K-means 聚类算法

JerryLead

csxulijie@gmail.com

K-means 也是聚类算法中最简单的一种了，但是里面包含的思想却是不一般。最早我使用并实现这个算法是在学习韩爷爷那本数据挖掘的书中，那本书比较注重应用。看了 Andrew Ng 的这个讲义后才有些明白 K-means 后面包含的 EM 思想。

聚类属于无监督学习，以往的回归、朴素贝叶斯、SVM 等都是有类别标签 y 的，也就是说样例中已经给出了样例的分类。而聚类的样本中却没有给定 y，只有特征 x，比如假设宇宙中的星星可以表示成三维空间中的点集(x, y, z)。聚类的目的是找到每个样本 x 潜在的类别 y，并将同类别 y 的样本 x 放在一起。比如上面的星星，聚类后结果是一个个星团，星团里面的点相互距离比较近，星团间的星星距离就比较远了。

在聚类问题中，给我们的训练样本是$\{x^{(1)}, \dots, x^{(m)}\}$，每个$x^{(i)} \in \mathbb{R}^n$，没有了 y。

K-means 算法是将样本聚类成 k 个簇（cluster），具体算法描述如下：

1、 随机选取 k 个聚类质心点（cluster centroids）为$\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$。

2、 重复下面过程直到收敛 {

对于每一个样例 i，计算其应该属于的类

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

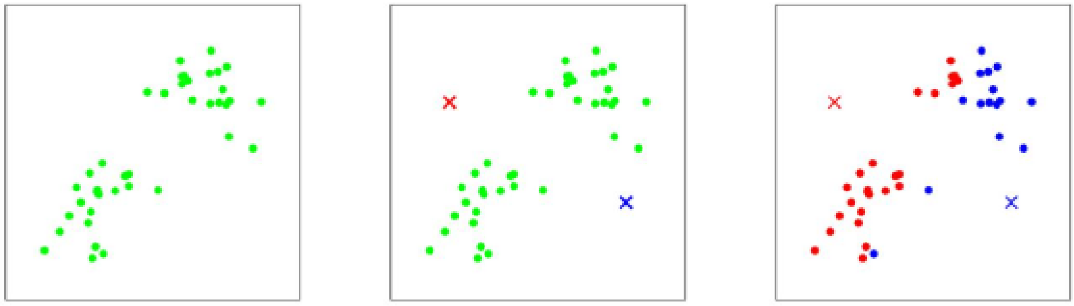
对于每一个类 j，重新计算该类的质心

$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

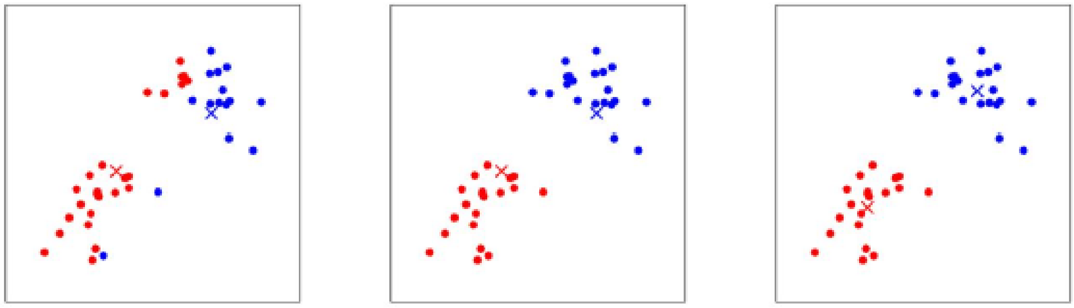
}

K 是我们事先给定的聚类数，$c^{(i)}$代表样例 i 与 k 个类中距离最近的那个类，$c^{(i)}$的值是 1 到 k 中的一个。质心μ_j代表我们对属于同一个类的样本中心点的猜测，拿星团模型来解释就是要将所有的星星聚成 k 个星团，首先随机选取 k 个宇宙中的点（或者 k 个星星）作为 k 个星团的质心，然后第一步对于每一个星星计算其到 k 个质心中每一个的距离，然后选取距离最近的那个星团作为$c^{(i)}$，这样经过第一步每一个星星都有了所属的星团；第二步对于每一个星团，重新计算它的质心μ_j（对里面所有的星星坐标求平均）。重复迭代第一步和第二步直到质心不变或者变化很小。

下图展示了对 n 个样本点进行 K-means 聚类的效果，这里 k 取 2。



(a) (b) (c)



(d) (e) (f)

K-means 面对的第一个问题是如何保证收敛，前面的算法中强调结束条件就是收敛，可以证明的是 K-means 完全可以保证收敛性。下面我们定性的描述一下收敛性，我们定义畸变函数（distortion function）如下：

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

J 函数表示每个样本点到其质心的距离平方和。K-means 是要将 J 调整到最小。假设当前 J 没有达到最小值，那么首先可以固定每个类的质心μ_j，调整每个样例的所属的类别$c^{(i)}$来让 J 函数减少，同样，固定$c^{(i)}$，调整每个类的质心μ_j也可以使 J 减小。这两个过程就是内循环中使 J 单调递减的过程。当 J 递减到最小时，μ和 c 也同时收敛。（在理论上，可以有多组不同的μ和 c 值能够使得 J 取得最小值，但这种现象实际上很少见）。

由于畸变函数 J 是非凸函数，意味着我们不能保证取得的最小值是全局最小值，也就是说 k-means 对质心初始位置的选取比较感冒，但一般情况下 k-means 达到的局部最优已经满足需求。但如果你怕陷入局部最优，那么可以选取不同的初始值跑多遍 k-means，然后取其中最小的 J 对应的μ和 c 输出。

下面累述一下 K-means 与 EM 的关系，首先回到初始问题，我们目的是将样本分成 k 个类，其实说白了就是求每个样例 x 的隐含类别 y，然后利用隐含类别将 x 归类。由于我们事先不知道类别 y，那么我们首先可以对每个样例假定一个 y 吧，但是怎么知道假定的对不对呢？怎么评价假定的好不好呢？我们使用样本的极大似然估计来度量，这里是就是 x 和 y 的联合分布 P(x,y)了。如果找到的 y 能够使 P(x,y)最大，那么我们找到的 y 就是样例 x 的最佳类别了，x 顺手就聚类了。但是我们第一次指定的 y 不一定会让 P(x,y)最大，而且 P(x,y)还依赖于其他未知参数，当然在给定 y 的情况下，我们可以调整其他参数让 P(x,y)最大。但是调整完参数后，我们发现有更好的 y 可以指定，那么我们重新指定 y，然后再计算 P(x,y)最大时的参数，反复迭代直至没有更好的 y 可以指定。

这个过程有几个难点，第一怎么假定 y？是每个样例硬指派一个 y 还是不同的 y 有不同的概率，概率如何度量。第二如何估计 P(x,y)，P(x,y)还可能依赖很多其他参数，如何调整里面的参数让 P(x,y)最大。这些问题在以后的篇章里回答。

这里只是指出 EM 的思想，E 步就是估计隐含类别 y 的期望值，M 步调整其他参数使得在给定类别 y 的情况下，极大似然估计 P(x,y)能够达到极大值。然后在其他参数确定的情况下，重新估计 y，周而复始，直至收敛。

上面的阐述有点费解，对应于 K-means 来说就是我们一开始不知道每个样例$x^{(i)}$对应隐含变量也就是最佳类别$c^{(i)}$。最开始可以随便指定一个$c^{(i)}$给它，然后为了让 P(x,y)最大（这里是要让 J 最小），我们求出在给定 c 情况下，J 最小时的μ_j（前面提到的其他未知参数），然而此时发现，可以有更好的$c^{(i)}$（质心与样例$x^{(i)}$距离最小的类别）指定给样例$x^{(i)}$，那么$c^{(i)}$得到重新调整，上述过程就开始重复了，直到没有更好的$c^{(i)}$指定。这样从 K-means 里我们可以看出它其实就是 EM 的体现，E 步是确定隐含类别变量c，M 步更新其他参数μ来使 J 最小化。这里的隐含类别变量指定方法比较特殊，属于硬指定，从 k 个类别中硬选出一个给样例，而不是对每个类别赋予不同的概率。总体思想还是一个迭代优化过程，有目标函数，也有参数变量，只是多了个隐含变量，确定其他参数估计隐含变量，再确定隐含变量估计其他参数，直至目标函数最优。