Clustering - Optimization Objective

Most of the supervised learning algorithms we've seen, things like linear regression, logistic regression, and so on, all of those algorithms have an optimization objective or some cost function that the algorithm was trying to minimize. It turns out that k-means also has an optimization objective or a cost function that it's trying to minimize.

- Knowing this is useful because it helps for debugging and helps find better clusters.

K-means optimization objective

While K-means is running we keep track of two sets of variables:

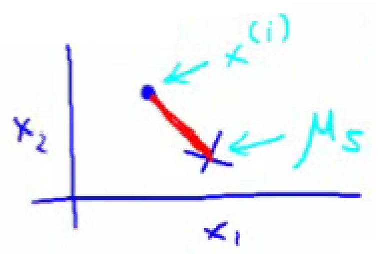
- $c^{(i)}$ = index of clusters $(1, 2, \dots, K)$ to which example $x^{(i)}$ is currently assigned
- μ_k = cluster centroid k ($\mu_k \in \mathbb{R}^n$)
  - For k-means we use capital K to denote the total number of clusters
  - So lower case k is going to be an index into the cluster centroids $k \in \{1, 2, \dots, k\}$
  - So these the centroids which exist in the training data space
- $\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned
  - We could look up that example $x^{(i)}$ is indexed to cluster k (using the $c^{(i)}$ vector).
  - Then look up the value associated with cluster k in the μ vector (i.e. what are the features associated with $\mu_{c^{(i)}}$)
  - But instead, for easy description, we have this variable which gets exactly the same value
    - Lets say $x^{(i)}$ as been assigned to cluster 5, means that:
    - $c^{(i)}$, $\mu_c^{(i)} = \mu_5$

Optimization Objective

Out with this notation, we're now ready to write out what is the optimization objective of the k-means clustering algorithm:

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

- i.e. squared distances between training example $x^{(i)}$ and the cluster centroid to which $x^{(i)}$ has been assigned to.
  - This is just what we've been doing, as the visual description below shows:



  - The red line here shows the distances between the example $x^{(i)}$ and the cluster to which that example has been assigned μ_5
    - Means that when the example is very close to the cluster, the cost function is small
    - When the cluster is very far away from the example, the cost function is large

- This is sometimes called the **distortion** (or **distortion cost function**)

- So we are finding the values which minimizes this function:

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

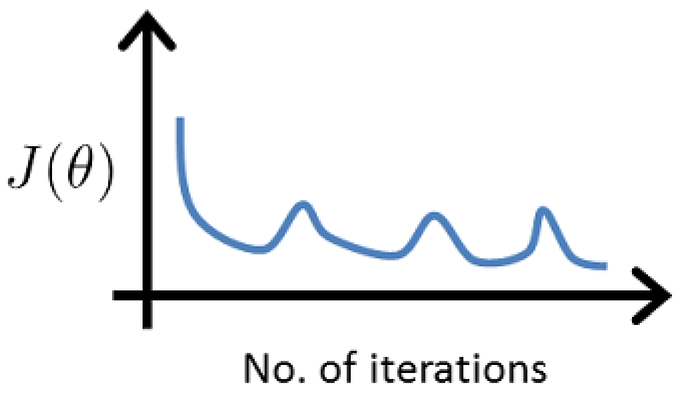
K-means Algorithm

Randomly initialize K clusters centroids $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^n$

```
Repeat {
        for $i$ = 1 to $m$
            $c^{(i)}$ := index (from 1 to $K$ ) of cluster centroid
                    closest to $x^{(i)}$
        for $k$ = 1 to $K$
            $\mu_k$ := average (mean) of points assigned to cluster $k$
        }
```

- 1st **for loop** - The cluster assigned step is minimizing $J(\dots)$ with respect to $c^{(1)}, c^{(2)}, \dots, c^{(i)}$
  - i.e. find the centroid closest to each example
- 2nd **for loop** - The move centroid step
  - We can show this step is choosing the values of μ which minimizes $J(\dots)$ with respect to $\mu_1, \mu_2, \dots, \mu_K$
- So, we're partitioning the algorithm into two parts
  - First part minimizes the $c^{(i)}$ variables
  - Second part minimizes the J variables
  - then it keeps on iterating

Video Question: Suppose you have implemented k-means and to check that it is running correctly, you plot the cost function $J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_k)$ as a function of the number of iterations. Your plot looks like this:



What does this mean?

- The learning rate is too large.
- The algorithm is working correctly.
- The algorithm is working, but k is too large.

> It is not possible for the cost function to sometimes increase. There must be a bug in the code.