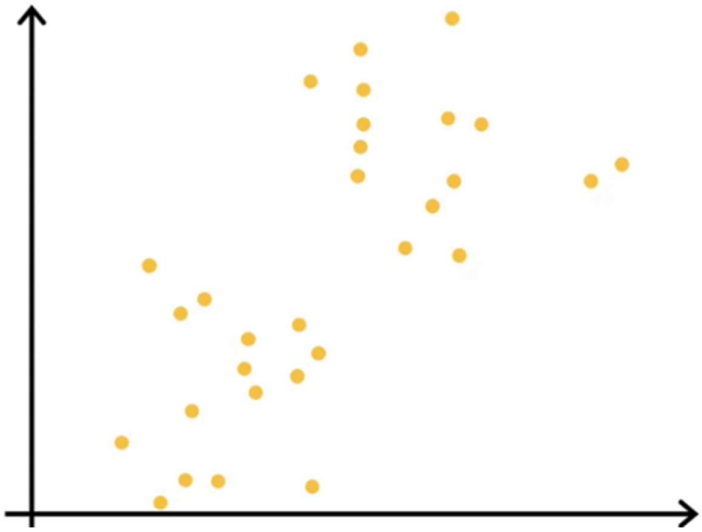
Clustering

What is clustering? A clustering algorithm looks at a number of data points and automatically finds data points that are related or similar to each other.

K-means intuition

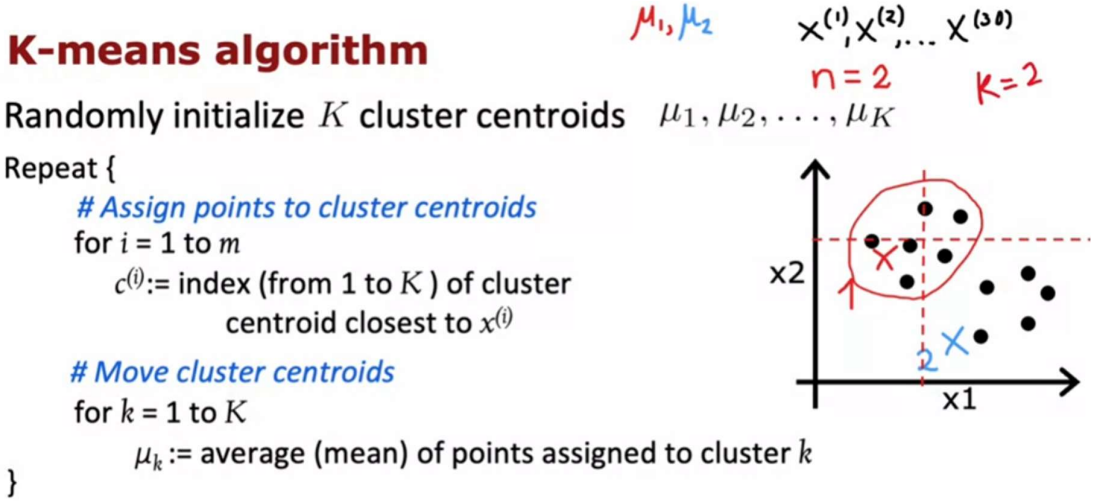


K-means algorithm

$x^{(1)}, x^{(2)}, \ldots x^{(30)}$ 30 training examples

n = 2 two features,

m = 30



Two Step process.

The first step is to assign points to clusters, centroids

In this case K=2, so cluster centroid locations are μ_1, μ_2 are vectors same dimension as training examples

Assign all data points (i= 1 to m) to cluster centroids one or two when K = 2
set c^i to be equal to the index, which can be anything from one to K of the cluster centroid closest to the training example x^i.
Mathematically you can write this out as computing the distance between x^i and Mu k. In math, the distance between two points is often written like this.

$$\| x^{(i)} - \mu_k \|$$

It is also called the L2 norm.
What you want to find is the value of k that minimizes this,

$$min_k \| x^{(i)} - \mu_k \|$$

because that corresponds to the cluster centroid Mu k that is closest to the training example x^i.
Then the value of k that minimizes this is what gets set to c^i.

Repeat {
    # Assign points to cluster centroids
    for i = 1 to m
        $c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$
            $min_k \| x^{(i)} - \mu_k \|^2$

When implementing, it is a little bit more convenient to minimize the squared distance because the cluster centroid with the smallest square distance should be the same as the cluster centroid with the smallest distance.

Second step is to move the cluster centroids
set the cluster centroid location to be updated to be the average or the mean of the points assigned to that cluster k.

$$\mu_1 = \frac{1}{4}[x^{(1)} + x^{(5)} + x^{(6)} + x^{(10)}]$$

Note: x values are vectors with two numbers (or n numbers, in this n=2)

Sometimes, cluster center will not get assigned any and one option is to eliminate that cluster centroid or randomly initialized one, next loop

Optimization objective

K-Means algorithm is also optimizing a specific cost function similar to supervise learning algorithms but it is not gradient descent.

what is the cost function for K-means

$c^{(I)}$ = index of cluster (1, 2, …, K) to which example $x^{(i)}$ is currently assigned.
ci is some number 1 to K of the index of the cluster to which training example Xi is currently assigned
And new K is the location of the cluster centroid K
μ_k = cluster centroid K

μ_c(i) = cluster centroid of cluster to which example $x^{(i)}$ has been assigned.

$x^{(10)}$ is the training example

$c^{(10)}$ is the location of cluster centroid to which $x^{(10)}$ has been assigned. (red/blue 1/2 cluster)
μ_c $^{(10)}$ is the location of the cluster centroid $x^{(10)}$

Cost function

- $J(c^{(1)}, \dots, c^{(m)}, \mu_1, .., \mu_K) = \frac{1}{m}\sum_{i=1}^{m} \|x^{(i)} - \mu_{c^{(i)}}\|^2$

$\min_{c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K} J(c^{(1)}, \dots, c^{(m)}, \mu_1, .., \mu_K)$

Distortion

the cost function good for Kenyans is the average squared distance between every training example XI. And the location of the cluster centroid to which the training example exile has been assigned.

What the K-means algorithm is doing is trying to find assignments of points of clusters centroid as well as find locations of clusters centroid that minimizes the squared distance.

This cost function J also has a name in the literature called the distortion function.

Cost function for K-means

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, .., \mu_K) = \frac{1}{m}\sum_{i=1}^{m} \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

Repeat {

 # Assign points to cluster centroids

for i = 1 to m

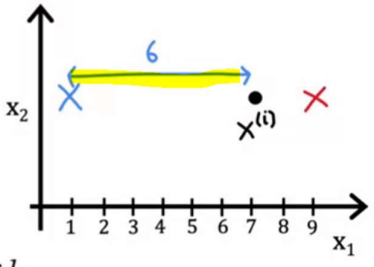
$c^{(i)}$:= index of cluster centroid closest to $x^{(i)}$

 # Move cluster centroids

for k = 1 to K

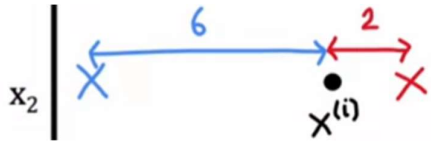
μ_k := average of points in cluster k

}



if you have two clusters centroid say 1 and 2 anda single training example, XI. If you were to sign it to cluster central one, this square distance here ($||x^{(i)}$ - μ_c(i) $||^2$ would be this large distance (6), well squared.

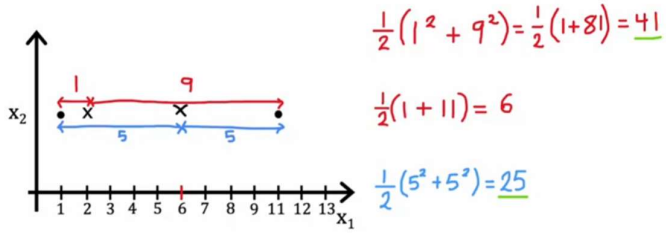
if you were to sign it to cluster centroid 2 then this square distance would be the square of this much smaller distance.



So if you want to minimize this term, you will take X I and assign it to the closer centroid
The second step of the K-means algorithm that is to move to clusters centroids
choosing mu K to be average and the mean of the points assigned is the choice of these terms mu that will minimize this expression.

$\|x^{(i)} - \mu_{c^{(i)}}\|^2$

Moving the centroid



Suppose there are two cluster centroids x
The above calculation shows that blue value minimize the distance square and optimizing the cost function

Every iteration, distortion cost function should go down. If it remains the same, it usually means K-means has converged.

Initializing K-means

You can take multiple attempts at the initial guesses with mu_1through mu_K. That will result in your finding a better set of clusters

K-means algorithm

Step 0: Randomly initialize K cluster centroids $\mu_1, \mu_1, \ldots, \mu_k$

Repeat {
Step 1: Assign points to cluster centroids
Step 2: Move cluster centroids
}

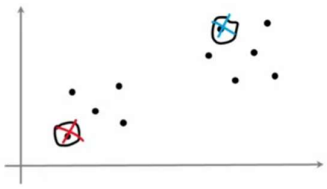
choose the number of cluster central's K to be lessened to training examples m

Random initialization

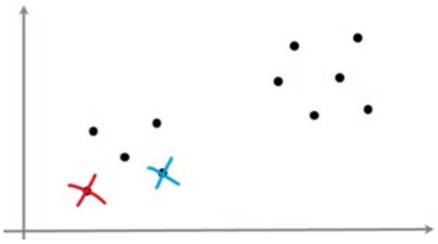
Choose $K < m$

Randomly pick K training examples.

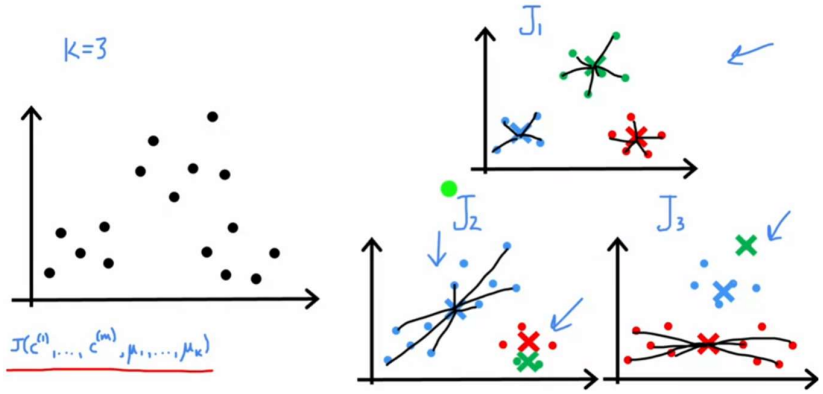
Set $\mu_1, \mu_1, \ldots, \mu_k$ equal to these K examples.



k = 2
Previous illustrations might show initializing the cluster centroids mu_1 and mu_2 to be just random points rather than sitting on top of specific training examples.
You might end up initialising these two cluster centroids depending how random initialization



Three clusters and three different initialization



Top right is pretty good choice.
lower middle, turns out to be a local optima, in which K-means is trying to minimize the distortion cost function, just happened to get stuck in a local minimum.

So if you want to give k means multiple shots at finding the best local optimum. If you want to try multiple random initialization, so give it a better chance of finding this good clustering up on top.
run it multiple times and then to try to find the best local optima.

calculate the cost, top right is the lowest

Random initialization

50-1000

For $i = 1$ to 100 {

k random examples

Randomly initialize K-means.

Run K-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \mu_1, \ldots, \mu_k$

Computer cost function (distortion)

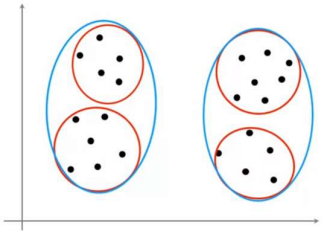
$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \mu_1, \ldots, \mu_k)$

}

Pick set of clusters that gave lowest cost J

Choosing the number of clusters

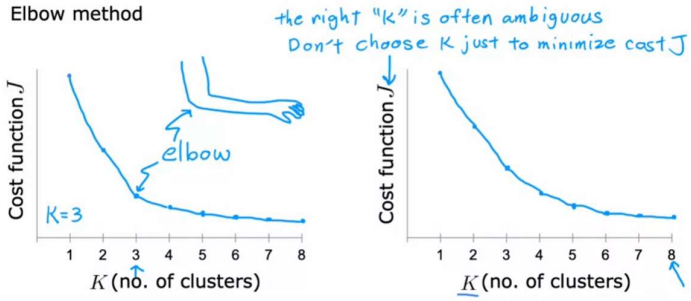
What is the right value of K?



2 or 4?

one way to try to choose the value of K is called the **elbow method** and what that does is you would run K-means with a variety of values of K and plot the cost function or the distortion function J as a function of the number of clusters. What you find is that when you have very few clusters, say one cluster, the distortion function or the cost function J will be high and as you increase the number of clusters, it will go down,

Choosing the value of K

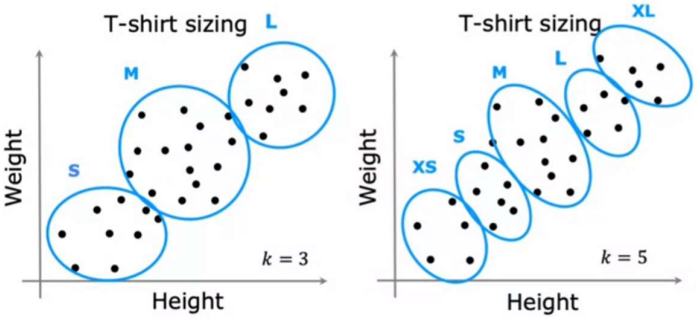


cost function is decreasing rapidly until we get to 3 clusters but the decrease is more slowly after that.
Andrew does not use the elbow method …

one technique that does not work is to choose K so as to minimize the cost function J because doing so would cause you to almost always just choose the largest possible value of K because having more clusters will pretty much always reduce the cost function J. Choosing K to minimize the cost function J is not a good technique.

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.



What I usually do and what I recommend you do is to evaluate K-means based on how well it performs for that later downstream purpose.
There will be extra costs as well associated with manufacturing and shipping five types of t-shirts instead of three different types of t-shirts. What I would do in this case is to run K-means with K =

3 and K = 5 and then look at these two solutions to see based on the trade-off between fits of t-shirts with more sizes, results in better fit versus the extra cost of making more t-shirts where making fewer t-shirts is simpler and less expensive to try to decide what makes sense for the t-shirt business.

Clustering

1. Question 1

Which of these best describes unsupervised learning?

- A form of machine learning that finds patterns using unlabeled data (x). ANS
- A form of machine learning that finds patterns without using a cost function.
- A form of machine learning that finds patterns using labeled data (x, y)
- A form of machine learning that finds patterns in data using only labels (y) but without any inputs (x) .

Correct

Unsupervised learning uses unlabeled data. The training examples do not have targets or labels "y". Recall the T-shirt example. The data was height and weight but no target size.

2. Question 2

Which of these statements are true about K-means? Check all that apply.

- If each example x is a vector of 5 numbers, then each cluster centroid μ_k is also going to be a vector of 5 numbers. ANSWER

Correct

The dimension of μ_k matches the dimension of the examples.

- The number of cluster assignment variables $c^{(i)}$ is equal to the number of training examples. ANSWER

Correct

$c^{(i)}$ describes which centroid example (i) is assigned to.

- If you are running K-means with $K=3$ clusters, then each $c^{(i)}$ should be 1, 2, or 3. ANSWER

Correct

$c^{(i)}$ describes which centroid example(i) is assigned to. If $K=3$, then $c^{(i)}$ \would be one of 1,2 or 3 assuming counting starts at 1.

- The number of cluster centroids μ_k is equal to the number of examples.

3. Question 3

You run K-means 100 times with different initializations. How should you pick from the 100 resulting solutions?

- Pick the one with the lowest cost J ANSWER
- Average all 100 solutions together.
- Pick the last one (i.e., the 100th random initialization) because K-means always improves over time
- Pick randomly -- that was the point of random initialization.

Correct

K-means can arrive at different solutions depending on initialization. After running repeated trials, choose the solution with the lowest cost.

4. Question 4

You run K-means and compute the value of the cost function $J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K)$ after each iteration. Which of these statements should be true?

- There is no cost function for the K-means algorithm.
- The cost will either decrease or stay the same after each iteration. ABSWER
- Because K-means tries to maximize cost, the cost is always greater than or equal to the cost in the previous iteration.
- The cost can be greater or smaller than the cost in the previous iteration, but it decreases in the long run.

Correct

The cost never increases. K-means always converges.

5. Question 5

In K-means, the elbow method is a method to

- Choose the number of clusters K ANSWER
- Choose the best random initialization
- Choose the maximum number of examples for each cluster
- Choose the best number of samples in the dataset

Correct

The elbow method plots a graph between the number of clusters K and the cost function. The 'bend' in the cost curve can suggest a natural value for K. Note that this feature may not exist or be significant in some data sets.