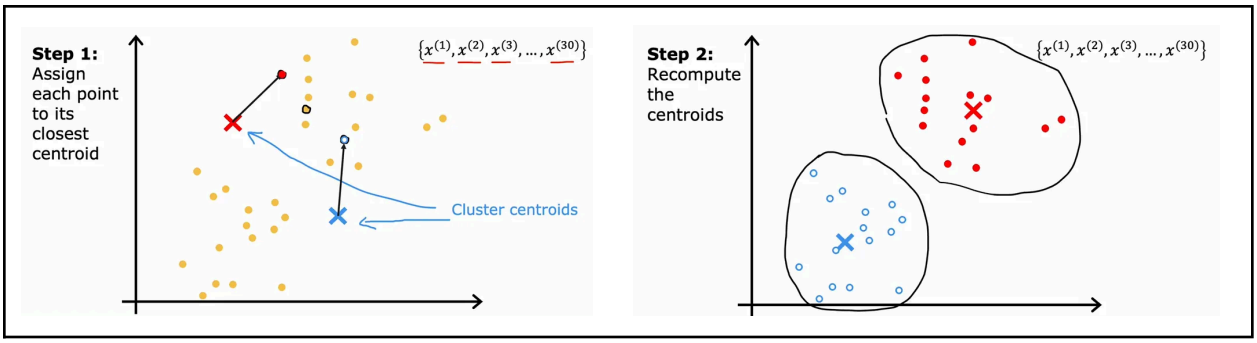
Unsupervised Learning, Recommenders, Reinforcement Learning

Clustering

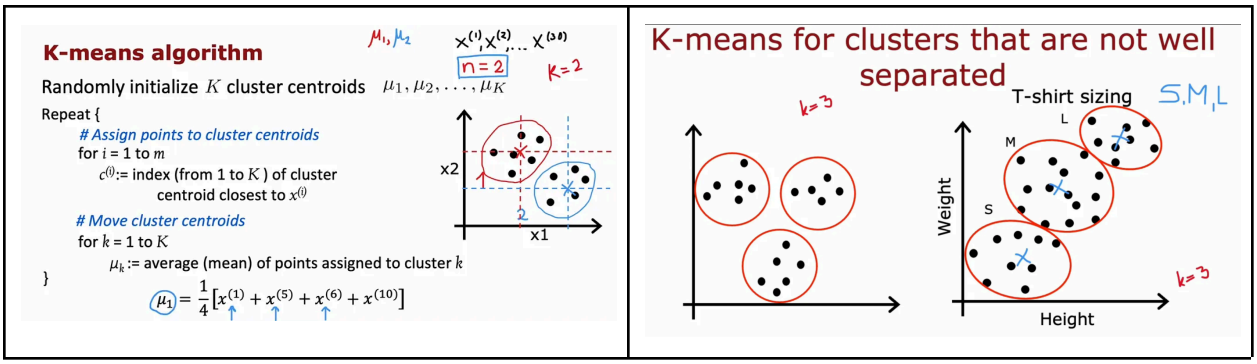
Unlike supervised learning, unsupervised learning analyze data that does not have labels not and tries to discover patterns within it. Two main task in unsupervised learning are clustering and anomaly detection. Clustering identifies groups of similar data points without using labeled outputs. Unlike binary classification, which is a type of supervised learning, clustering finds hidden structures using only the input data.

1. K-means Algorithm

K-means clustering is an unsupervised learning algorithm that groups unlabeled data into K clusters. It begins by randomly guessing K cluster centroids, then assigns each data point to the nearest centroid. Next, each centroid is moved to the mean position of the points assigned to it. This process is repeated until there are no further changes in the assignments or centroid positions.

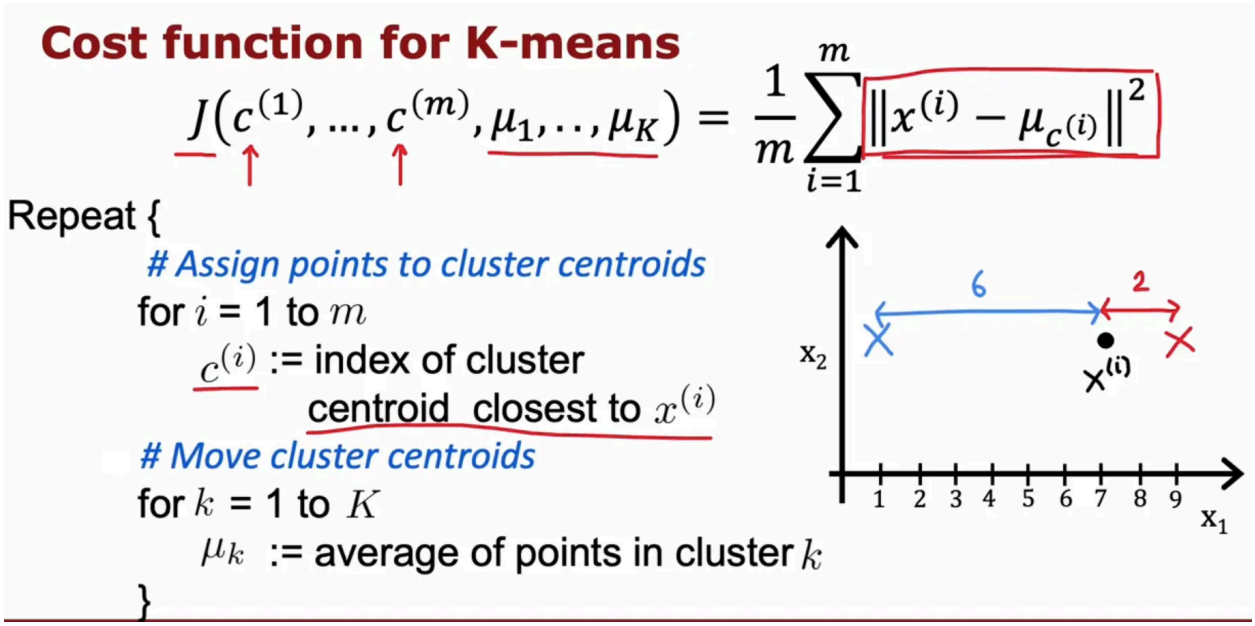


As mentioned earlier, the K-means algorithm starts by randomly initializing K cluster centroids, which have the same dimensionality as the input data. If a centroid ends up with no points assigned to it, it is usually either removed or randomly reinitialized.



Cost function for K means

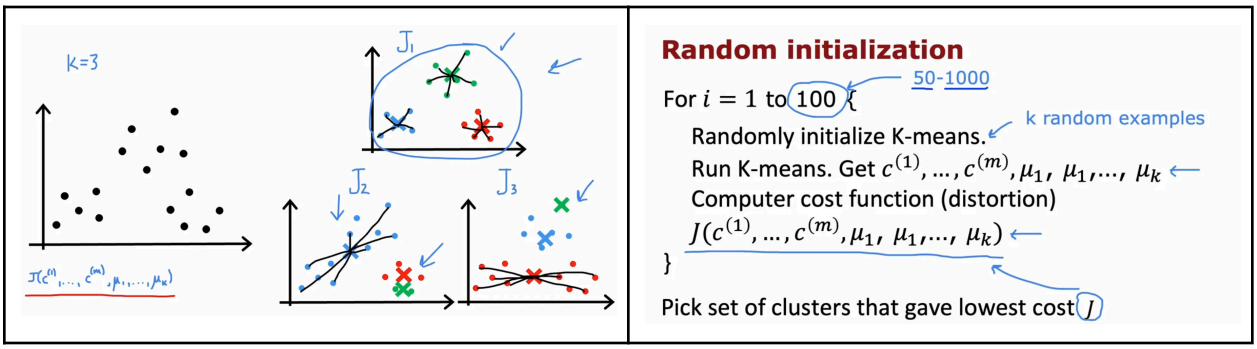
Cost function of the k means is squared distance between every training example $X^{(i)}$ and the location of the cluster centroid $\mu_{c(i)}$ to which training example $X^{(i)}$ has been assigned.



The initial step in K-means clustering is to select K training examples as the initial cluster centroids. However, different random initializations can lead to different clustering results, and sometimes the algorithm may get stuck in poor local minima of the cost function.

Therefore, instead of relying on a single run, it is better to run K-means multiple times with different random initializations. Afterward, you can select the clustering result with the lowest distortion cost.

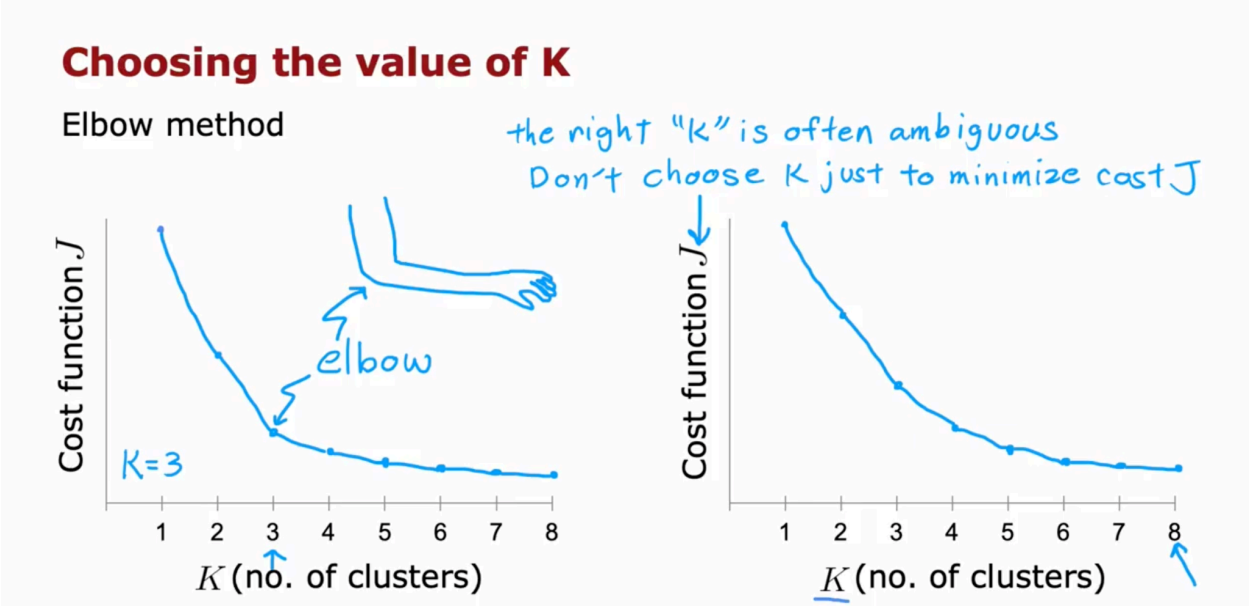
Initializing K-means



1. Always choose K < m because if k = m then you are end u choosing one cluster for each training example.
2. Randomly pick a K training example.
3. Set $\mu 1, \mu 2, \ldots, \mu k$ equal to these k examples.

But problem with 3rd is that ki random initialization hr baar kaam nhi krti kai baar randomly intialize cluster same cluster me sub groups bana deta h isliye jbb bhi hum k-means algorithm chalate h to ek loop me rkhte h say 100 baar randomly initalize kr k cost function nikala or fir jisme sbse kmm cost function mila use best algo assign kr diya

Choosing the number of cluster



First method is elbow curve jahan pr sharp drop mile use pick kr lo baki second methods is

So how do we determine the ‘right’ value of K? It’s often ambiguous because clustering is unsupervised and lacks true labels. A common approach is to try different values of K and choose based on the trade-off in usefulness for the downstream task

Choosing the value of K

Often, you want to get clusters for some later (downstream) purpose.
Evaluate K-means based on how well it performs on that later purpose.