## Clustering - Random Initialization

How to initialize K-means, and more importantly how to make K-means avoid local optima as well.

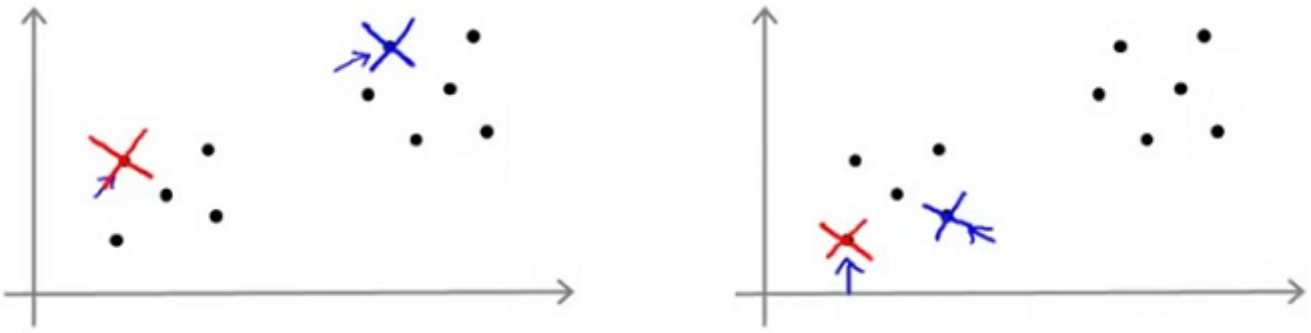
Consider clustering algorithm:

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

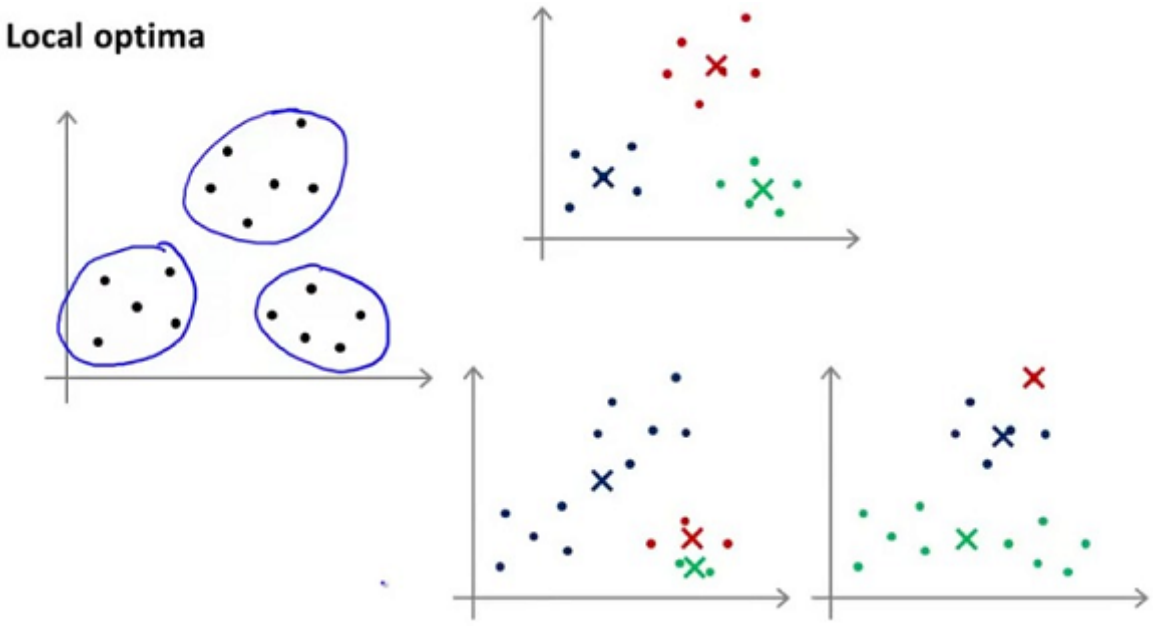
Repeat {
for $i$ = 1 to $m$
$c^{(i)}$ := index (from 1 to $K$) of cluster centroid closest to $x^{(i)}$
for $k$ = 1 to $K$
$\mu_k$ := average (mean) of points assigned to cluster $k$
}

**Should have $K < m$:** Have number of centroids set to less than number of examples $K < m$ (It would be really weird to run $K$-means with a number of cluster centroids that's equal or greater than the number of examples we have).

- Randomly pick $K$ training examples.
- Set $\mu_1, \dots, \mu_K$ equal to these $K$ examples.
  - i.e. 2 clusters (K = 2)
    - $\mu_1 = x^{(i)}, \mu_2 = x^{(j)}$

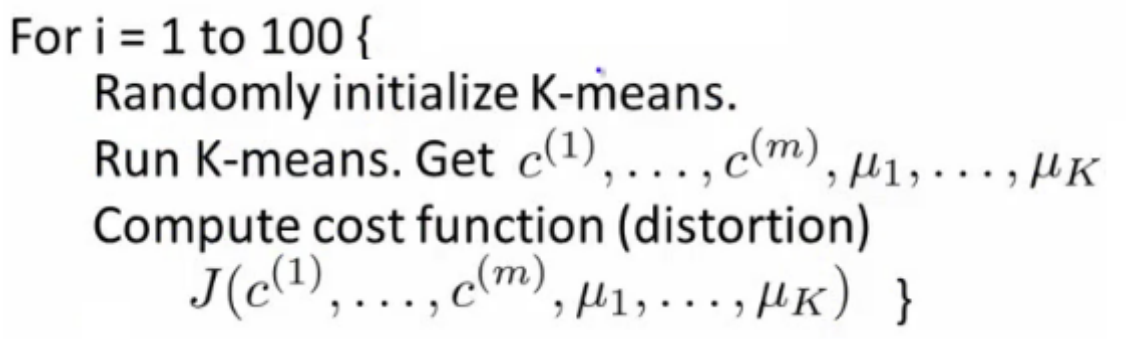


$K$-means can converge to different solutions depending on the initialization setup (risk of local optimum)



So, if we're worried about $K$-means getting stuck in local optima, if we want to increase the odds of $K$-means finding the best possible clustering, what we can do, is try multiple, random initializations - many same results are likely to indicate a global optimum.

Algorithmically we can do this as follows:




- A typical number of times to initialize K-means is 50 - 1000
- Randomly initialize $K$-means
  - For each 100 random initialization run $K$-means
  - Then compute the distortion on the set of cluster assignments and centroids at convergent
  - End with 100 ways of cluster the data,
  - Pick clustering that gave lowest cost, the lowest distortion $J\left(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K\right)$
- If we're running $K$ means with 2-10 clusters can help find better global optimum
  - If $K$ is larger than 10, then multiple random initializations are less likely to be necessary
  - First solution is probably good enough (better granularity of clustering)

**Video Question:** Which of the following is the recommended way to initialize k-means?

- Pick a random integer $i$ from $\{1, \ldots, k\}$. Set $\mu_1 = \mu_2 = \cdots = \mu_k = x^{(i)}$.
- Pick $k$ distinct random integers $i_1, \ldots, i_k$ from $\{1, \ldots, k\}$. Set $\mu_1 = x^{(i_1)}, \mu_2 = x^{(i_2)}, \ldots, \mu_k = x^{(i_k)}$.

> Pick $k$ distinct random integers $i_1, \ldots, i_k$ from $\{1, \ldots, m\}$. Set $\mu_1 = x^{(i_1)}, \mu_2 = x^{(i_2)}, \ldots, \mu_k = x^{(i_k)}$.

- Set every element of $\mu_i \in \mathbb{R}^n$ to a random value between –$\epsilon$– and $\epsilon$, for some small $\epsilon$.