K-means聚类算法

簇分配步骤：

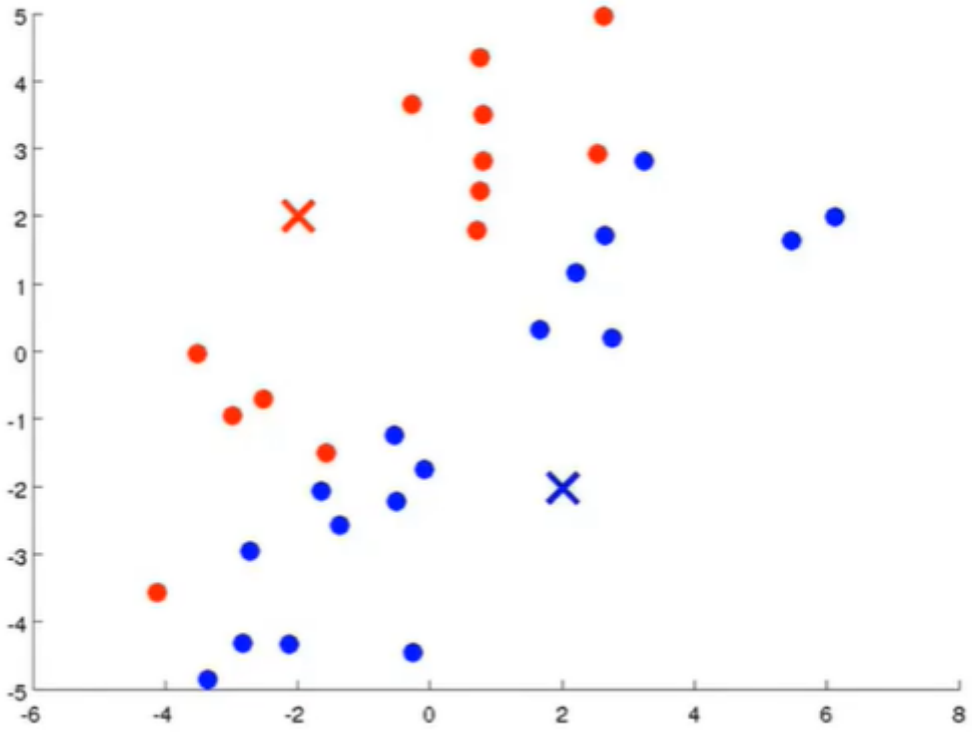
1.选取好分类点，离某一个分类点更近的点都归类于那一类。

2.移动聚类中心：计算所有某一类的点的平均位置，再将那一类的分类点移动到平均位置。

3.再次将离新的分类点更近的点分配到那一类。

4.再次移动聚类中心

....直到聚类中心不发生改变。



K-means algorithm

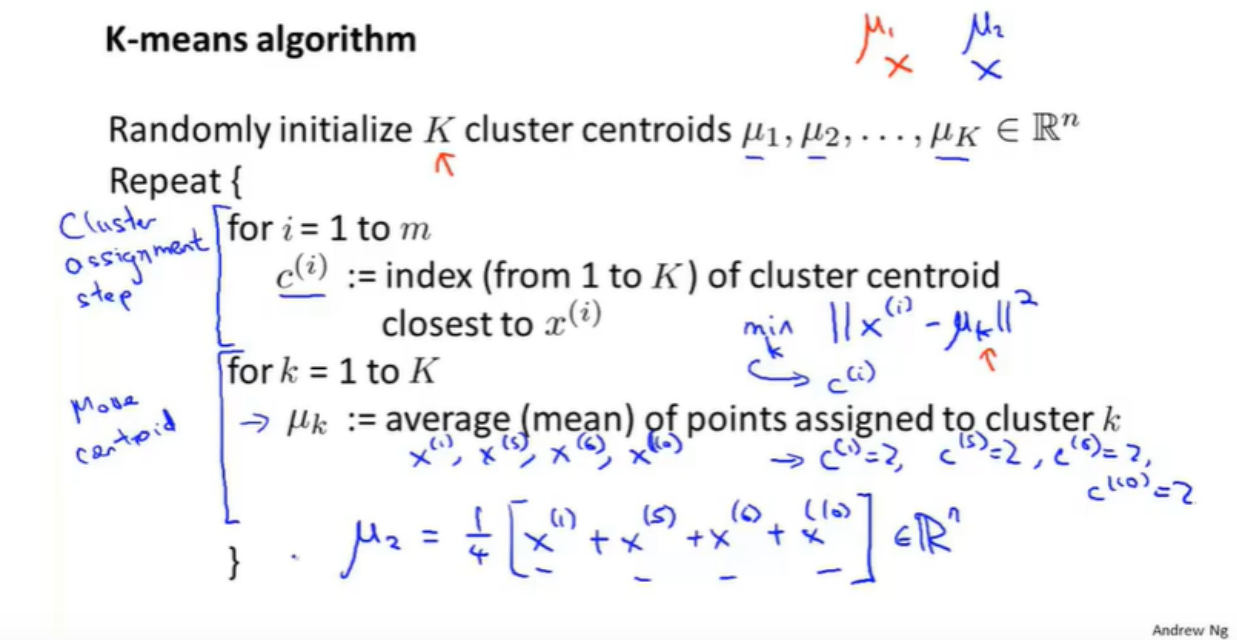
Input:

- K (number of clusters)
- Training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

$x^{(i)} \in \mathbb{R}^n$ (drop $x_0 = 1$ convention)

K-means接受两个输入，k是想要的簇个数，还有一个数据集

规定x(i)是一个n维实数向量，所以训练样本是n维向量



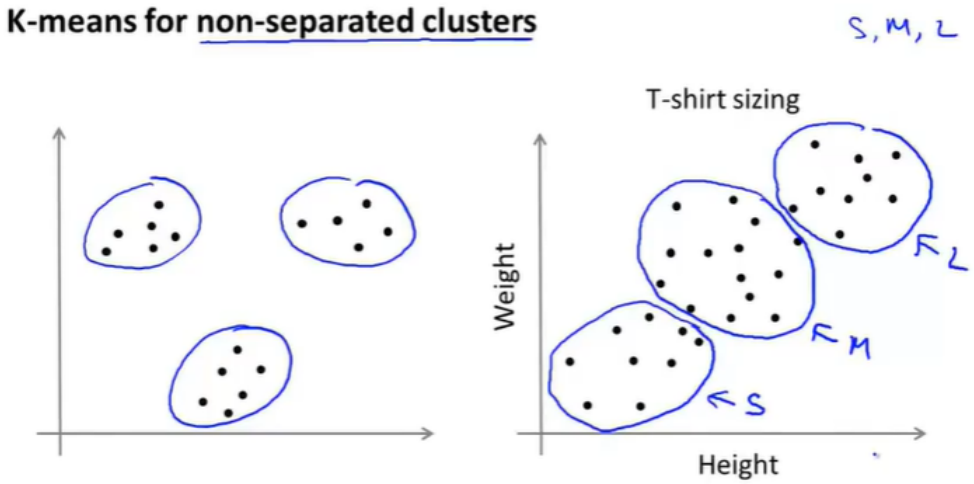
第一步随机初始化k个聚类中心

内循环中第一步进行簇分配，将每个样本根据它离聚类中心的距离，将它染成对应的颜色。c(i)是1-k之间的数，表示它是属于哪个簇。

另一步是移动聚类中心，将聚类中心分配到该簇中所有点的均值处。

如果出现了一个没有点的聚类中心，这时可以直接移除这个聚类中心，但不会有k个簇，而是有k-1个簇。如果我们确实需要k个簇，那么我们可以重新初始化这个没有点的聚类中心。

K均值用于市场划分：



K-means优化目标函数

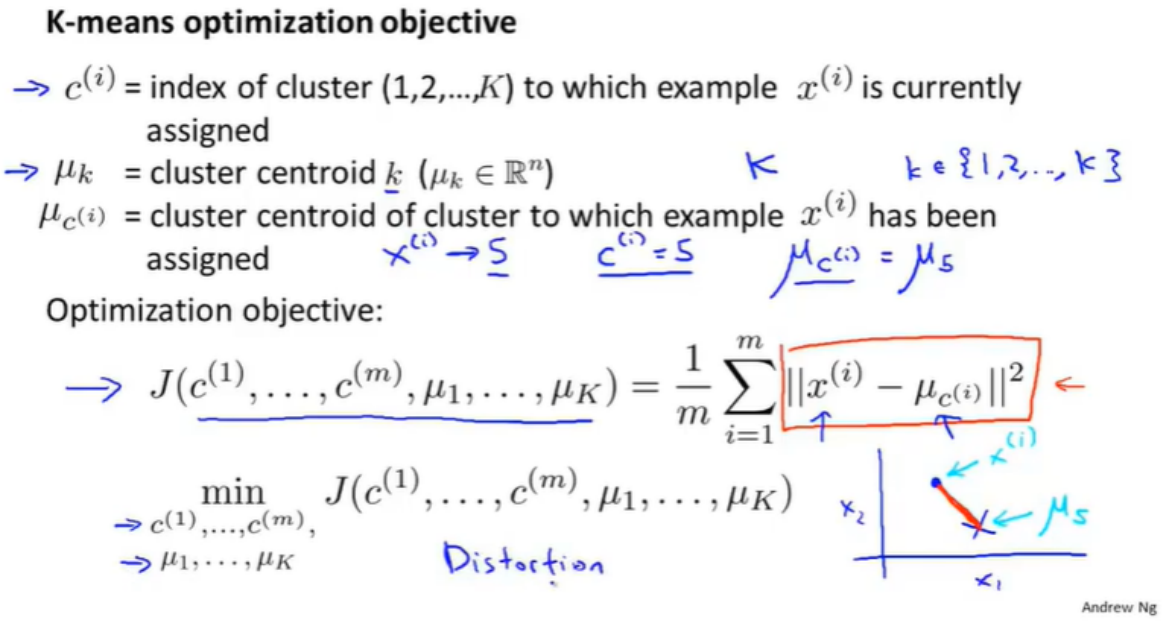
K-means中，大写的K表示簇的数量，小写的k表示聚类中心的下标(取值范围1-K)。

c(i)表示xi所属的聚类中心的索引值

μk表示第k个聚类中心的位置

μc(i)表示xi所属的聚类中心的位置

代价函数是每个数据样本到其聚类中心的距离的平方的加和。代价函数也被叫做算法的失真(distortion)函数。



第一步，簇分配步骤就是在保持聚类中心不变，选择c1....cm来最小化代价函数，把每个点分配给离它最近的距离中心，这样就可以减少它到聚类中心的距离。

第二步，移动聚类中心，就是选择μ值来最小化代价函数。

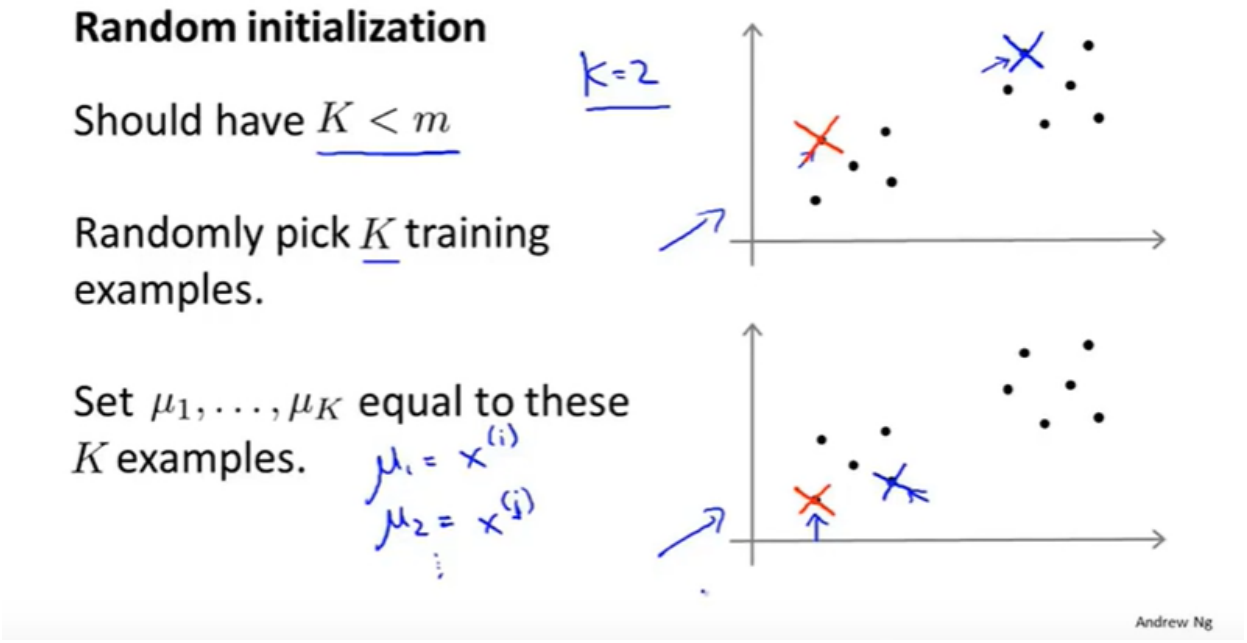
然后保持迭代。

K-means随机初始化（如何闭开局部最优）

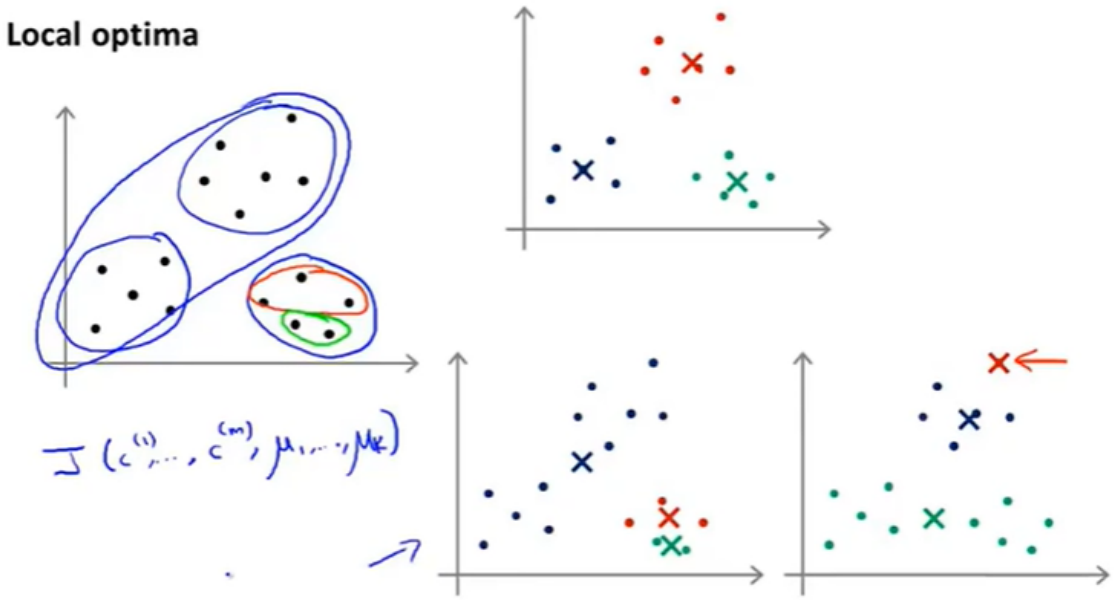
如何初始化K个随机聚类中心?

通常选取的方法：

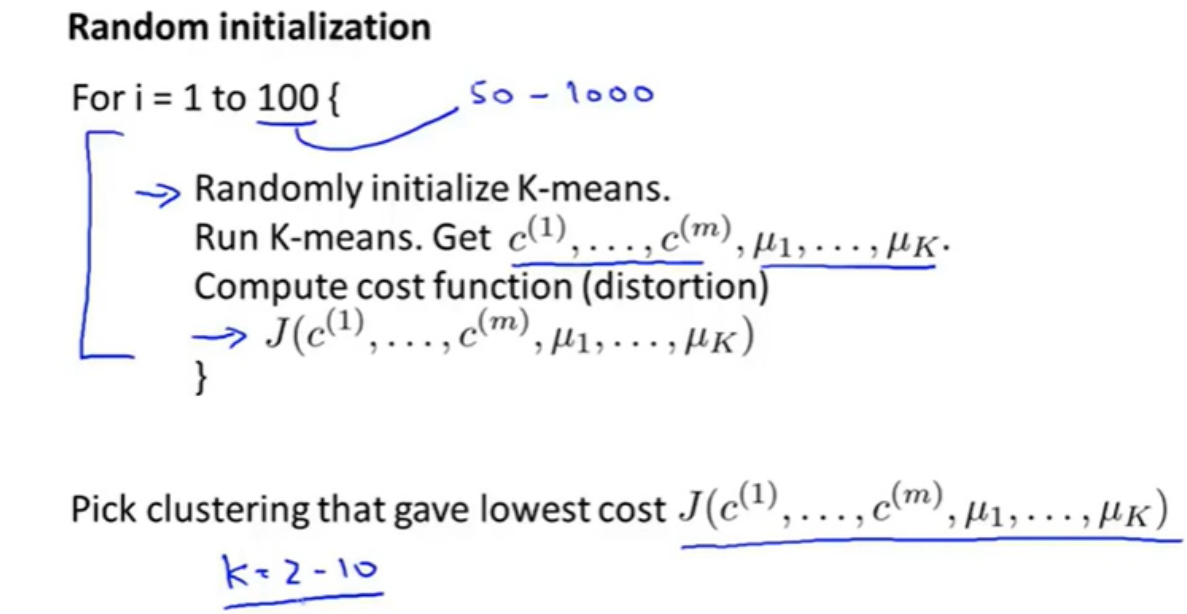
随机将k个训练样本当作聚类中心。



K-means可能会出现效果很不好的局部最优解，也就是J失真函数的局部最优解。



为了避免这种情况，我们可以多次随机初始化，并多次运行kmeans算法，以此来保证获得一个较好的结果。



在聚类数K较小(2-10)时，多次随机初始化可以有很好的效果，但当K较大，多次随机初始化帮助很有限。

K-means 选择聚类数量K

最常见的方法：根据图像手动选择

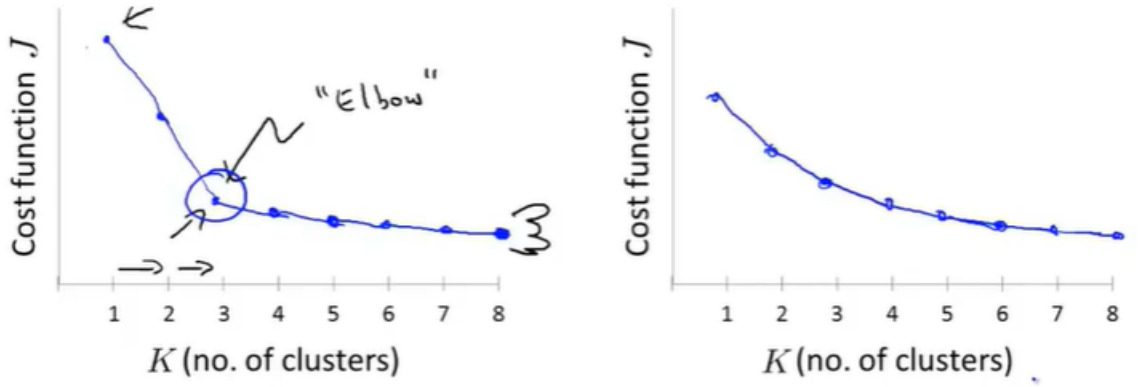
肘部法则：

“肘部”：指畸变值（代价函数的值）开始下降的缓慢的点。K选择肘部的点。如图一，K=3拐点。

但有些时候图像曲线均匀下降，很难找到清晰的拐点，如图二。

Choosing the value of K

Elbow method:



另一种思路：看哪个聚类数量能更好地应用于后续目的。比如体恤尺寸的例子。

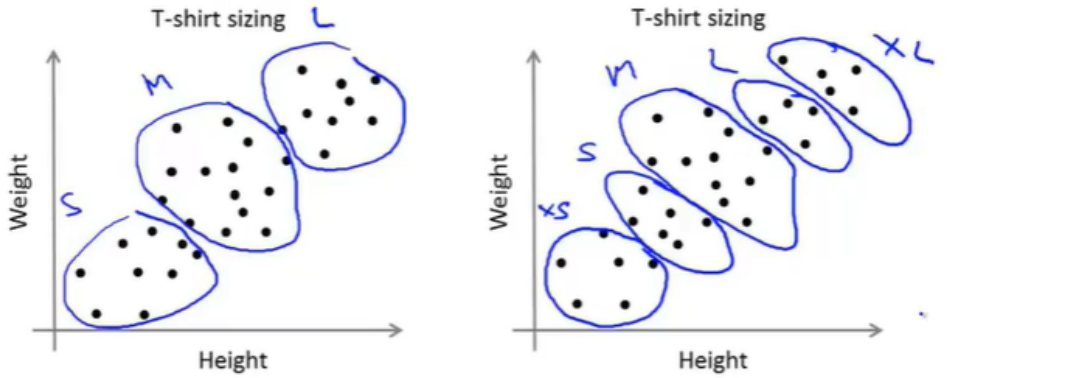
Choosing the value of K

Sometimes, you're running K-means to get clusters to use for some later/downstream purpose. Evaluate K-means based on a metric for how well it performs for that later purpose.

K=3 S, M, L

K=5 XS, S, M, L, XL

E.g.



Andrew Ng