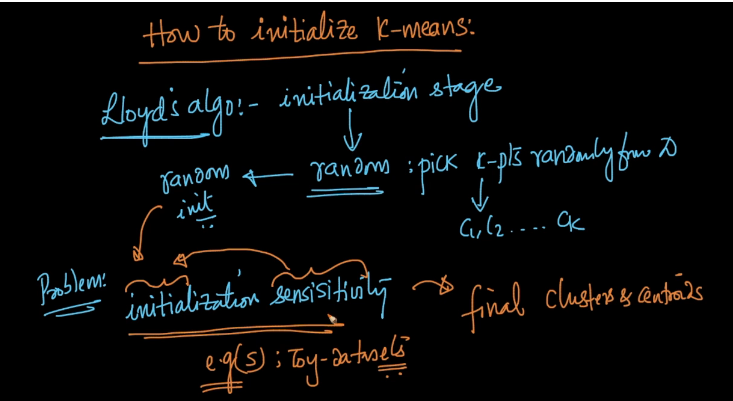
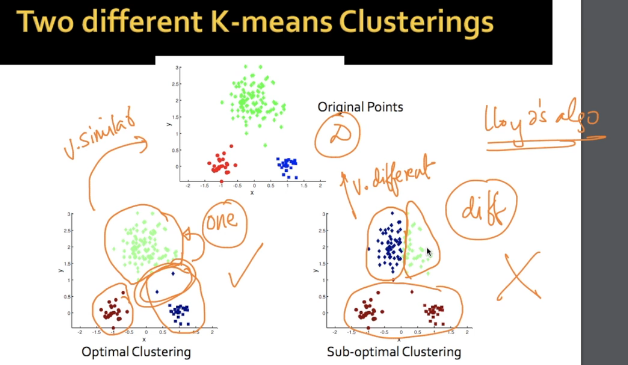
So now there is one problem with the random initialization stage i.e. the clusters formed at initial stage are sensitive to initialization because everytime we are picking random k points.

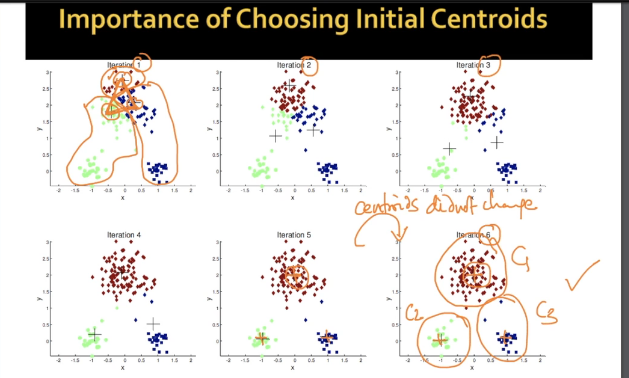


Lets see how it behaves on different toy datasets.

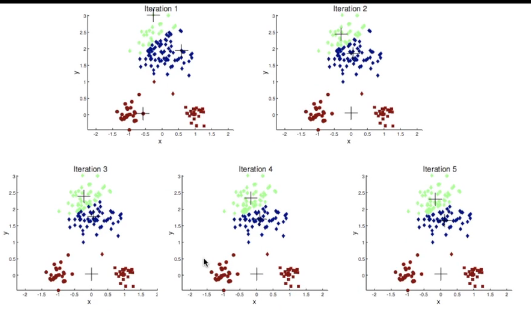
So in below image we have three graphs showing clusters where one graph is very similar to original or human made cluster graph and another is not what we expect.



SO when we choose our centroids as shown in below image it gives us well separated and desired clusters.



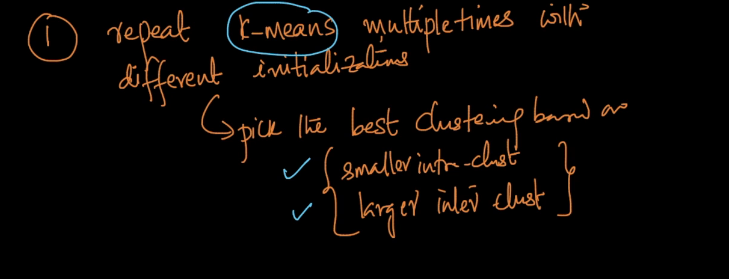
But when we choose below clusters it gives undesirable clusters.



This shows how important it is to choose initial clusters.

TO overcome this problem there are some ways:

1. To repeat k-means multiple times and pick best initial clusters.

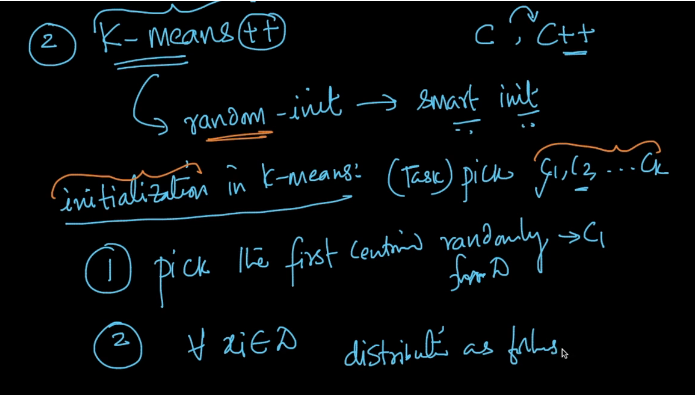


1. K-means++ is the second way to overcome this problem.

Now lets see how k-means++ works.

It is all same as k-means but works differently in choosing initial cetroids where we were facing problem.

It says instead of choosing all centroids randomly choose one centroid in random fashion and call it C1.



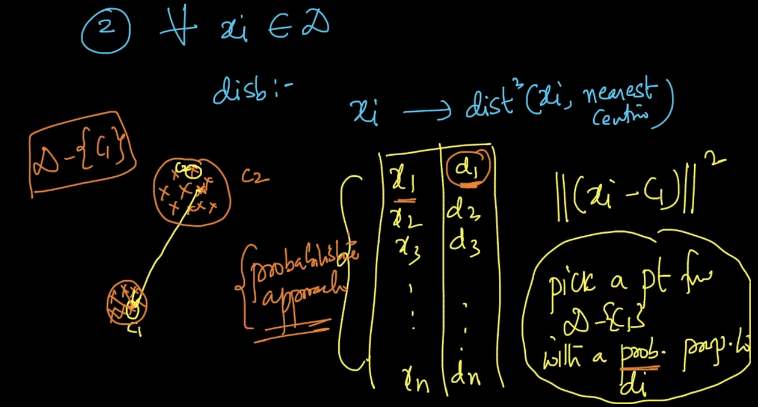
Now draw a distribution for every point Xi belongs to Dataset.

How to determine the distribution?

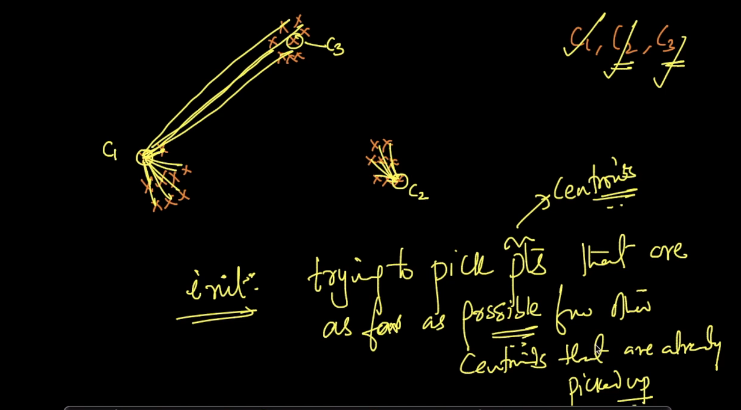
Distance^2(Xi, nearest\_centroid) and since we have picked one centroid so we form a table for every point and its distance squared.

Now we go with probabilistic approach of determining which point has higher probability of getting chosen as next centroid.

For this we choose max of ||(Xi – C1)||^2.

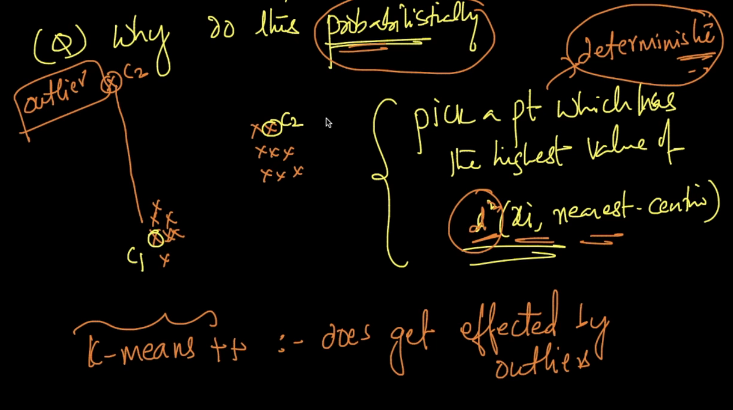


Now lets take an example where we need to pick 3 centroids .



After picking C1 we will check maximum distance and choose C2 but after choosing C2 for calculating distances we always choose nearest centroids and this fashion we will choose C3.

Now the question here is why choose Probablisticly?



So if will always choose the farthest point then there are chances of choosing outlier as centroid and so we choose it probabilisticl so as to reduce the impact of outliers.

But yes k-means++ is affected by outliers and so we can repeat k-means++ so to reduce impact.

**Comments:**

Refer: <https://cs.wmich.edu/alfuqaha/summer14/cs6530/lectures/ClusteringAnalysis.pdf>

