# 机器学习与深度学习面试系列十一（聚类和EM）

## 什么是聚类？常见的聚类算法包括哪些?

聚类是一种机器学习技术，它涉及到数据点的分组。给定一组数据点，我们可以使用聚类算法将每个数据点划分为一个特定的组。理论上，同一组中的数据点应该具有相似的属性和/或特征，而不同组中的数据点应该具有高度不同的属性和/或特征。**聚类是一种无监督学习的方法**，是许多领域中常用的统计数据分析技术。

1. **层次聚类。**进一步地看，又有自下而上和自上而下，其中前者最开始时每个样本自成一类，之后将最相似的两类合并称为一个新的类，重复直到满足停止条件，这里的停止条件可能是类的个数，也可能是相似性阈值等等，自上而下则相反，最开始时将所有样本都分为一类，迭代地将类拆分，直到满足类似的停止条件。层次聚类中合并类或拆分类一般是根据类间距离，类似Fisher LDA中所说的“类间间距最大”，衡量不同类之间的距离在不同的距离测度之上还有很多种应用方法，比如类间最短距离、最长距离、类中心距离、类平均距离。
2. **基于划分的聚类**。简单说就是对于一堆待聚类的数据点，先确定最后期望聚成几类，然后挑选几个点作为初始中心点，根据预定的启发式的方法做迭代，直到达到我们的停止条件。例如：*K-means算法*。
3. **基于密度的聚类。**这个类型则是为了处理以密度为特征的类而设计的算法，例如：DBSCAN。
4. **基于网格的聚类。**这类算法将整个数据空间划分为网格单元，将数据对象集映射到网格单元中，然后计算每个单元的密度，将满足预设阈值的网格合并组成类。可想而知，这种方法虽然简单处理速度快，但对数据维数极为敏感，而且对网格大小阈值等参数也很敏感。
5. **基于模型的聚类。**进一步地看，主要有基于概率模型的和基于神经网络的；前者主要是认为每一类数据属于一个概率分布，样本集合是由混合概率分布生成的，其中每一个数据点不再是一定属于某一类，而是以概率的形式来看，典型的是*高斯混合模型（Gaussian Mixed Mode，GMM）*；基于神经网络例如自组织映射神经网络（Self-Organizing Map，SOM）。

下面内容主要包括K-means算法和GMM。

## K-means(K均值)算法是怎样的?

K-means是最普及的聚类算法，算法接受一个未标记的数据集，然后将数据聚类成不同的组。K-means是一个迭代算法，假设我们想要将数据聚类成 n 个组，其方法为:

- 首先选择 个随机的点，称为聚类中心（cluster centroids）；
- 对于数据集中的每一个数据，按照到 个中心点的距离，将其与距离最近的中心点关联起来，与同一个中心点关联的所有点聚成一类。
- 计算每一个组的平均值，将该组所关联的中心点移动到平均值的位置。
- 重复步骤，直至中心点不再变化。

```
Repeat {
    for i = 1 to m:
       c(i) := index (form 1 to K) of cluster centroid closest to x(i)
    for k = 1 to K:
       μk := average (mean) of points assigned to cluster k
}
```

K-means算法主要就是两个for循环。第一个 for 循环是赋值步骤，即：对于每一个样例 $x_i$ ，计算其应该属于的类 $c^{(i)}$ 。第二个 for 循环是聚类中心的移动，即：对于每一个类k，重新计算该类的质心 $u_k$ 。

## K-means算法的损失函数是什么?

K-means是要最小化所有的数据点与其所关联的聚类中心点之间的距离之和，因此K-means的代价函数（又称畸变函数 Distortion function）为：

$$J(c,u)=\frac{1}{m}\sum_{i=1}^{m}||x_i-u_{c(i)}||^2$$

上述伪代码第一个for循环，可以理解为： $c^{(i)} \leftarrow \arg\min_k ||x_i-u_k^{(i)}||^2$

第二个for循环，可以理解为： $u_k \leftarrow \arg\min_u \sum_{i:c^{(i)}=k} ||x_i-u||^2$

## K值如何选择?

K值的选择一般基于经验和多次实验结果。例如采用手肘法，我们可以尝试不同的K值，并将不同K值所对应的损失函数画成折线，横轴为K的取值，纵轴为误差平方和定义的损失函数。

由上图可见，K值越大，距离和越小。并且，当K=3时，存在一个拐点，就像人的肘部一样。当 $K \in (1,3)$ 时，曲线急速下降;当K>3 时，曲线趋于平稳。手肘法认为拐点就是K的最佳值。

## K-means算法需要数据归一化吗?

K均值聚类本质上是一种基于欧式距离度量的数据划分方法，均值和方差大的维度将对数据的聚类结果产生决定性的影响，所以未做归一化处理和统一单位的 数据是无法直接参与运算和比较的。

同时，离群点或者少量的噪声数据就会对均 值产生较大的影响，导致中心偏移，因此使用K均值聚类算法之前通常需要对数据做预处理。

## K-means算法的主要缺点?

1. 需要人工预先确定初始K值，且该值和真实的数据分布未必吻合。
2. K均值只能收敛到局部最优，效果受到初始值很大。
3. 易受到噪点的影响。
4. 样本点只能被划分到单一的类中。

## K-means算法有哪些改进算法?

**K-means++算法**。K-means++主要是对初始值选择的改进。原始K均值算法最开始随机选取数据集中 K个点作为聚类中心，而K-means++假设已经选取了n个初始聚类中心(0<n<K)，则在选取第n+1个聚类中心时，距离当前n个 聚类中心越远的点会有更高的概率被选为第n+1个聚类中心。在选取第一个聚类中心(n=1)时同样通过随机的方法。可以说这也符合我们的直觉，聚类中心当然是互相离得越远越好。当选择完初始点后，K-means++后续的执行和经典K均值算法相同，这也是对初始值选择进行改进的方法等共同点。

**ISODATA(迭代自组织数据分析法)算法**。在K均值算法中，聚类个数K的值需要预先人为地确定，并且在整个算法过程中无法更改。而当遇到高维度、海量的数据集时，人们往往很难准 确地估计出K的大小。ISODATA算法就是针对这个问题进行了改进，它的思想也很直观。当属于某个类别的样本数过少时，把该类别去除。当属于某个类别的样本数过多、分散程度较大时，把该类别分为两个子类别。ISODATA算法在K均值算法的基础之上增加了两个操作，一是分裂操作，对应着增加聚类中心数，二是合并操作，对应着减少聚类中心数。ISODATA算法是一个比较常见的算法，其缺点是需要指定的参数比较多，不仅仅需要一个参考的聚类数量 $K_0$ ，还需要制定3个阈值。下面介绍ISODATA算法的各个输入参数。

1. 预期的聚类中心数目 $K_0$ 。在ISODATA运行过程中聚类中心数可以变化， $K_0$ 是一个用户指定的参考值，该算法的聚类中心数目变动范围也由其决定。 具体地，最终输出的聚类中心数目常见范围是从 $K_0$ 的一半，到两倍 $K_0$ 。
2. 每个类所要求的最少样本数目 $N_{min}$ 。如果分裂后会导致某个子类别所包含样本数目小于该阈值，就不会对该类别进行分裂操作。
3. 最大方差Sigma。用于控制某个类别中样本的分散程度。当样本的分散程度超过这个阈值时，且分裂后满足K的数量在 $\frac{1}{2}K_0$ 到 $2K_0$ 之间，进行分裂操作。
4. 两个聚类中心之间所允许最小距离 $D_{min}$ 。如果两个类靠得非常近(即这两个类别对应聚类中心之间的距离非常小)，小于该阈值时，则对这两个类进行合并操作。

## 高斯混合模型(GMM)是怎样的?

高斯混合模型假设每个簇的数据都是符合高斯分布的，当前数据呈现的分布就是各个簇的高斯分布叠加在一起的结果。

高斯混合模型的核心思想是，假设数据可以看作从多个高斯分布中生成出来 的。在该假设下，每个单独的分模型都是标准高斯模型，其均值 $u_i$ 和方差 $\Sigma_i$ 是待估计的参数。此外，每个分模型都还有一个参数 $\pi_i$ ，可以理解为权重或生成数据的概率。高斯混合模型的公式为:

$$P(x) = \sum_{i=1}^{K} \pi_i N(x|u_i, \Sigma_i)$$

GMM的训练过程和K-means类似。首先初始化所有的参数（类似于K-means初始化所有的聚类中心）。所以每次循环时，先固定当前的高斯分布不变，获得每个数据点由各个高斯分布生成的概率（类似于K-means计算每个数据点关联到最近的聚类中心，只是这里是软分类，以不同的概率分给不同的高斯分布）。然后固定该生成概率不变，根据数据点和生成概率，获得一个组更佳的高斯分布（类似于K-means调整K个聚类中心，这里是调整高斯分布的参数）。循环往复，直到参数的不再变化，或者变化非常小时，便得到了比较合理的一组高斯分布。

## 为什么说K-means是GMM的一个特例?

由上一题我们知道，GMM是一种软分类，以不同的概率将点分给不同的分模型。假设的GMM中所有分模型都满足共同的协方差 $\Sigma_k = \sigma^2 I$ ，在GMM算法的第一个循环里，固定当前的高斯分布不变，获得每个数据点由各个高斯分布生成的概率。

$$\gamma_{nk} = \frac{\pi_k N(x_n|u_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|u_j, \Sigma_j)} = \frac{\pi_k exp\{-\frac{1}{2\sigma^2}||x_n - u_k||^2\}}{\sum_{j=1}^{K} \pi_j exp\{-\frac{1}{2\sigma^2}||x_n - u_j||^2\}}$$

当 $\sigma \to 0$ 时，分母被含有最小的 $||x_n - u_j||^2$ 主导，

$\gamma_{nk} \approx \frac{\pi_j exp\{-\frac{1}{2\sigma^2}||x_n - u_j||^2\}}{\pi_j exp\{-\frac{1}{2\sigma^2}||x_n - u_j||^2\}} = 1$ ，而对于其他 $l \neq j$ ，都有 $\gamma_{nl} = 0$ ，这与K-means的硬分类是相同的。

## GMM与K-Means比较?

高斯混合模型与K均值算法的相同点是:

- 它们都是可用于聚类的算法;
- 都需要 指定K值;

- 都是使用EM算法来求解；
- 都往往只能收敛于局部最优。

而GMM相比于K-Means算法的优点是，可以给出一个样本属于某类的概率是多少；不仅仅可以用于聚类，还可以用于概率密度的估计；并且可以用于生成新的样本点。

## EM算法（Expectation Maximization）是什么?

由上述GMM聚类方法可以看出，其本质就是假设若干簇样本点服从混合高斯分布（因为单个高斯分布具有单峰性，难以拟合多簇样本，理论上混合高斯分布可以拟合任意样本分布），然后对混合高斯分布的参数进行估计。实际上在机器学习里，这是非常常用的手段，例如前面文章中我们说的线性回归，就是假设所有的样本点都满足一个 $N(x_i|w^Tx_i+b,\Sigma)$ 的高斯分布，然后利用最大似然法对这个高斯分布中的参数进行估计。在GMM中，我们假设样本满足 $P(x|\theta)=\sum_{i=1}^{K}\pi_iN(x|u_i,\Sigma_i)$ 这个混合高斯分布( $\theta=\{u,\Sigma\}$ )，我们尝试使用最大似然法来对其进行参数估计：

$$\begin{aligned}\hat{\theta}=\arg\max_{\theta}logP(X|\theta)&=\arg\max_{\theta}log\prod_{n=1}^{N}P(x_n|\theta)\\&=\arg\max_{\theta}\sum_{n=1}^{N}logP(x_n|\theta)\\&=\arg\max_{\theta=\{u,\Sigma\}}\sum_{n=1}^{N}log\sum_{i=1}^{K}\pi_iN(x_n|u_i,\Sigma_i)\end{aligned}$$

在log的内部还有一个连加符号，而对于log中存在求和符号无法继续往下求解，所以高斯混合模型无法使用最大似然估计求出解析解，但对于单个高斯分布是可以用最大似然估计进行求解的。

GMM是一种含隐变量的模型，参数 $\pi_i$ 可以看作是隐变量，$P(X|\theta)=\sum_{Z}P(X,Z|\theta)$ 。高斯分布正是满足这样形式的一个隐变量模型，其中P(X, Z)就是各个分模型（单个的高斯分布）。对于隐变量模型的求解，直接应用极大似然法是搞不定的，所以我们采用EM算法来做。

首先要明确的一点是，EM算法和极大似然法相似的一点是，还是求 $\hat{\theta}=\arg\max_{\theta}logP(X|\theta)$ 。

$$P(X|\theta)=\frac{P(X,Z|\theta)}{P(Z|X,\theta)} \Rightarrow logP(X|\theta)=log\frac{P(X,Z|\theta)}{P(Z|X,\theta)}=logP(X,Z|\theta)-logP(Z|X,\theta)$$

$$\Rightarrow logP(X|\theta)=log\frac{P(X,Z|\theta)}{q(Z|X,\theta)}-log\frac{P(Z|X,\theta)}{q(Z|X,\theta)}$$

$$\Rightarrow \int q(Z|X,\theta)logP(X|\theta)dZ=\int q(Z|X,\theta)log\frac{P(X,Z|\theta)}{q(Z|X,\theta)}dZ-\int q(Z|X,\theta)log\frac{P(Z|X,\theta)}{q(Z|X,\theta)}dZ$$

$$\Rightarrow logP(X|\theta)=\int q(Z|X,\theta)log\frac{P(X,Z|\theta)}{q(Z|X,\theta)}dZ+\int q(Z|X,\theta)log\frac{q(Z|X,\theta)}{P(Z|X,\theta)}dZ$$

$q(Z|X,\theta)$ 是我们取的一个提议分布，仔细观察最后一个等式右边最后一项，其实就是 $KL(q(Z|X,\theta)||P(Z|X,\theta))$ 。我们记最后一个等式右边第一项为 $ELBO$ ，则：$logP(X|\theta)=ELBO+KL(q(Z|X,\theta)||P(Z|X,\theta))$ 。注意， $q(Z|X,\theta)$ 是我们任意取的，也就是q(Z|X)取任意分布，这个等式都恒成立。EM的基本思路就是，在一次迭代中，固定KL项，然后去最大化ELBO，这样 $logP(X|\theta)$ 也就增大了，循环进行，直到收敛，逼近 $logP(X|\theta)$ 的最大值（固定一部分参数，优化另外的参数，类似于坐标上升法）。

根据KL散度的定义易知， $KL(q(Z|X,\theta)||P(Z|X,\theta))\geq 0$ ，不妨使 $KL=0$ ，这样一次迭代可以最大化的优化 $ELBO$ ，进而增大 $logP(x,\theta)$ 。综上，EM算法一次迭代可以总结为两步：

> 第一步：取 $q(Z|X,\theta^{(t)})=P(Z|X,\theta^{(t)})$ ，这样 $KL=0$，其实就是固定了 $\theta$ 为 $\theta^{(t)}$ 。
> 第二步，求 $ELBO=\int q(Z|X,\theta^{(t)})log\frac{P(X,Z|\theta)}{q(Z|X,\theta^{(t)})}dZ=\int q(Z|X,\theta^{(t)})logP(X,Z|\theta)dZ-\int q(Z|X,\theta^{(t)})logq(Z|X,\theta^{(t)})dZ$ 的最大值。最后一个等式的最后一项积分可以看作一个定值（因为 $\theta^{(t)}$ 看作常数），实际上就是求第一项 $\int q(Z|X,\theta^{(t)})logP(X,Z|\theta)dZ$ 的最大值。

EM算法反复重复上述迭代，直至收敛。

## 用EM算法再看GMM?

理解了EM算法后，我们再看GMM，其实GMM中一次循环中的两步就是EM算法的两步。

第一步固定 $\theta^{(t)}$ ，就是保持所有单个高斯分布分布不变，计算 $q(Z|X,\theta^{(t)})=P(Z|X,\theta^{(t)})$，就是在当前各个分模型下，一个样本属于某个分模型的概率 $\gamma_{nk}$ 。

第二步取定 $q(Z|X,\theta^{(t)})$ 就是保持该概率不变，根据样本点调整单个高斯分布的参数，这其实就是最大似然法。

对于GMM来说，一般隐变量Z是离散的，所以上式中求积分符号应该全部为求和符号，考虑到数学意义类似，本文不再做区分。