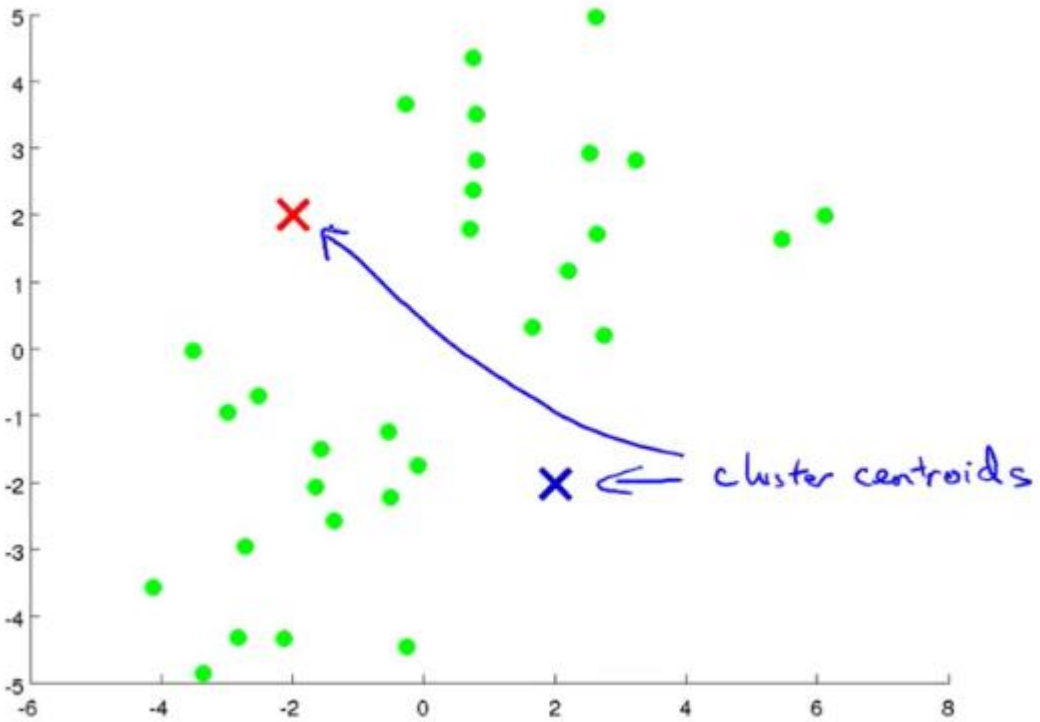
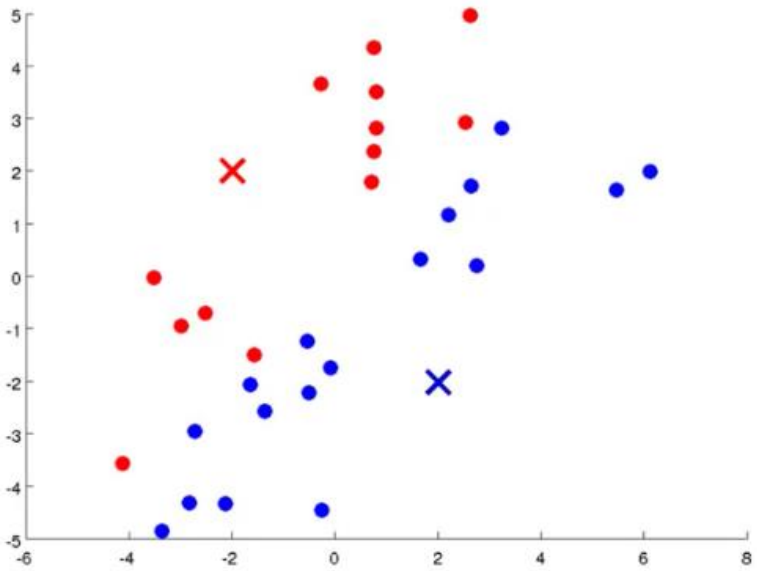
Page No.
Date :

K-Means Alogorithm :

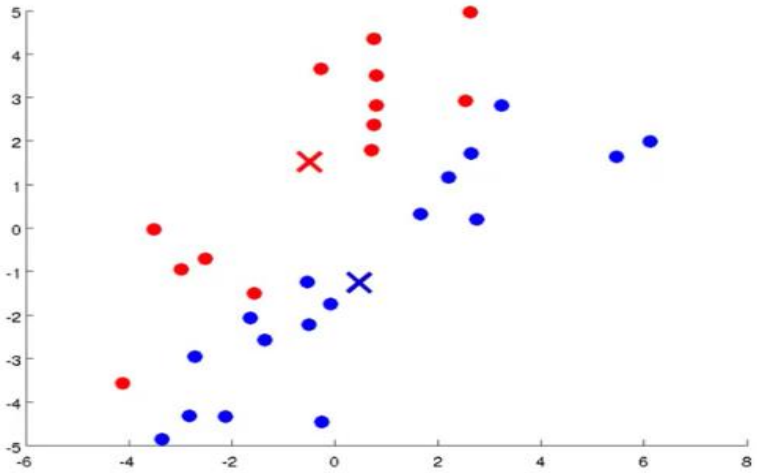
1 We first initialise some points on the features graph (say X_1 vs X_2) : These points are called cluster centroids :



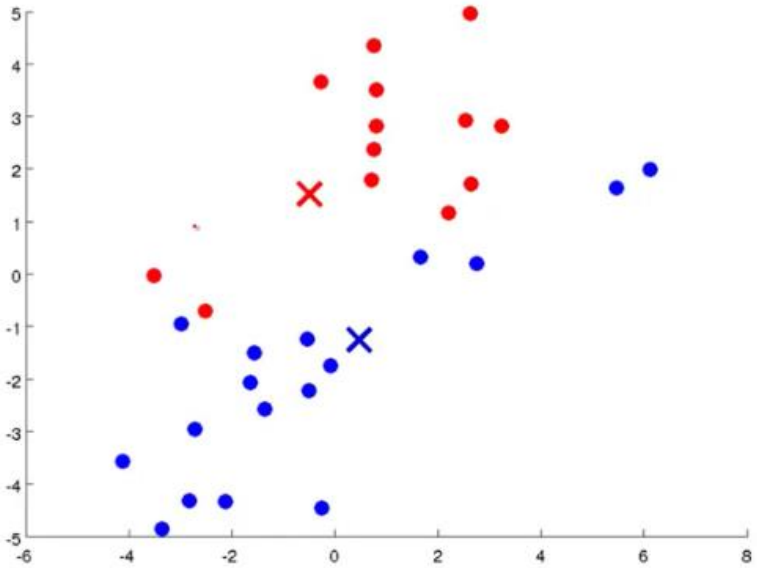
2 Now the points near the blue cross are organised into a blue cross group & near the red cross are organised into a red cross group.



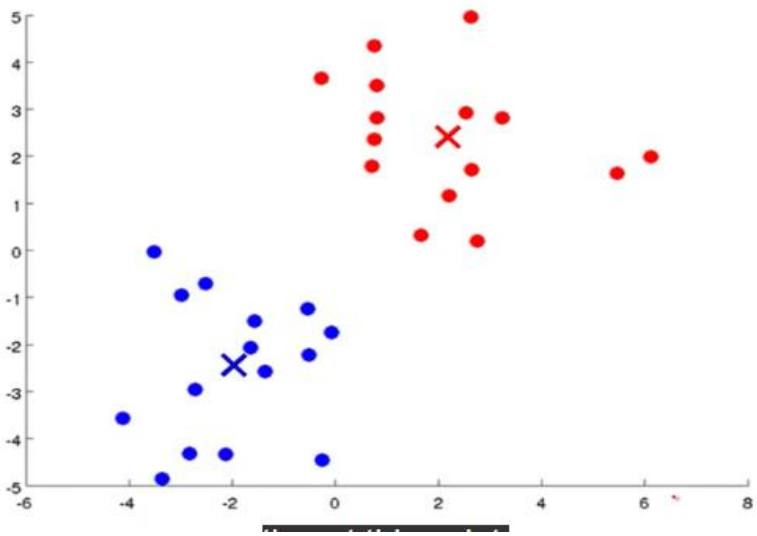
3 Then we take the mean of the points present in blue group & the mean is the new position for the blue cluster centroid. We do the same for red cluster centroid.



4 Then we again find points near the blue cross & label them as blue group & points near the red cross as red group.



We repeat the above-mentioned process again and again until we arrive at the right classification:



If you keep running additional iterations of K means from here the cluster centroids will not change any further and the colours of the points will not change any further. And so, this is the, at this point, K means has converged and it's done a pretty good job finding .

K-means algorithm explained mathematically:

Note:
1.k=1 to K represents number of cluster centroids.
2.i=1 to m represents the number of datasets/rows of data

Steps in which k means algorithm works:
1.We first randomly initialise K cluster centroids at positions mu_1, mu_2,…….,mu_k
Note that mu_1 ,mu_2…mu_n are n row vectors:

$$\underline{\mu}_1, \underline{\mu}_2, \ldots, \underline{\mu}_K \in \mathbb{R}^n$$

2. Let us consider we have only 2 columns/features in X i.e. x1 and x2. Now for each row of X, we plot x1 vs x2 and find the distance of these points from mu_1 and mu_2 and the points belong to the cluster centroid from which they have the minimum linear distance.

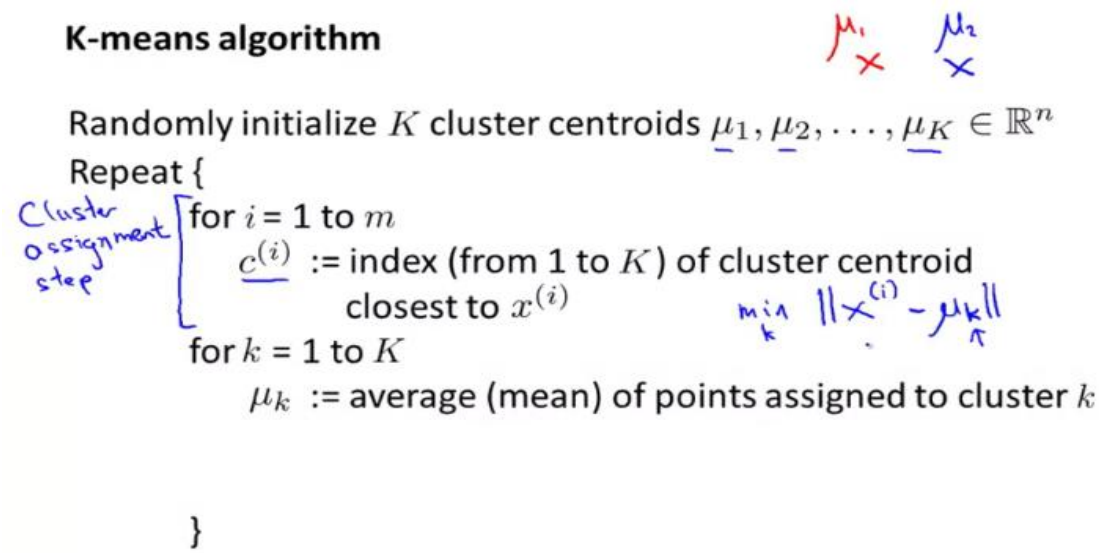
3. Say X^1 has the least linear distance from mu_2 and X^2 from mu_1 and X^3 and X^4 from mu_1 and mu_2 respectively and so on.

4. The we find the mean of the points belonging to mu_1 and mu_2 respectively. Since X^1,X^2… X^n are 2 row vectors(since they have 2 columns each representing 2 different features), their mean is also a 2 row vector . The mean of the points belonging to the two groups mu_1 and mu_2 indicates the new coordinates of mu_1 and mu_2.

5. We then again find the distance of the points on the graph from mu_1 and mu_2 and the points belong to the cluster centroid from which they have the minimum linear distance.

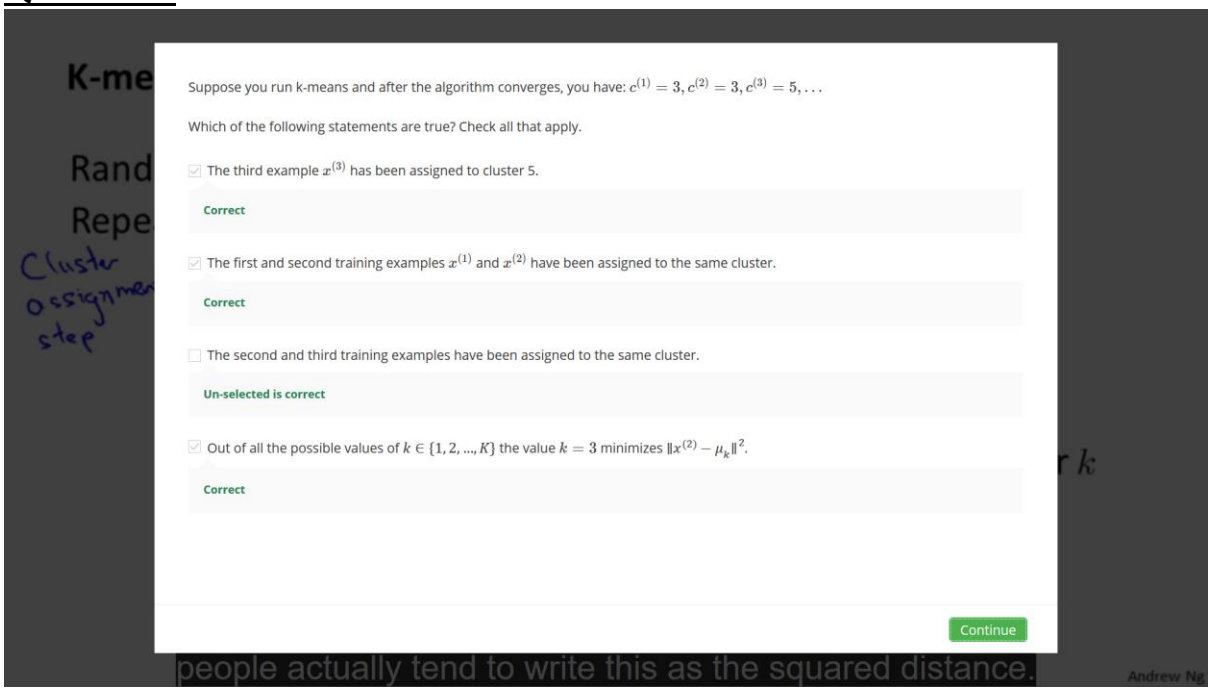
6. Then we repeat steps 3 and 4.

The process is illustrated below:



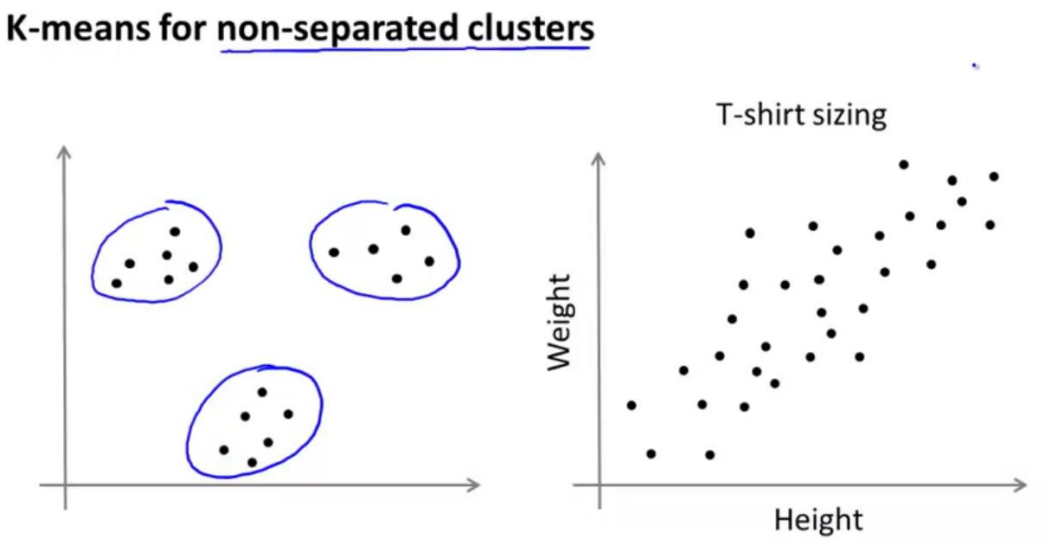
Instead of taking $\min_k \|x^{(i)} - \mu_k\|$ $\rightarrow c^{(i)}$, we can also take $\min_k \|x^{(i)} - \mu_k\|^2$ $\rightarrow c^{(i)}$ i.e. square of the linear distance between cluster centroids and the points to find which points belong to which cluster centroid.

Question:

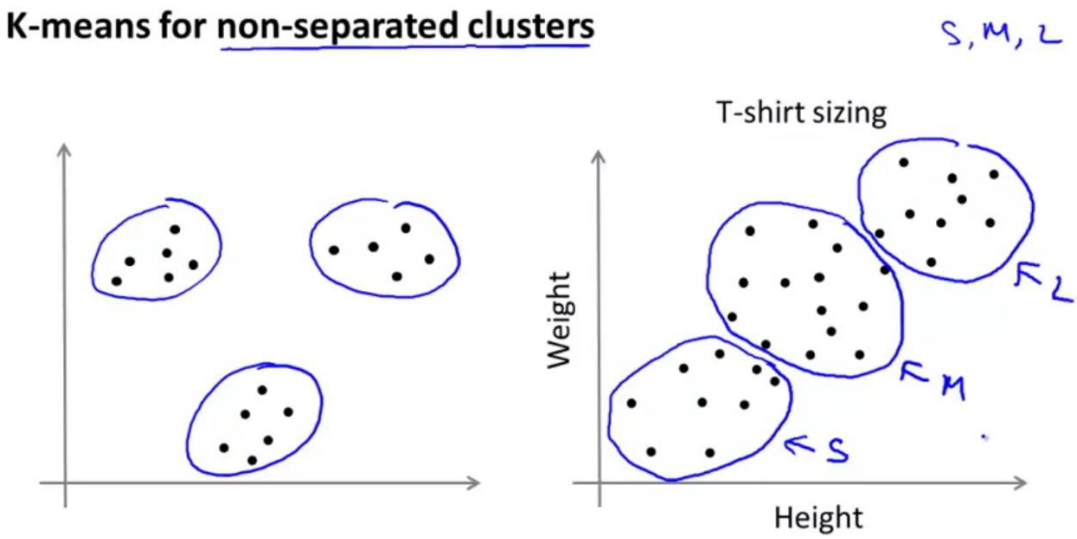


K-means for non-separated clusters:

In the graph shown on the right side of the figure , we see data that doesn't seem to be easily separable into clusters:



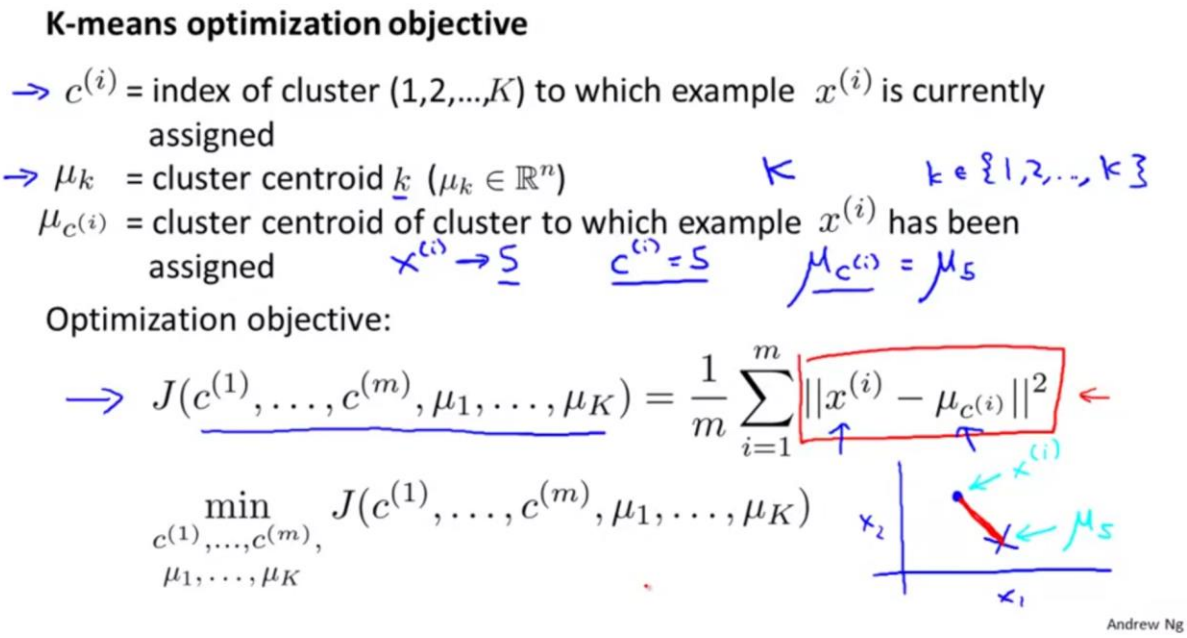
The K-means algorithm can separate this data into following cluster:



The clusters represent small, medium and large t-shirt sizes.

Optimisation Objective:

Refer to the image below:



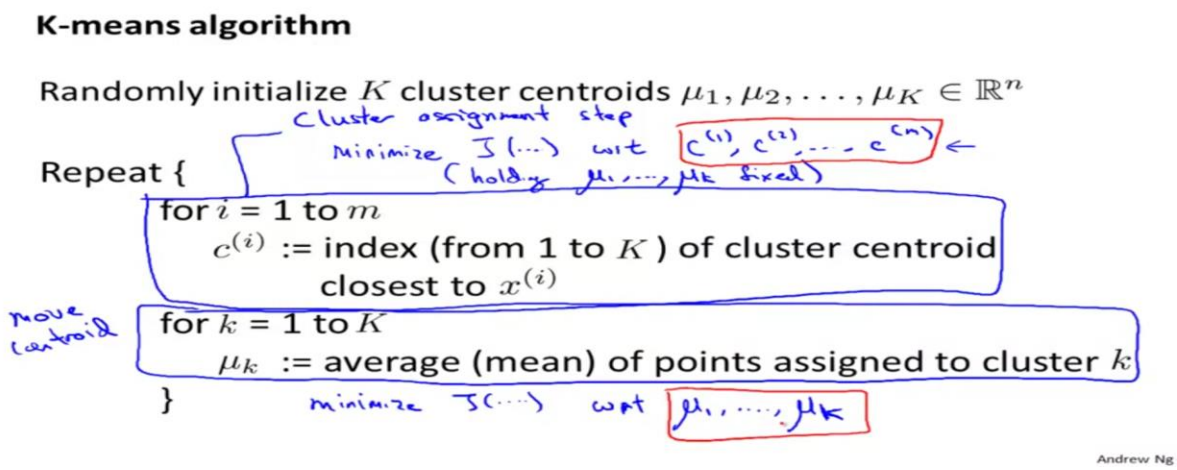
We want to minimize the cost function J by obtaining values of mu_1 ,mu_2…mu_n and c^1,c^2,…c^n such that when we find the sum of the square of distance between the mean of the cluster centroid and a point say x^i belonging to that cluster for all x^1 to x^m , the sum obtained is the minimum possible one.

The above lines mean that we want to classify the points in the best possible way.

Showing how the cost function is calculated:

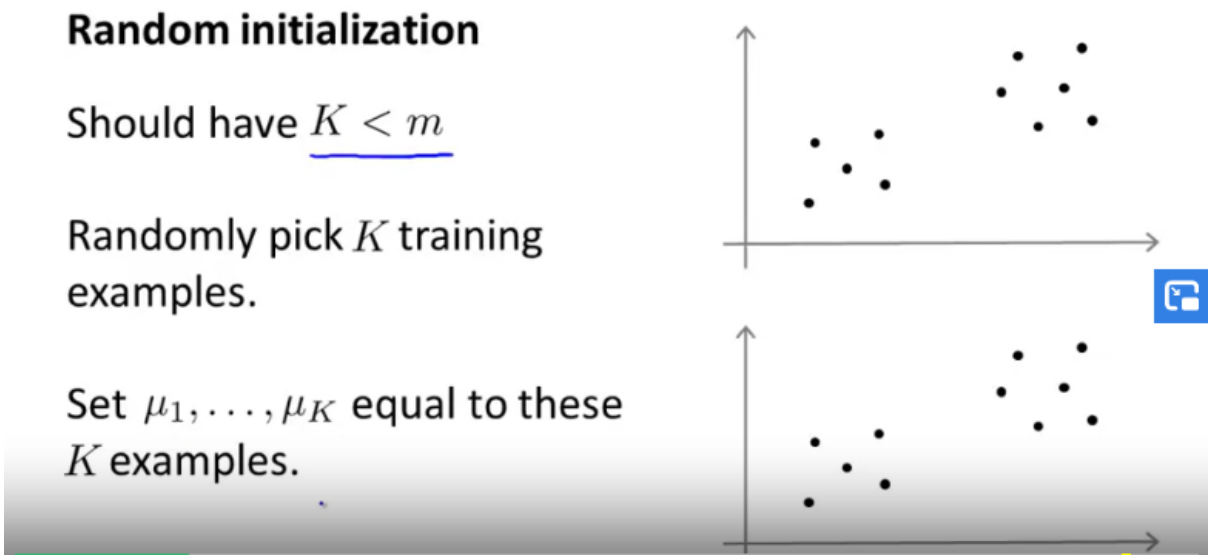
1.In the first step we randomly initialise mu values and find cost function for all i ranging from 1 to m.

2. In the second step we find mean of points assigned to cluster k and repeat step 1.

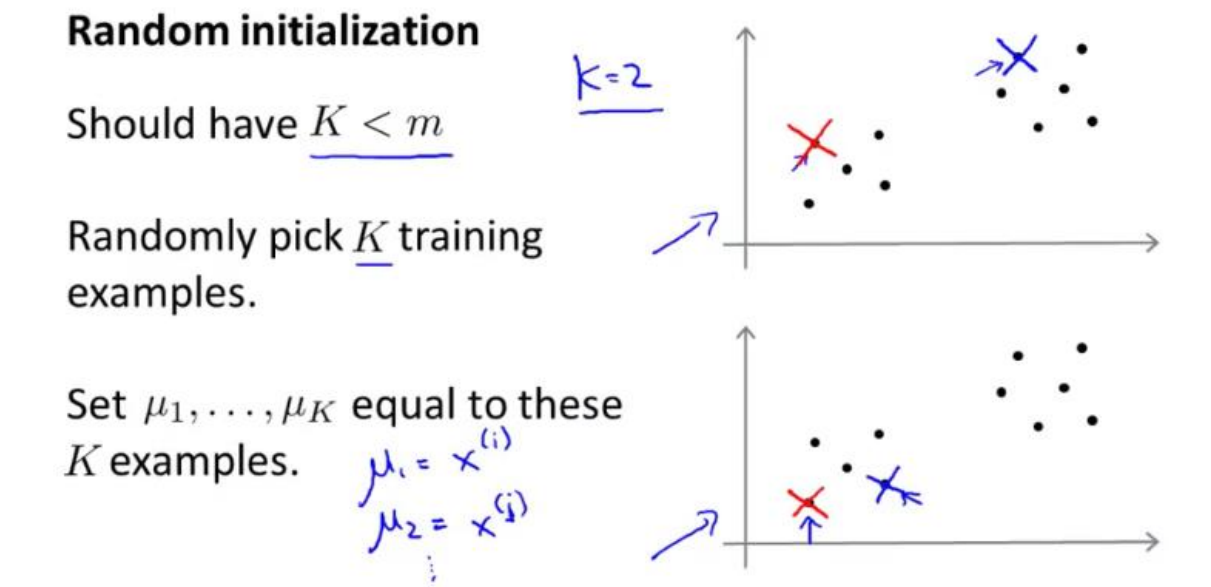


Random initialisation:

How do we initialise the mu_1, mu_2, ….. mu_n ?



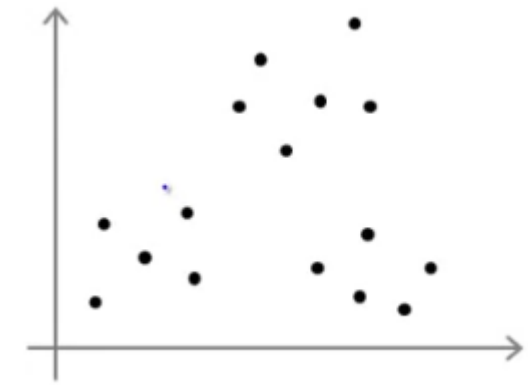
Now, it can be that we end up on different local optima values depending on how we initialise the mu values.



Above 2 graphs show 2 different initialisations of mu_1 and mu_2.

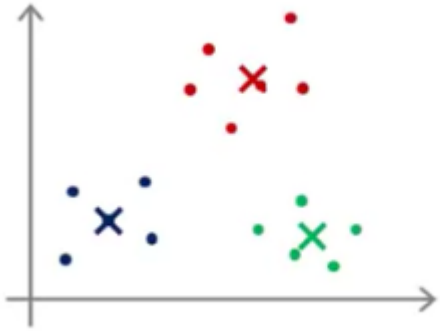
<u>The case of different local optima:</u>

Consider the graph below.

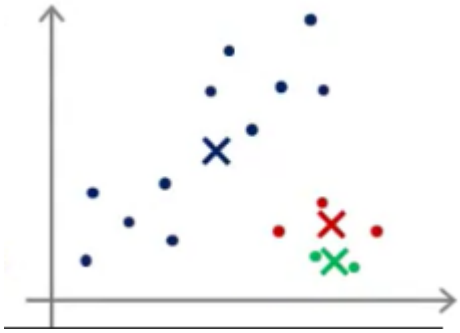


Now depending on how we initialise mu values, we may end up on different local optima and hence different clusters:

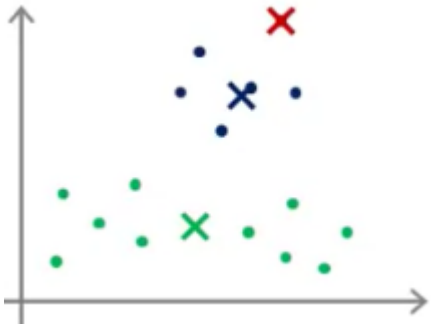
1. Initialisation 1:



2. Initialisation 2:



3. Initialisation 3:



Solving the above problem

We see above that different initialisations can lead to different clusters. So how do we solve this problem?

There are usually two cases that occur:

a) If K=2 – 10

Run the whole process of k-means algorithm anywhere between 50-1000 times depending on requirement. Then pick the value of clustering for which the cost is minimum. This prevents the algorithm from staying on the wrong local optima.

Random initialization

For i = 1 to 100 { 50 – 1000

Randomly initialize K-means.
Run K-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K$.
Compute cost function (distortion)
$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$
}

Pick clustering that gave lowest cost $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$

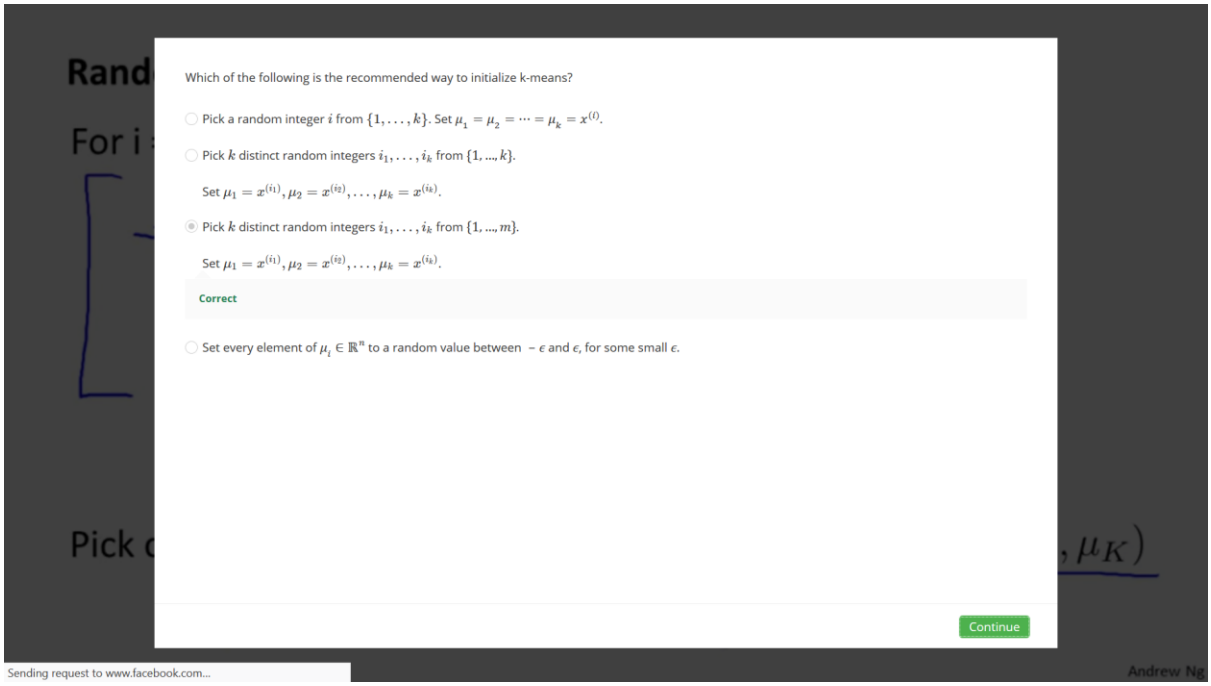
K = 2 - 10

b) If K>10

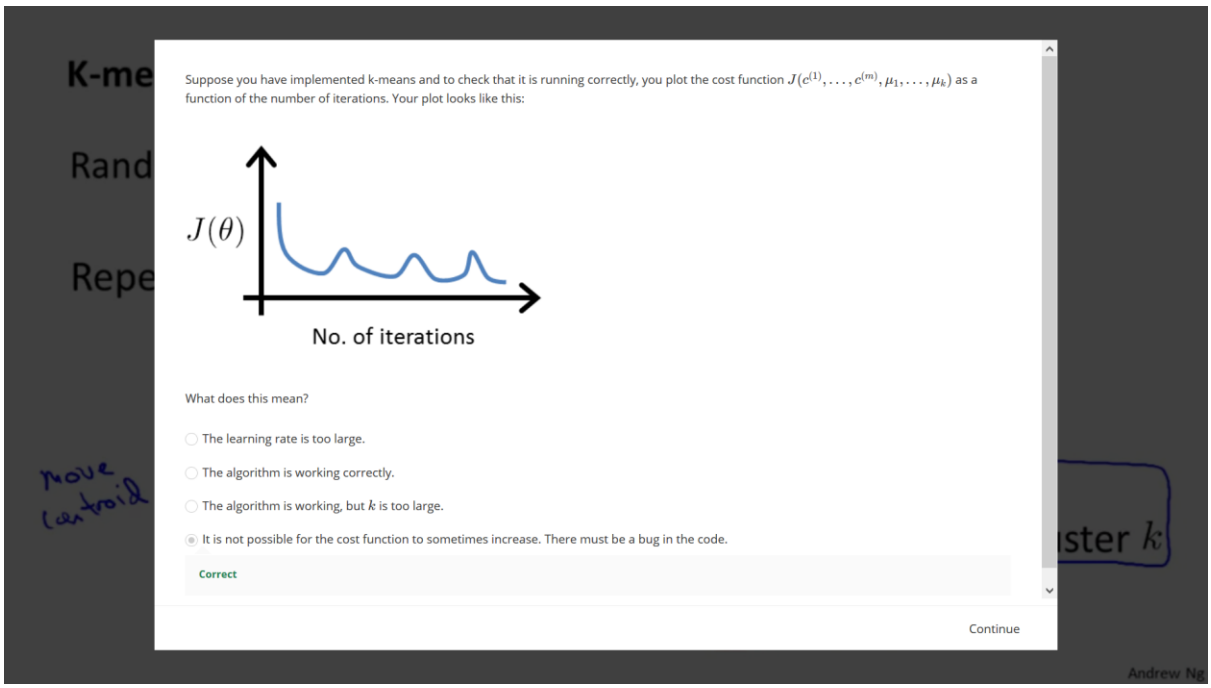
Usually doing the whole process pertaining to the algorithm once with single time initialisation of mu values produces the correct result. Although doing the whole process pertaining to the algorithm multiple times can lead to a little better result.

Questions:

Q1:

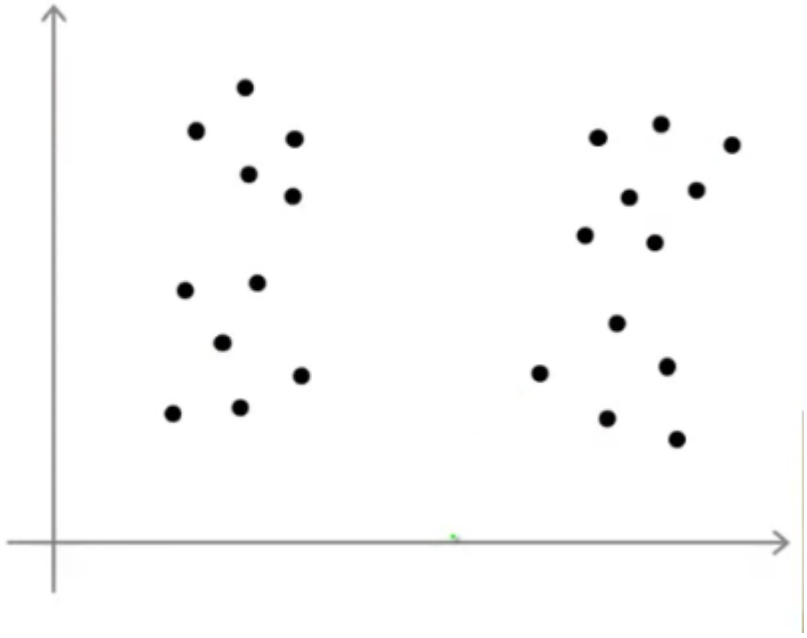


Q2:



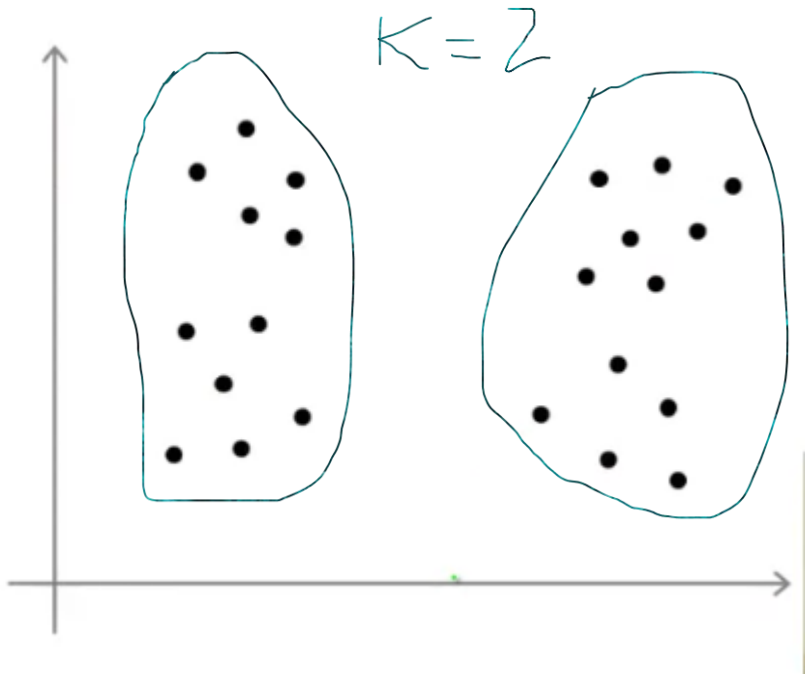
Choosing the value of K:

There is no particular way of choosing the value of K. Consider the figure below:

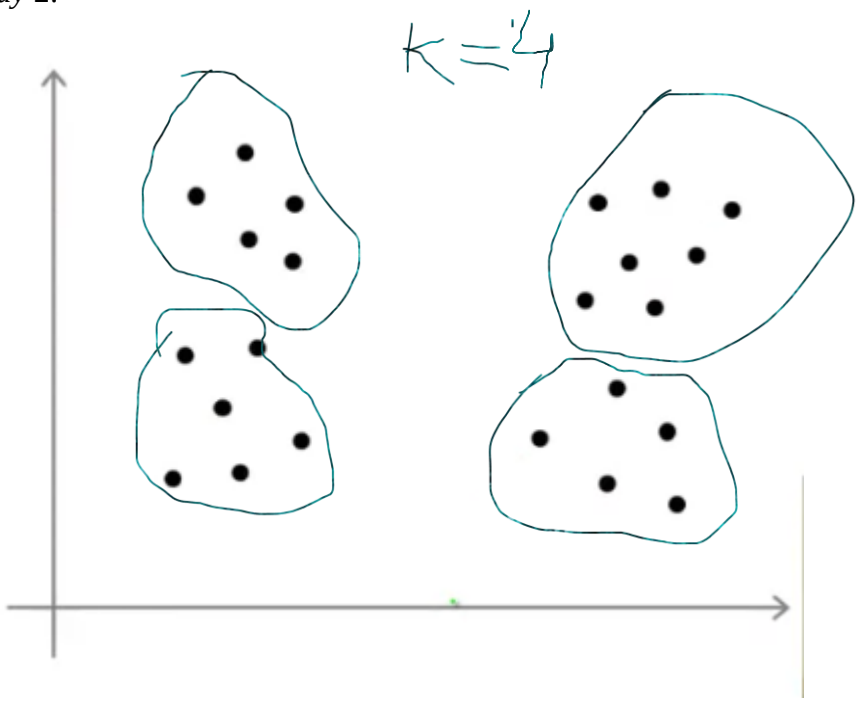


Now this can be made into clusters in many ways, two of which are shown below:

Way 1*:*

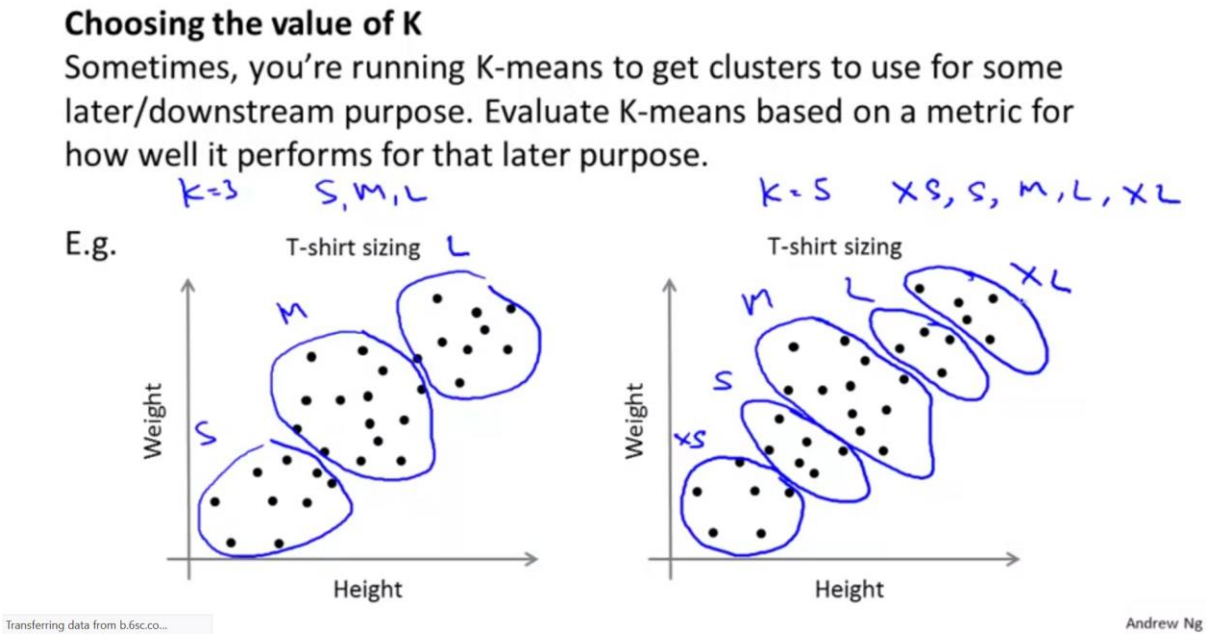


Way 2:



So, we see that there is no particular correct way of choosing a value of K and usually we choose a value by hand. Though there are two ways that may be used to choose a value of K:

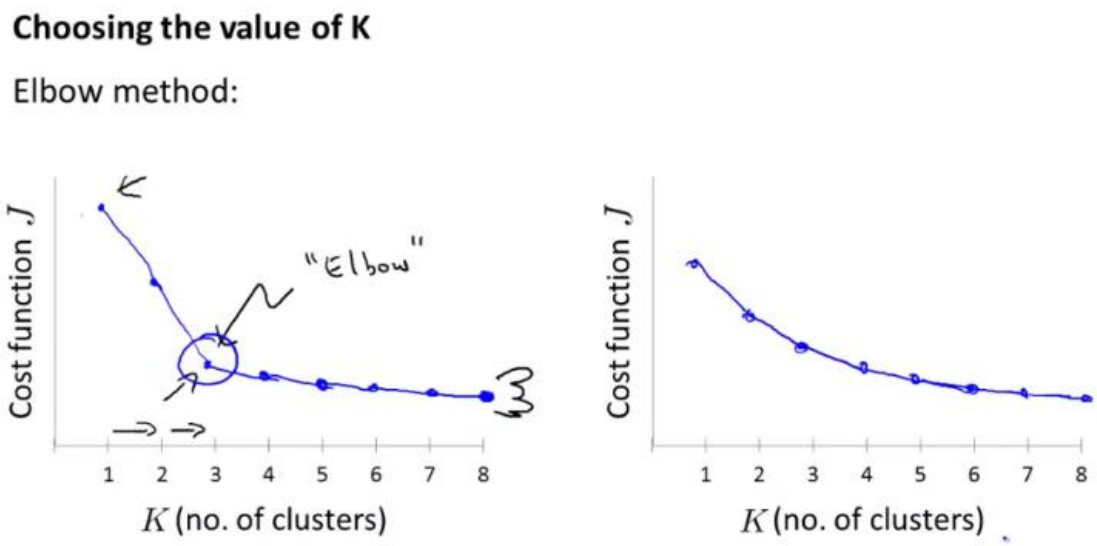
1. Choosing K value according to need.



Now, if you are a T-shirt seller then you can choose a value of K according to your need. For example if you feel that using K=3 which classifies the t-shirts into 3 groups that is S,M or L

is more useful than using K=5 which classifies the t-shirts into 5 groups that is XS,S,M,L or XL then you can choose K=3.

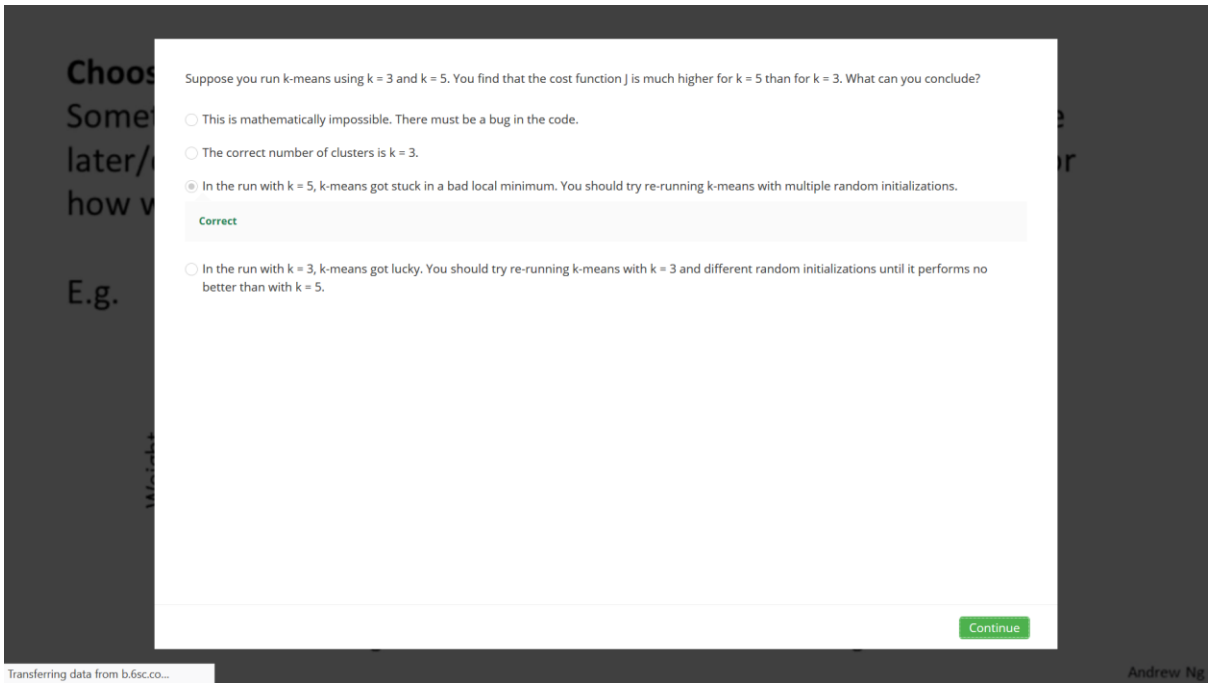
2. Using the elbow method:



If we plot cost function vs K and get the graph on the left side of above figure then the elbow point can be taken as value of K.

However usually we get the graph shown on the right side of the figure and there is no clear elbow. Hence this method is not always useful.

Questions:



1. For which of the following tasks might K-means clustering be a suitable algorithm? Select all that apply.

1 point

- [ ] Given historical weather records, predict if tomorrow's weather will be sunny or rainy.
- [x] From the user usage patterns on a website, figure out what different groups of users exist.
- [ ] Given many emails, you want to determine if they are Spam or Non-Spam emails.
- [x] Given a set of news articles from many different news websites, find out what are the main topics covered.

2. Suppose we have three cluster centroids $\mu_1 = \begin{matrix} 1 \\ 2 \end{matrix}$, $\mu_2 = \begin{matrix} -3 \\ 0 \end{matrix}$ and $\mu_3 = \begin{matrix} 4 \\ 2 \end{matrix}$. Furthermore, we have a training example $x^{(i)} = \begin{matrix} 3 \\ 1 \end{matrix}$. After a cluster assignment step, what will $c^{(i)}$ be?

- ( ) $c^{(i)} = 2$
- (•) $c^{(i)} = 3$
- ( ) $c^{(i)} = 1$
- ( ) $c^{(i)}$ is not assigned

3. K-means is an iterative algorithm, and two of the following steps are repeatedly carried out in its inner-loop. Which two?

- [ ] Feature scaling, to ensure each feature is on a comparable scale to the others.
- [ ] Using the elbow method to choose K.
- [x] The cluster assignment step, where the parameters $c^{(i)}$ are updated.
- [x] Move the cluster centroids, where the centroids μ_k are updated.

4. Suppose you have an unlabeled dataset $\{x^{(1)}, \ldots, x^{(m)}\}$. You run K-means with 50 different random

initializations, and obtain 50 different clusterings of the

data. What is the recommended way for choosing which one of

these 50 clusterings to use?

- ( ) Always pick the final (50th) clustering found, since by that time it is more likely to have converged to a good solution.
- ( ) The answer is ambiguous, and there is no good way of choosing.
- ( ) The only way to do so is if we also have labels $y^{(i)}$ for our data.
- (•) For each of the clusterings, compute $\frac{1}{m}\sum_{i=1}^{m}||x^{(i)} - \mu_{c^{(i)}}||^2$, and pick the one that minimizes this.

5. Which of the following statements are true? Select all that apply.

1 poin

- [ ] Since K-Means is an unsupervised learning algorithm, it cannot overfit the data, and thus it is always better to have as large a number of clusters as is computationally feasible.

- [ ] The standard way of initializing K-means is setting $\mu_1 = \cdots = \mu_k$ to be equal to a vector of zeros.

- [x] If we are worried about K-means getting stuck in bad local optima, one way to ameliorate (reduce) this problem is if we try using multiple random initializations.

- [x] For some datasets, the "right" or "correct" value of K (the number of clusters) can be ambiguous, and hard even for a human expert looking carefully at the data to decide.