DBSCAN & GMM

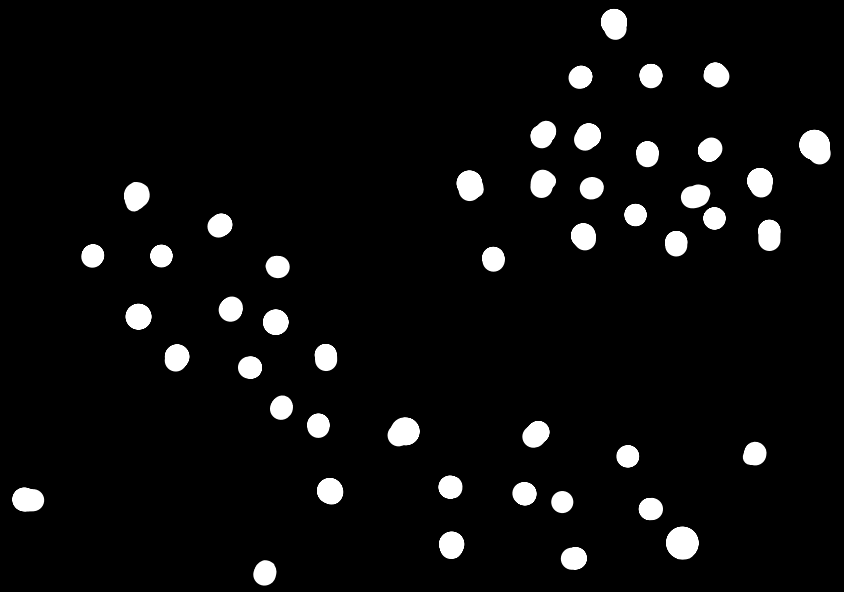
[Clustering]

→ 2 new ideas !!

→ Comparision of algorithms

DBSCAN → 3rd big idea

Density based spacial clustering application with noise.



Q: How many clusters do you see?

a) 1

b) 2

c) 3

d) 4

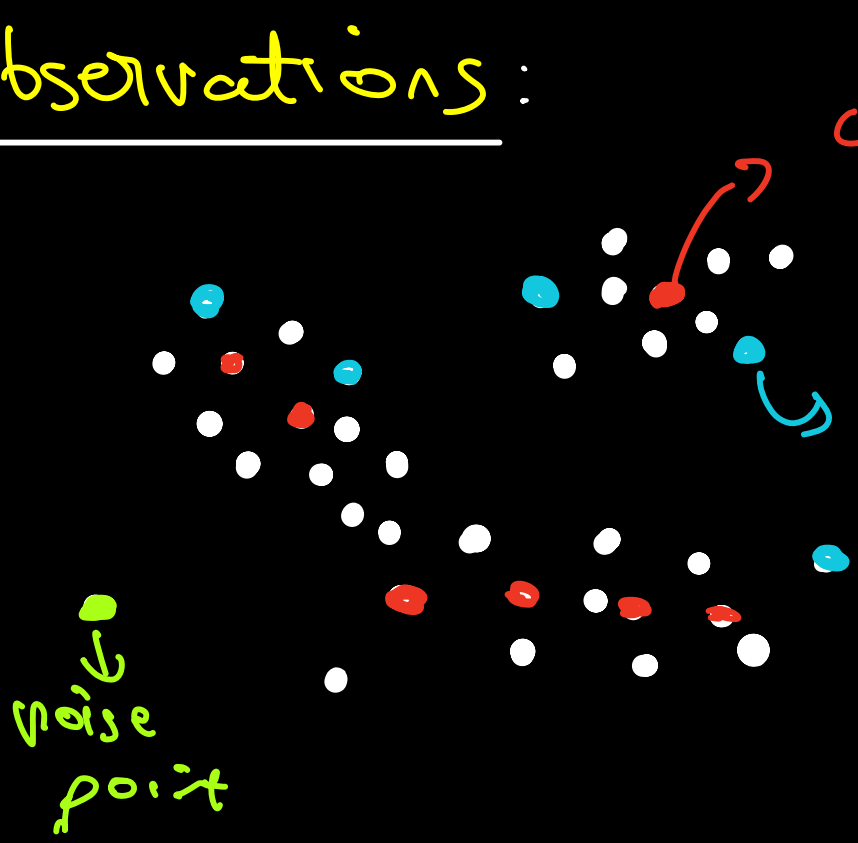
Q: Do you think KMeans will work?
↳ No

Idea:

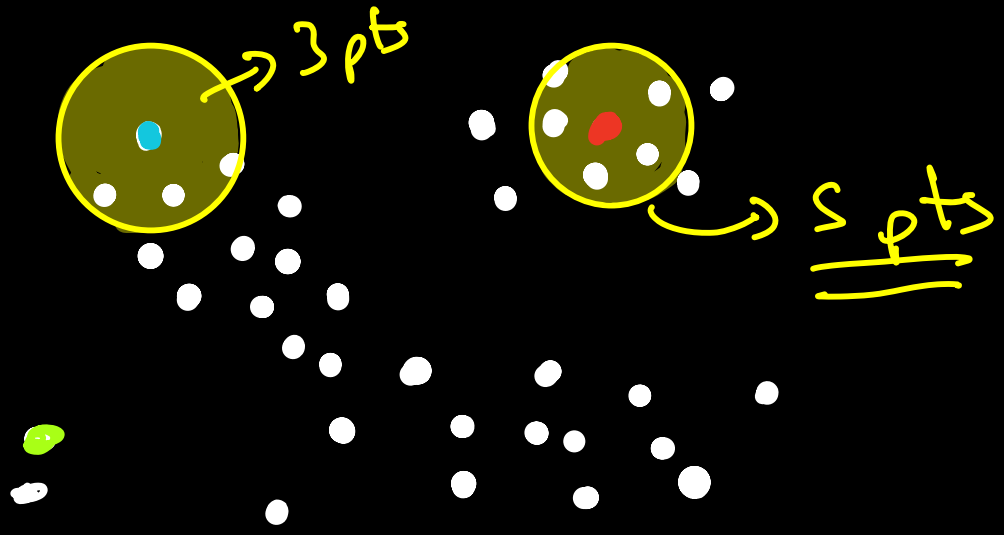
→ if a point is surrounded by

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 pts 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
→ code

→ Time complexity : $O(n^2)$

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

Quiz: Which of the following algos can be used to detect outliers

a) Kmeans

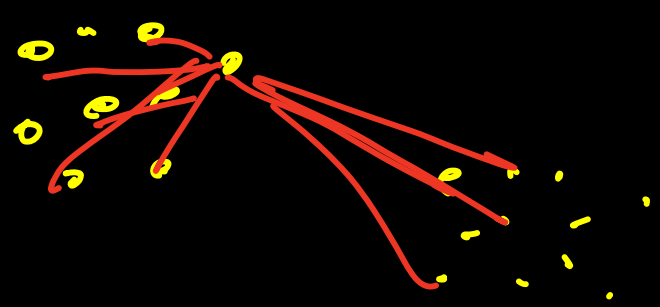
b) Hierarchical

c) DBSCAN

d) Clustering algos can't be used for outliers.

Deciding epsilon

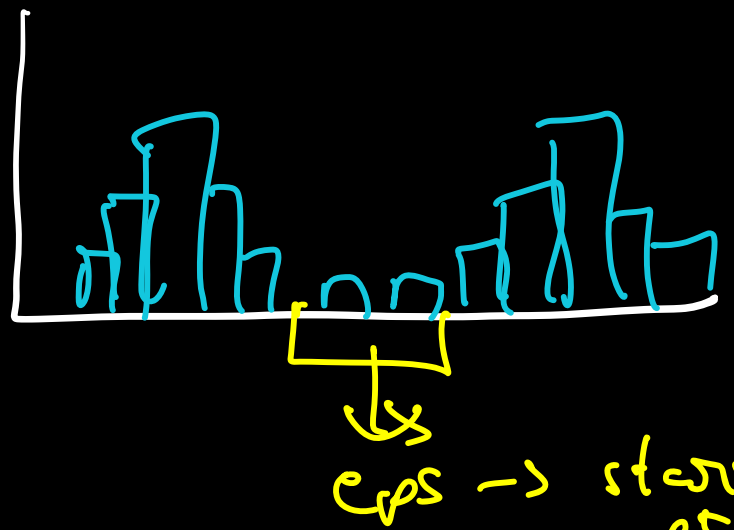
One way to estimate for far away clusters is:



→ Calc distances between each point

→ plot a histogram of those distances

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

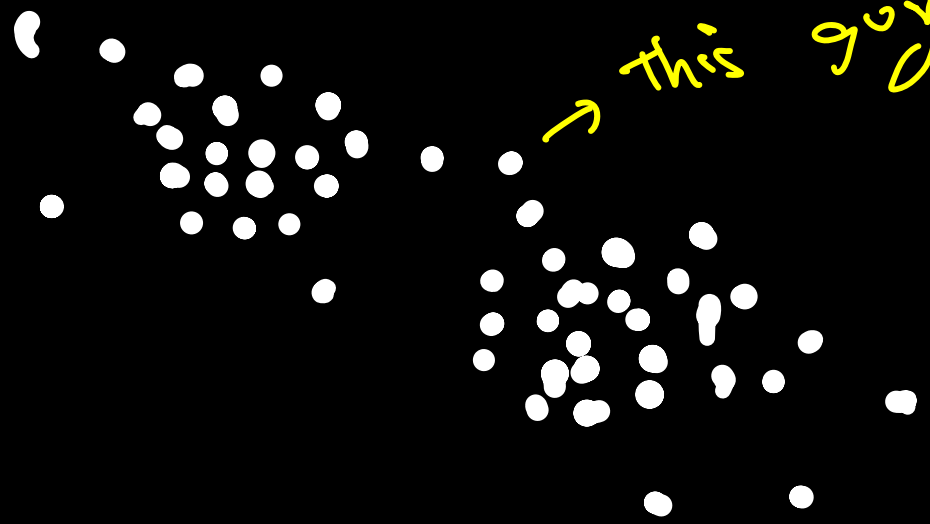


Gaussian Mixture Models

Soft Clustering → 4th big idea!

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

Q: Any ideas ??



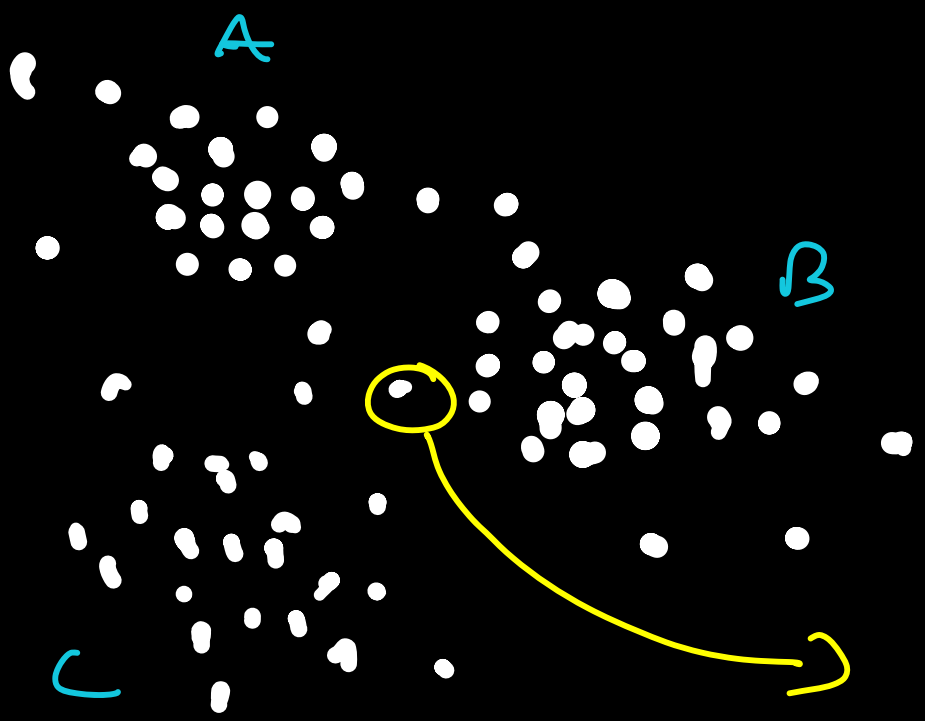
Q: How many clusters do you see?

→ 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?

Ai → 50% B 20% A 30% C

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

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

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

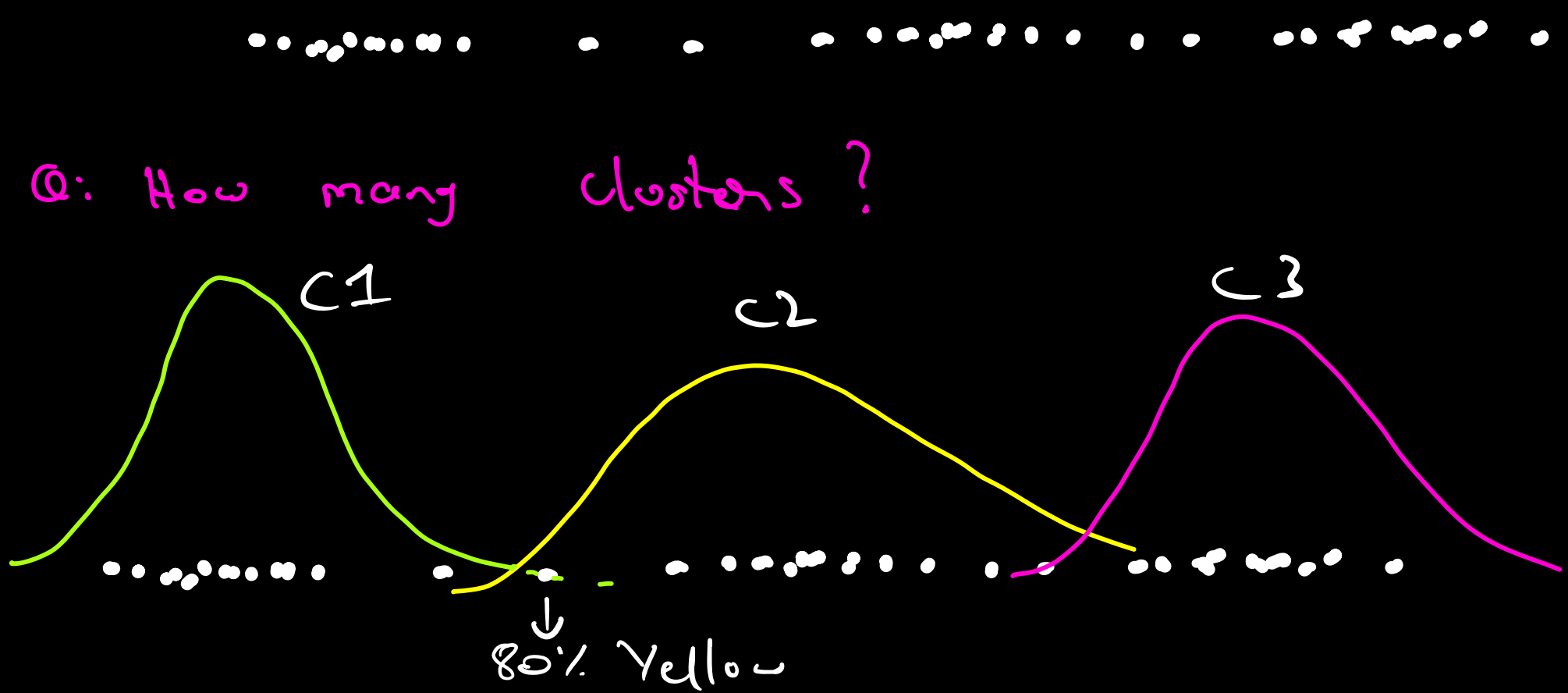
$$= 53\% \ \text{Discount}$$

$$= 47\% \ \text{Ads}$$

Whatever budget we have per customer, for this guy we should spent 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:

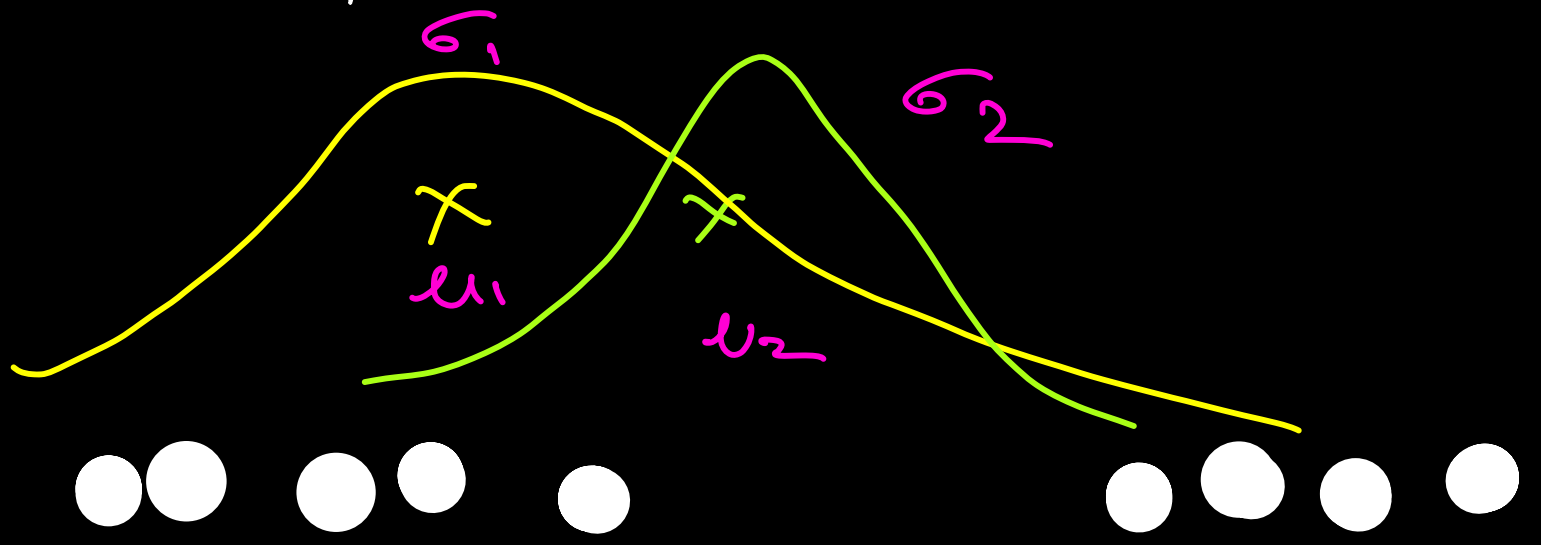
μ_1 μ_2 μ_3

σ_1 σ_2 σ_3

Algorithm:

Very similar to K-means

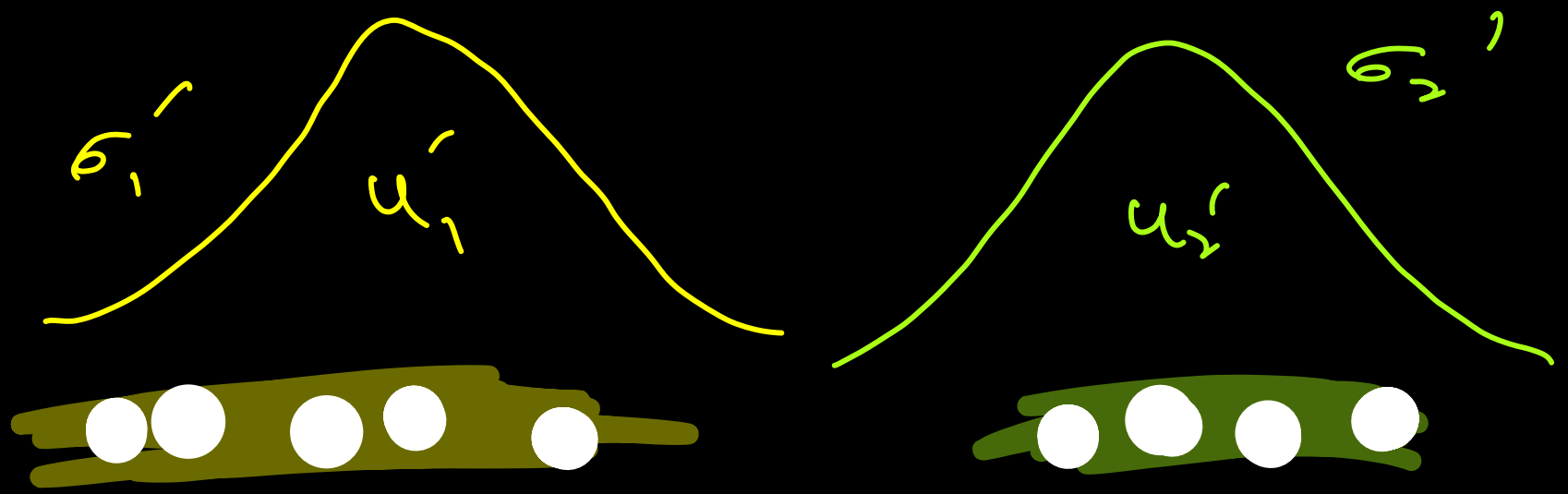
-> Random μ, σ initialise



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

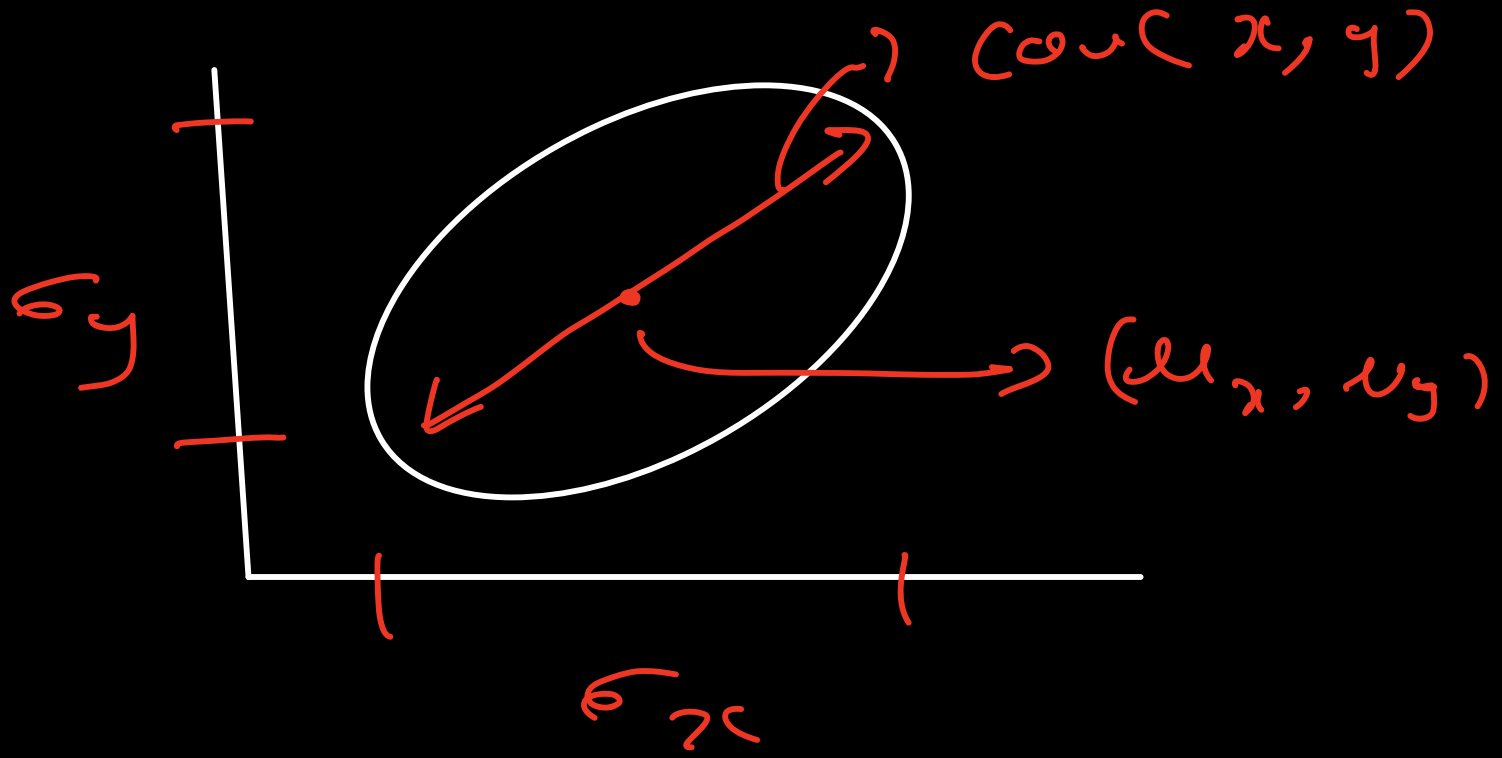
$$\mu_1' = \sum p_i(Y) \cdot x_i$$

Similarly we will code variance



After multiple updates, you will have tightly fitting gaussians.

2D Gaussians!



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

$u_x, u_y, \sigma_x, \sigma_y, S_{xy}$

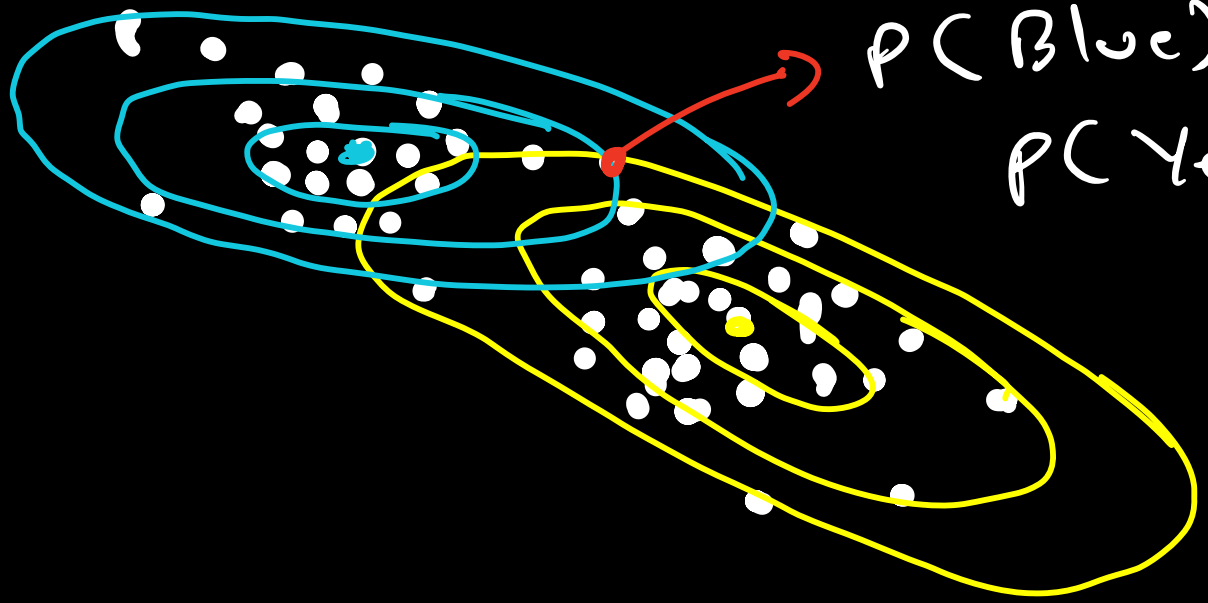
↓

same algo

How to get these probabilities?

↓

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



→ 2D gaussian

→ animations

→ code

Pros and Cons are similar to KMeans

Results are also very similar !