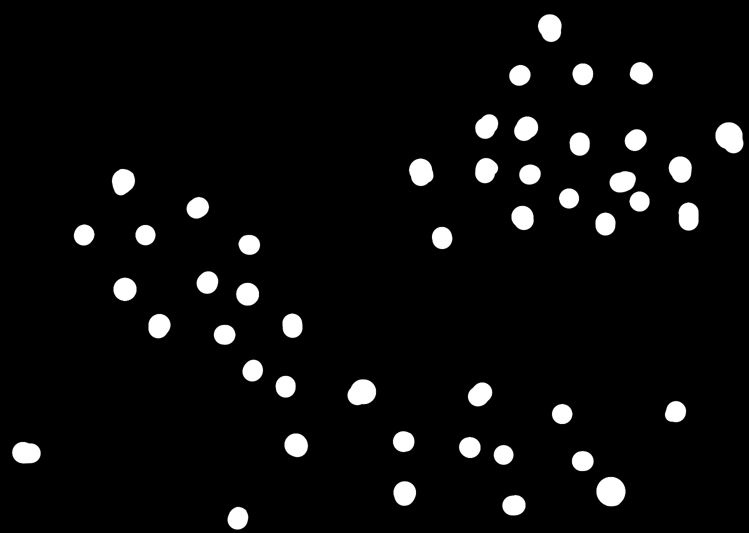
Hierarchical: Join closest points ...



Let's look at 2 more ideas today.

DBSCAN → 3rd big idea

Density **b**ased **s**pacial **c**lustering **a**pplication with **n**oise.



Q: How many clusters do you see?

a) 1

b) 2

c) 3

d) 4

Q: Do you think KMeans will work?

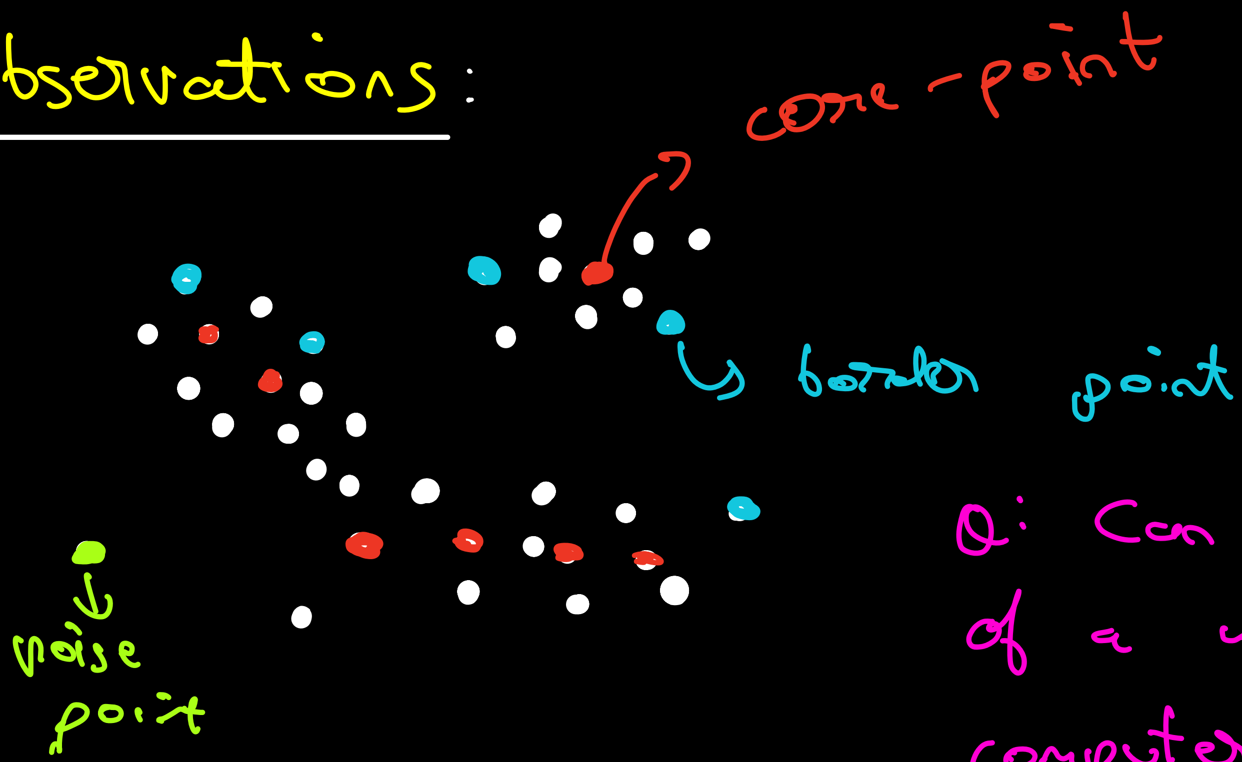
↳ No

Idea:

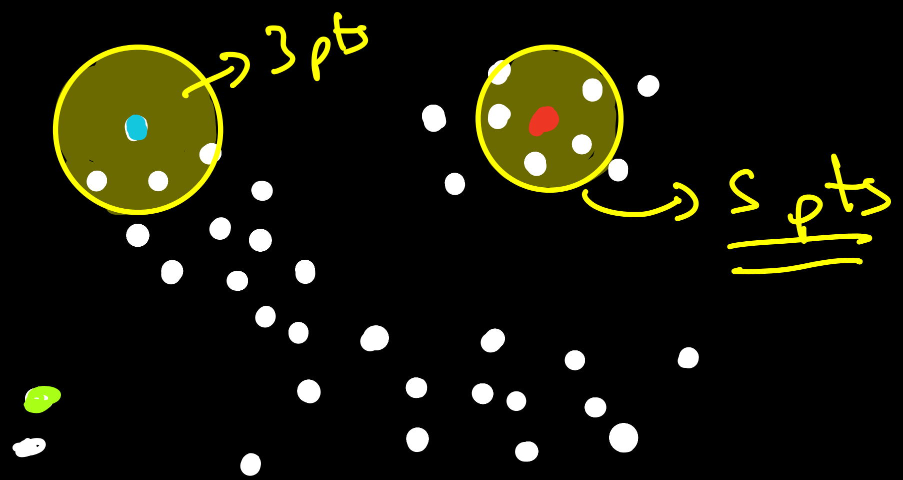
→ if a point is surrounded by

many other points its in the cluster!!

Observations:



Q: Can you think of a way for a computer to find these??



→ Draw a circle of radius eps

→ Count # pts in circle

→ if # pts > min pts

↳ core pt

else

↳ non-core pt.

→ if any non-core pt is inside circle of any core pt, then → border pt

else

→ noise pt

→ animation

→ categorise each pt into

- → core
- → border
- → noise

→ Join them based on neighbours, dont join 2 seperate borders,

Pros:

- → works with arbitrary shapes
- → No need to decide 'K'

Cons:

- → Does not work well with sparse points (high dim)
- → needs entire data set for inference.

→ code

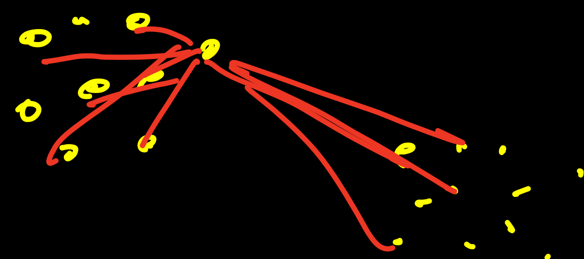
→ Time complexity : $O(n^2)$

$\downarrow$

need to calc distances of all points w.r.t all

Deciding epsilon [Extra]

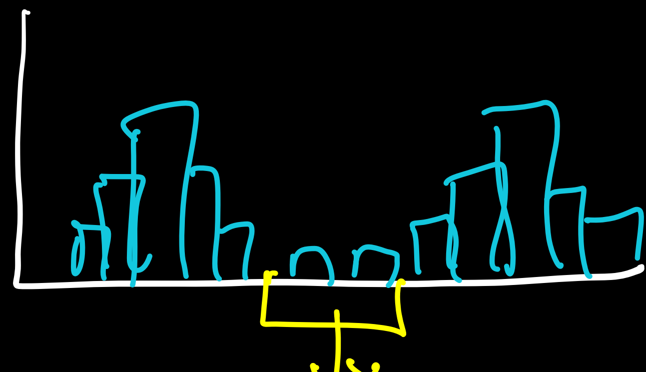
One way to estimate for far away clusters is:



→ calc distances between each point

→ plot a histogram of those distances

→ You may get 2 peaks, eps in b/w these peaks.



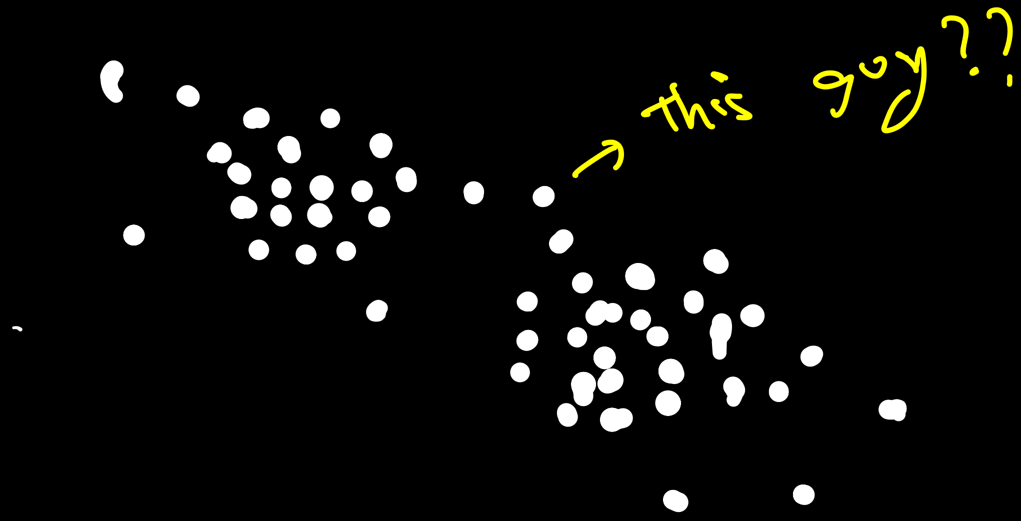
eps → starting pt

Gaussian Mixture Models

Soft Clustering → 4th big idea!

Problem: With classification algos I could get probabilities. How do I get probability of a pt belonging to a cluster?

Q: Any ideas ??



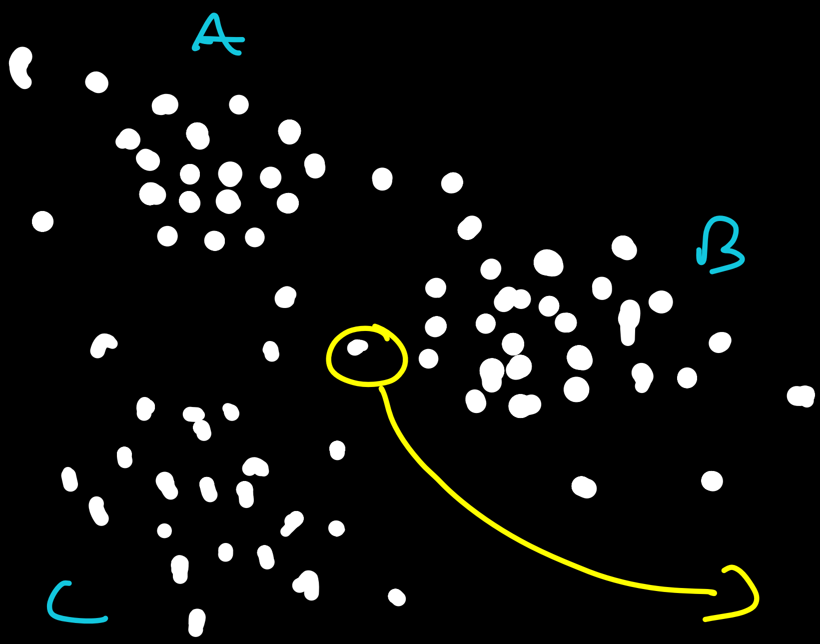
Q: How many clusters do you see?

→ In business we can make multiple policies:

→ Eg: A Rich / Premium — More ads

B Medium — discounts + ads

C Discount lovers — More discounts



closest to B
then to C
then to A

Q: So what % of ads and discount do I give to this guy?

$x_i \longrightarrow$ 50% B   20% A   30% C

$= 0.5(Dis + Ads) + 0.2(Ads) + 0.3(Dis)$

$= 0.8(Dis) \quad 0.7.(Ads)$

$= \frac{0.8}{0.7+0.8}(Discount) + \frac{0.7}{0.7+0.8}(Ads)$

$=$ 53% Discount

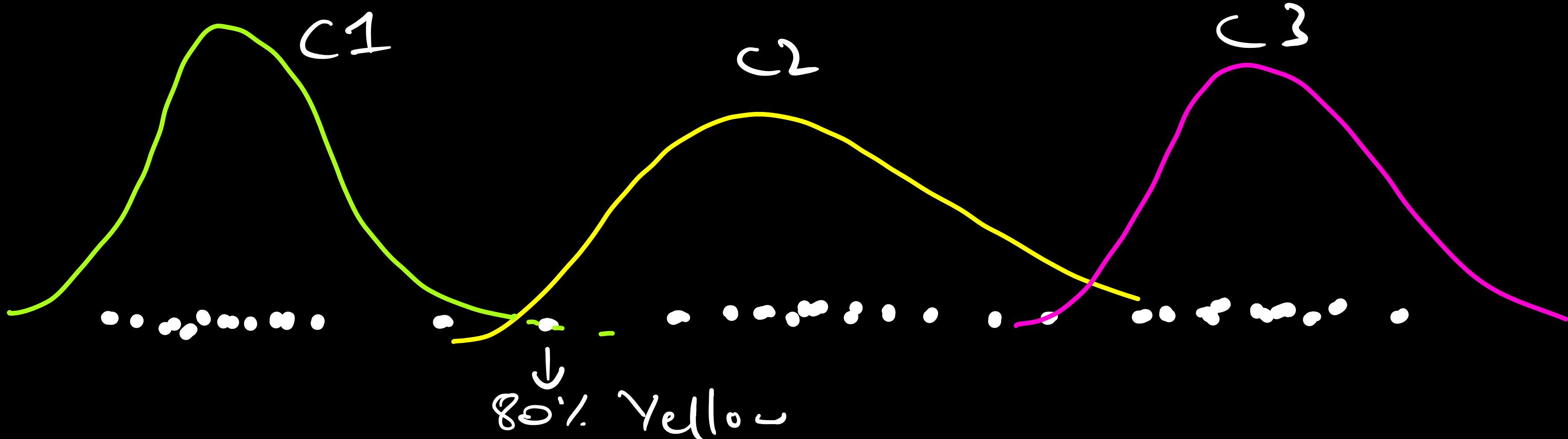
$=$ 47% Ads

Whateve budget we have per customer, for this guy we should spend 53% on Discounts, 47% on Ads.

Idea: Use n-d gaussian dist to express clusters !!

Lets discuss this in 1-d first

Q: How many clusters ?



19% Green
1% Pink

Q: What do you need for gaussian?

→ μ, σ

↳ I want 3 clusters:

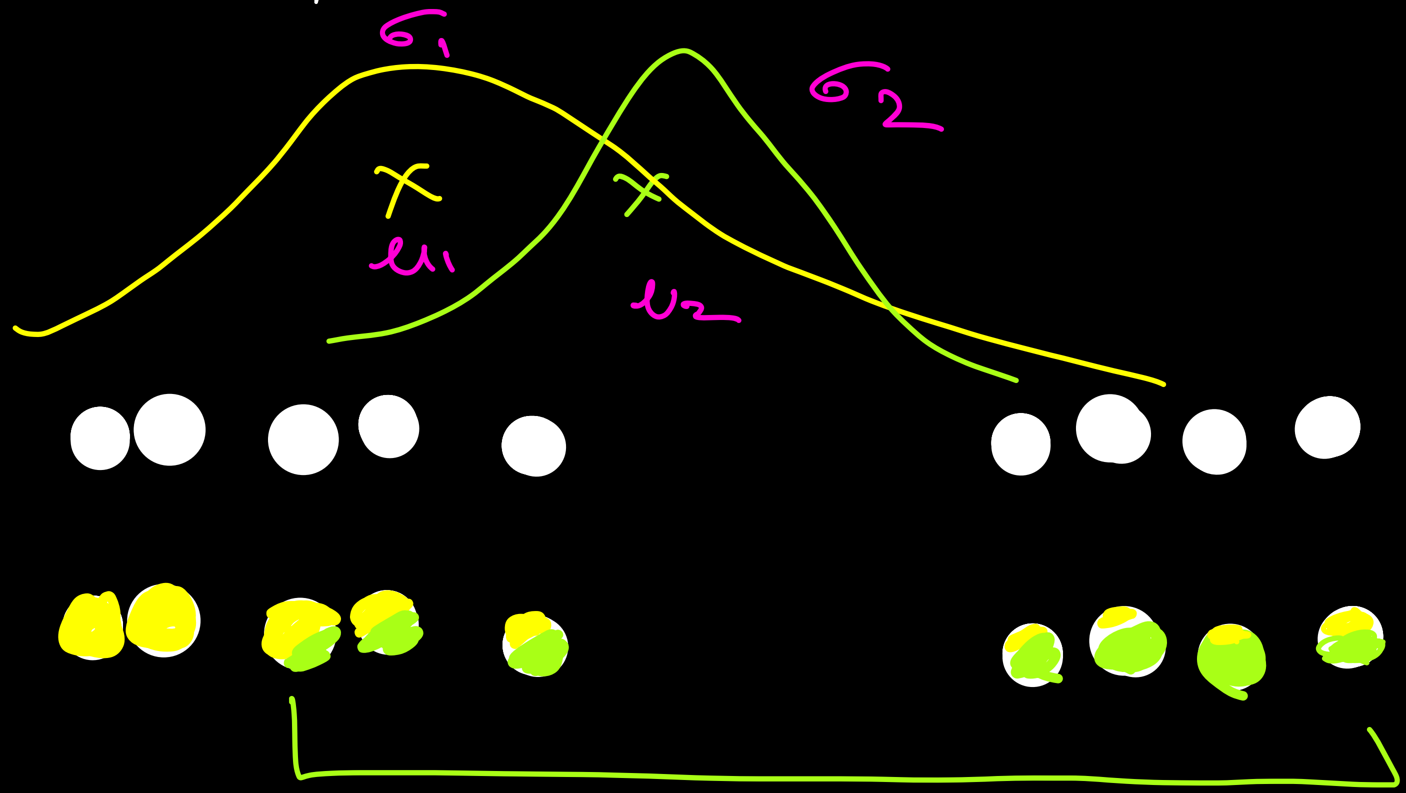
$\mu_1 \quad \mu_2 \quad \mu_3$

$\sigma_1 \quad \sigma_2 \quad \sigma_3$

Algorithm:

Very similar to K-means

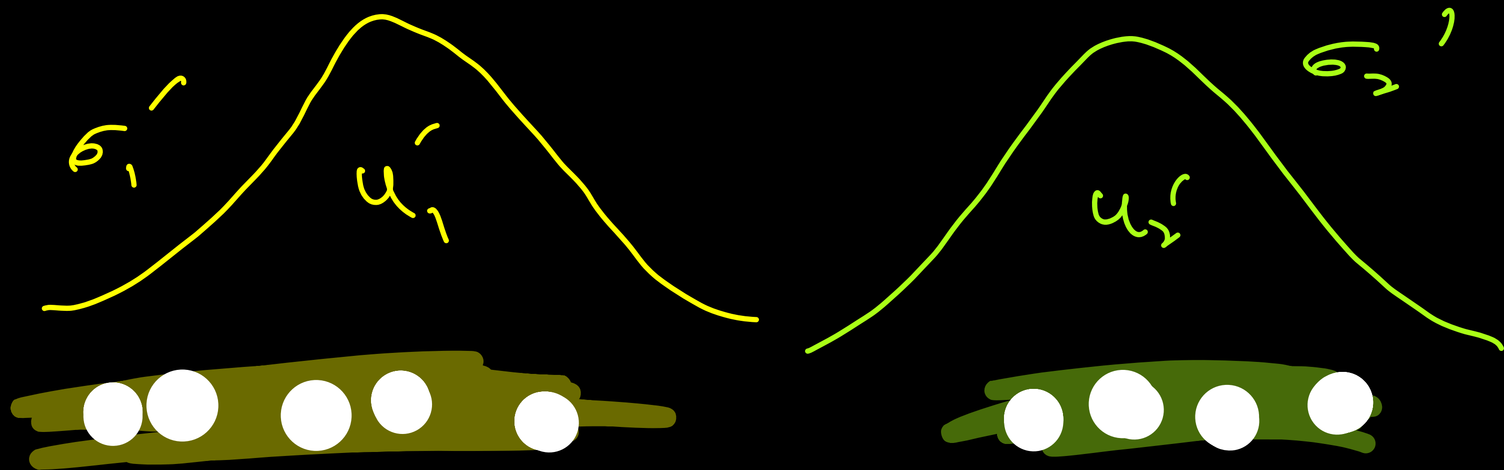
→ Random μ, σ initialise



$$\mu_2' = \sum p_i(G) \cdot x_i$$

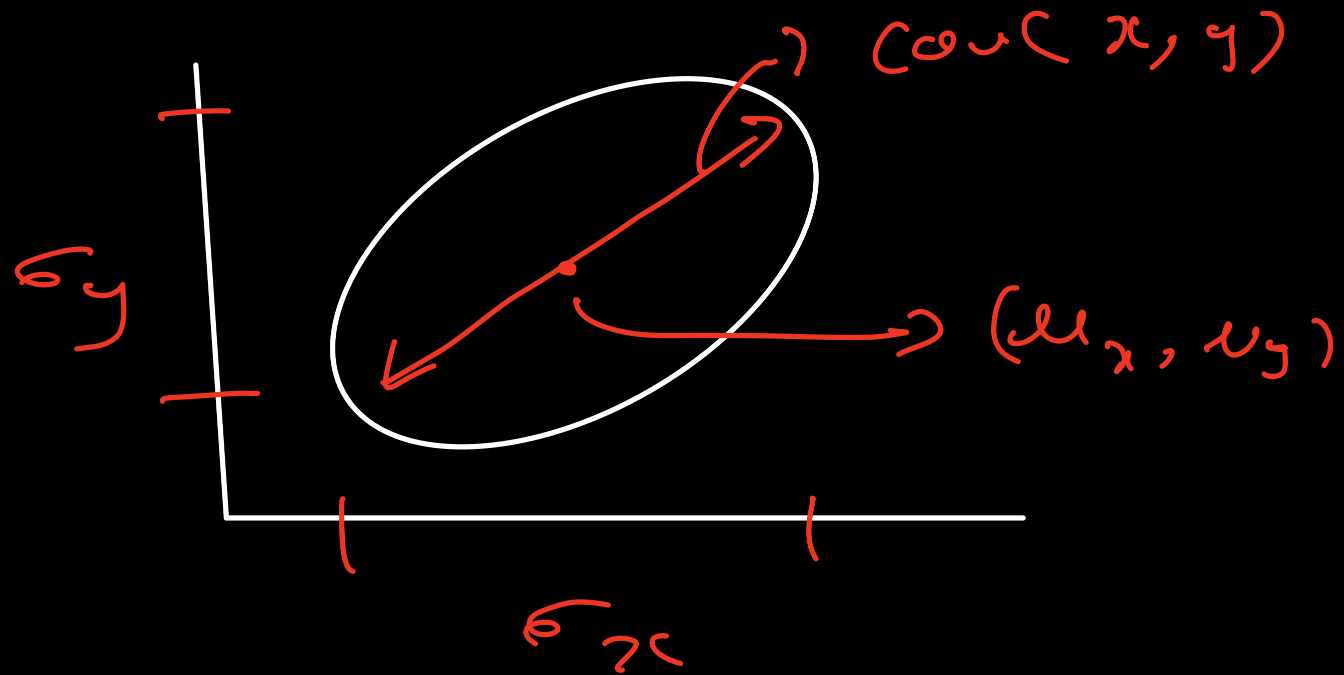
$$\mu_1' = \sum p_i(Y) \cdot x_i$$

$$\sigma_j' = \sqrt{\frac{1}{n-1} \sum_{i=c_j} (x_i - \bar{x}_j)^2}$$



After multiple updates, you will have tightly fitting gaussians.

2D Gaussians!



params to update = $\underline{\underline{5}}$

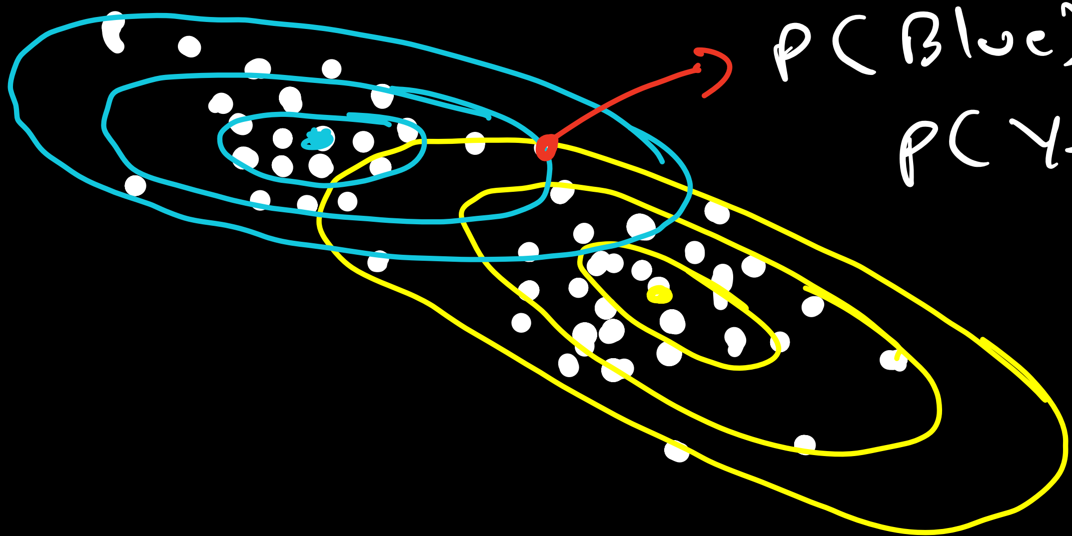
$\mu_x, \mu_y, \sigma_x, \sigma_y, \rho_{xy}$

same $\underline{\underline{\text{algo}}}$

How to get these probabilities?

↓

Gaussian Distributions → Modelling choice, you could create a variation with another dist.



p(Blue)

p(Yellow)

→ 2D gaussian

→ animations

→ code

Pros and Cons are similar to KMeans

Results are also very similar !

Extra Pro:

→ May work with diff size clusters because we also have control over variance, i.e size of clusters

→ May work with hyper-eliptical shapes too. [Kmeans can't]