EM（Expectation Maximization）算法及其应用

一、 背景

在科学研究或者生产过程中的各个领域，往往都有大量的数据处理工作。在数据爆炸的今天，每一秒钟都有成千上万各式各样的数据产生。尽管这些数据往往是各种不同类型的数据，但是通过分析这些数据，可以为我们提供很多有用的信息，帮助我们认识事物的内在规律，研究事物之间的关系以及预测事物的可能发展，这是指导生产实践和科学实验的重要基础。但是，有用的信息并不是一蹴而就的，越是重要的信息往往隐藏地就越深。为了在这大量的数据集中找到这些关键信息，需要花费大量的计算能力以及判断力，这对于我们来说并不是一件容易的事。因此我们必须对数据进行有效的分析和分类，以粗度和精确度为准，掌握主要矛盾，防止因表面的假象进行了错误的判断，最终提取出重要的信息。

数理统计为我们提供了许多有用的统计方法来分析和处理数据。数理统计方法是基于概率的，我们希望通过一定量的数据样本来理解和推断总体特征。就算是一定量的数据，对于人工处理来说，也是不可小觑的难度，因此我们更希望由计算速度更快的计算机来分析和处理。因此我们更加关心如何在计算机上运用好数理统计的方法如回归分析、时间序列分析等，来实现快速、有效地解决实际的问题。基于这个目标，产生了各种统计计算方法，例如随机模拟方法，多种线性回归计算方法等。

参数估计的目的是基于采样来估计和评估数据。点估计，区间估计和假设检验是参数估计中的一些常用方法。由于最大似然估计的渐近最优性质，它已成为参数估计的常用方法，并已广泛用于系统识别，语音处理，图像处理和模式识别等许多领域。但是，目前尚无求解似然函数方程的一般理论方法，在实际应用中主要采用数字优化方法。梯度函数可以使似然函数最大化，这通常要求似然函数具有良好的分析特性，但是在大多数情况下很难满足此要求，因此有必要寻找其他解。在这种情况下，人们提出了 EM 算法。EM 算法主要用于不完整数据的参数估计。通过假设存在隐变量来简化似然函数方程，从而解决方程求解问题。对于某些特殊的参数估计问题，可以轻松实现 EM 算法。

二、 模型介绍

EM 算法是一种迭代优化策略，由 Dempster 等人于 1977 年提出。由于它的计算方法中每一次迭代都分两步，其中一个为期望步（E 步），另一个为极大步（M 步），所以算法被称为 EM 算法。EM 算法受到缺失思想影响，最初是为了解决数据缺失情况下的参数估计问题，其基本思想是首先根据已经给出的观测数据的估计出模型参数的值;然后再依据上一步估计出的参数值估计缺失数据的值，再根据估计出的缺失数据加上之前已经观测到的数据重新再对参数值进行估计，然后反复迭代，直至最后收敛，迭代结束。

EM 算法具有简单且稳定的优点，并且在迭代中始终能保持收敛性。但是 EM 算法的缺点也是十分明显的：EM 算法十分依赖于初始值的设置，如果初始值设置不当，会大大增加计算量；对于大规模数据以及多维度的高斯模型，其迭代的过程量也会大大增加；EM 的极大步（M 步）采用的是求导的方式，因此选取的极值点为局部最优解，并不一定是全局最优解。虽然 EM 算法缺点明显，但是依然瑕不掩瑜，具有很强的实用性。

EM 算法的基本思想如下：记我们观察到的数据为 $X=\{x_1, x_2, x_3, \cdots, x_n\}$，但是有一些数据缺失了（或者称为隐变量），我们将这些数据记为 $Y=\{y_1, y_2, y_3, \cdots, y_m\}$，观测到的数据 X 和缺失的数据 Y 合起来即为全部的数据，记为 Z=(X,Y)。我们的目标是极大化观测数据 x 对于参数θ的对数似然函数$L(\theta|X)$。设$\theta^{(k)}$表示在第 k 次迭代时估计得到的最大值点，k=0，1，…。

定义$Q(\theta|\theta^{(k)})$为观测数据 X={$x_1, x_2, x_3, \cdots, x_n$}条件下完全数据的联合对数似然函数的期望，即：

$$Q(\theta|\theta^{(k)}) = E\{\log L(\theta|Z)|x,\theta^{(k)}\} = E\{\log p(z|\theta)|x,\theta^{(k)}\} = \int [\log(p(z|\theta))]p(y|x,\theta^{(k)})\,dy$$

由上式公式可知对于给定的样本集 X={$x_1, x_2, x_3, \cdots, x_n$}，Y 就是 Z 的唯一随机部分。通过对 Y 求条件期望，就能将 Y 去除，使得$Q(\theta|\theta^{(k)})$完全为θ的函数，这样就可以求解使得$Q(\theta|\theta^{(k)})$最大的θ，记为$\theta^{(k+1)}$，以供下一次迭代使用。EM 算法从初始化一个$\theta^{(0)}$开始，然后在 E 和 M 两步之间交替迭代：

E 步：在给定的观测数据集 X 和已知的θ条件下，求缺失数据 Y 的条件期望，即计算上式的对数似然函数的条件期望$Q(\theta|\theta^{(k)})$。

M 步：对求解出来的$Q(\theta|\theta^{(k)})$进行极大化求解，得到$\theta^{(k+1)}$

$$Q(\theta^{(k+1)}|\theta^{(k)},X) = \max_\theta Q(\theta|\theta^{(k)},X)$$

在应用 EM 的典型模型中，有以下特点：

1、 观测数据点 X 可以是离散的或连续的。这些数据可能存在与每个数据点相关联的观测向量。

2、 缺失值 Y（隐变量）是离散的，从固定数量的值中取，每个观测数据点有一个潜在变量。

3、 参数 θ 是连续的，有两种：一种是与所有数据点相关联的参数，另一种是与隐变量的特定值相关联的参数（即与对应潜在变量具有特定值的所有数据点相关联的参数）。

EM 模型也可以应用于其他类型的模型。比如说，如果我们知道参数θ的值，我们通常可以通过在所有可能的 Y 值上取最大化对数似然来找到隐变量 Y 的值，通过简单地迭代 Y 或通过诸如隐马尔可夫模型这样的算法取得。相反，如果我们知道隐变量 Y 的值，我们可以相当容易地找到参数θ的估计值，通常只需根据相关潜在变量的值对观测数据点进行分组，并平均每组中点的值或值的某些函数。这表明在θ和 Y 都未知的情况下，也能有一个可迭代的算法：

1、 首先，将参数θ初始化为一些随机值

2、 对于给定的这些参数值，计算 Y 的最佳值

3、 然后，使用刚计算出的 Y 值来计算参数θ的更好的估计值。与特定值 Y 关联的参数将仅使用其关联的具有隐变量 Y 的那些数据点

4、 重复步骤 2 和 3，直到收敛

刚才描述的算法单调地逼近损失函数（代价函数）的局部最小值，这种算法通常称为硬 EM 算法。k-means 算法就是这类算法的一个例子。

除此之外，EM 还可以在其他的一些情况下取得较好的效果，比如我们可以不必在给定当前参数值的情况下对 Y 进行艰难选择，只对与特定值 Y 相关联的一组数据点进行平均，从而确定每个数据点的每个可能值 Y 的概率，然后使用与特定值 Y 相关联的概率来计算整个数据点集的加权平均值。这种求解的算法通常称为软 EM，是与 EM 相关联的算法类型。用于计算这些加权平均值的计数称为软计数（与硬 EM 类型算法如 k-means 中使用的硬计数相反）。为 Y 计算的概率是后验概率，是在 E 步中计算的。用于计算新参数值的软计数是在 M 步中计算的。

三、 EM 算法分析

(一) 算法推导

1. Jensen 不等式

有 $f(x)$为定义在区间 I=[a,b]上的实值函数，若对于任意 $x_1, x_2 \in I$，$\lambda \in [0,1]$，有：

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

则$f(x)$为凸函数。凸函数的判定条件是$f''_{(x)} \geq 0$。

Jensen 不等式就是上述公式的推广：设$f(x)$为凸函数，$\lambda_i \geq 0$，$\Sigma_i \lambda_i = 1$，则：

$$f\left(\sum_{i=1}^{n} \lambda_i \lambda_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i)$$

若将λ_i视为一个概率分布，则可表示为期望值的形式：

$$f(E[x]) \leq E[f(x)]$$

2. 极大似然估计和极大后验估计

极大似然估计(Maximum likelihood estimation)是参数估计的常用方法，基本思想是在给定样本集的情况下，求使得该样本集出现的“可能性”最大的参数θ。将参数θ视为未知量，则参数θ对于样本集 X 的对数似然函数为：

$$L(\theta) = ln\, P(X|\theta)$$

这个函数反映了在观测结果 X 已知的条件下，θ的各种值的“似然程度”。这里是把观测值 X 看成结果，把参数θ看成是导致这个结果的原因。参数θ虽然未知但是有着固定值，并非事件或随机变量，无概率可言，因而改用“似然(likelihood)”这个词。

于是通过求导求解使得对数似然函数最大的参数θ即为极大似然法。

$$\theta = \arg max_\theta\, L(\theta)$$

极大后验估计(Maximum a posteriori estimation)是贝叶斯学派的参数估计方法，相比于频率学派，贝叶斯学派将参数θ视为随机变量，并将其先验分布$P(\theta)$包含在估计过程中。运用贝叶斯定理，参数θ的后验分布为：

$$P(\theta|X) = \frac{P(X,\theta)}{P(X)} = \frac{P(\theta)P(X|\theta)}{P(X)} \propto P(\theta)P(X|\theta)$$

上式中$P(X)$不依赖于θ因而可以做为常数项舍去，则最终结果为：

$$\theta = \arg max_\theta\, P(\theta)P(X|\theta)$$

3. EM 算法推导

我们的目标是对可观测数据集 $X=\{x_1, x_2, x_3, \cdots, x_n\}$求观测概率最大时的模型参数$\theta$，因此我们采用极大似然估计法，极大化模型分布的对数似然函数：

$$\theta = \arg max_\theta \sum_{i=1}^{n} \log p(x_i|\theta)$$

由于存在隐变量 y，因此也可以表示为极大化 X 的边缘分布，即：

$$l(\theta) = \sum_{i=1}^{n} \log p(x_i|\theta) = \sum_{i=1}^{n} \log \sum_{y=1}^{K} p(x_i, y|\theta)$$

由于上式没有一个很好的闭合解，无法进行优化化简，因此无法求出目标θ，因此必须通过其他方式进行化简。已知任意的 i=1,2,⋯,n 与 y=1,2,⋯,k 有：

$$\sum_{i=1}^{n} p(y|x_i, \theta^{(t)}) = 1 \quad , \quad p\left(y|x_i, \theta^{(t)}\right) \geq 0$$

对$l(\theta)$对引入 Jensen 不等式进行转化，可得：

$$l(\theta)=\sum_{i=1}^{n}\log\sum_{y=1}^{K}p(x_i,y|\theta)=\sum_{i=1}^{n}\log\sum_{y=1}^{K}\frac{p(y|x_i,\theta^{(t)})p(x_i,y|\theta)}{p(y|x_i,\theta^{(t)})}$$

$$\geq\sum_{i=1}^{n}\sum_{y=1}^{K}p(y|x_i,\theta^{(t)})\log\frac{p(x_i,y|\theta)}{p(y|x_i,\theta^{(t)})}=Q(\theta,\theta^{(t)})$$

对于$Q(\theta,\theta^{(t)})$，$\theta^{(t)}$是上一次迭代的参数，是一个定值；$p(y|x_i,\theta^{(t)})$在观测样本给定时是一个常数，不会影响求解最大化 Q 时的参数θ，可以忽略，因此上式可以简化为：

$$Q(\theta,\theta^{(t)})=\sum_{i=1}^{n}\sum_{y=1}^{K}p(y|x_i,\theta^{(t)})\log p(x_i,y|\theta)$$

对于$\sum_{y=1}^{K}p(y|x_i,\theta^{(t)})\log p(x_i,y|\theta)$可以理解为$\log p(x_i,y|\theta)$基于条件概率分布$p(y|x_i,\theta^{(t)})$的期望，因此当$\theta=\theta^{(t)}$时，有：

$$Q(\theta^{(t)},\theta^{(t)})=\sum_{i=1}^{n}\sum_{y=1}^{K}p(y|x_i,\theta^{(t)})\log\frac{p(x_i,y|\theta^{(t)})}{p(y|x_i,\theta^{(t)})}=\sum_{i=1}^{n}\sum_{y=1}^{K}p(y|x_i,\theta^{(t)})\log p(x_i,\theta^{(t)})$$

$$=\sum_{i=1}^{n}\log p(x_i,\theta^{(t)})=l(\theta^{(t)})$$

由此可得，$Q(\theta,\theta^{(t)})$是$l(\theta)$的下界，只要使$Q(\theta,\theta^{(t)})$的最大值增大，就可以使$l(\theta^{(t)})$不断增大。因此我们的目标就变成了最大化$Q(\theta,\theta^{(t)})$：找到一个$\theta=\theta^{(t+1)}$使得

$$l(\theta^{(t+1)})\geq Q(\theta^{(t+1)},\theta^{(t)})\geq Q(\theta^{(t)},\theta^{(t)})=l(\theta^{(t)})$$

上式不断地迭代寻找更好的θ最终使得$Q(\theta,\theta^{(t)})$达到最大值，此时$l(\theta)$也达到了局部最大值。

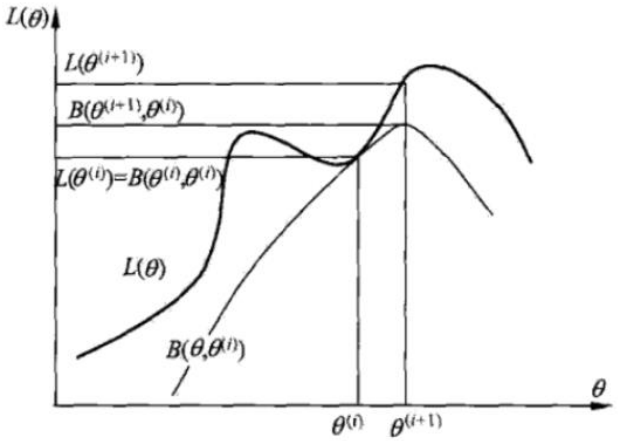


图 1. EM 算法迭代更新过程

（二） EM 算法收敛性分析

上述的算法在未证明的情况下，认为$l(\theta)$在$\theta>\theta^{(0)}$上是单调有界递增的，因此才能通过求解最大似然值来逼近，EM 算法才算有解。下面我们就来讨论$l(\theta)$是否真实满足条件。

首先，由于可观测数据 X 的样本数量为有限的，因此$l(\theta)$一定是有上界的，因此我们只需要证明$l(\theta)$是否是单调递增的即可。即：

$$l(\theta^{(i+1)})\geq l(\theta^{(i)})\quad=>\quad\sum_{i=1}^{n}\log p(y|x_i,\theta^{(i+1)})\geq\sum_{i=1}^{n}\log p(y|x_i,\theta^{(i)})$$

写作向量形式有：

$$P(\mathrm{X}|\theta^{(i+1)}) \geq P(\mathrm{X}|\theta^{(i)}), i = 1,2,3\ldots$$

由贝叶斯公式可得：

$$P(\mathrm{X}|\theta) = \frac{P(\mathrm{X},\mathrm{Z}|\theta)}{P(Z|X,\theta)}$$

两边同时取对数可得：

$$\log P(X|\theta) = \log P(X,Z|\theta) - \log P(Z|X,\theta)$$

由于前面计算有：

$$Q(\theta,\theta^{(i)}) = \sum_Z \log P(X,Z|\theta)P(Z|X,\theta^{(i)})$$

令：

$$H(\theta,\theta^{(i)}) = \sum_Z \log P(Z|X,\theta)P(Z|X,\theta^{(i)})$$

则对数似然函数可以写成：

$$\log P(X|\theta) = Q(\theta,\theta^{(i)}) - H(\theta,\theta^{(i)})$$

取$\theta^{(i+1)}$和$\theta^{(i)}$相减可得：

$$\log P(X|\theta^{(i+1)}) - \log P(X|\theta^{(i)}) = [Q(\theta,\theta^{(i+1)}) - H(\theta,\theta^{(i+1)})] - [Q(\theta,\theta^{(i)}) - H(\theta,\theta^{(i)})]$$
$$= [Q(\theta,\theta^{(i+1)}) - Q(\theta,\theta^{(i)})] - [H(\theta,\theta^{(i+1)}) - H(\theta,\theta^{(i)})]$$

由于$Q(\theta,\theta^{(i+1)}) > Q(\theta,\theta^{(i)})$，因此当$H(\theta,\theta^{(i+1)}) \leq H(\theta,\theta^{(i)})$时，必有$P(\mathrm{X}|\theta^{(i+1)}) \geq P(\mathrm{X}|\theta^{(i)})$成立，从而$l(\theta^{(i+1)}) \geq l(\theta^{(i)})$成立。

$$H(\theta,\theta^{(i+1)}) - H(\theta,\theta^{(i)}) = \sum_Z P(Z|X,\theta^{(i)}) \log \frac{P(Z|X,\theta^{(i+1)})}{P(Z|X,\theta^{(i)})}$$
$$\leq \log \sum_Z P(Z|X,\theta^{(i)}) \frac{P(Z|X,\theta^{(i+1)})}{P(Z|X,\theta^{(i)})} = \log \sum_Z P(Z|X,\theta^{(i+1)})$$
$$= 0 \qquad (Jenson不等式)$$

因此$P(\mathrm{X}|\theta^{(i+1)}) \geq P(\mathrm{X}|\theta^{(i)})$成立，故 EM 算法能够收敛。

四、 EM 算法的应用

(一) K-means

1. 原理

K-means 是一种原型聚类算法，其目标是将样本集划分为 K 个簇，使得同一个簇内样本的距离尽可能小，不同簇之间的样本距离尽可能大，即最小化每个簇中样本与质心的距离。原型聚类指聚类结构能通过一组原型刻画，而原型即为样本空间中具有代表性的点。K-means 的原型就是每个簇的质心μ。

如果已知每个数据所属的簇，则可以直接求解质心μ，但实际的应用中，往往大部分情况下，数据集所属的簇是是不能全部得知的或者只有部分可知，因此我们需要通过 EM 算法来进行分类。使用 EM 算法求解 K-means 时，EM 算法的参数即每个簇的质心μ，隐变量即每个数据所属的簇。假定有 k 个簇，则先随机选定 k 个簇的质心$\{\mu_1,\mu_2,\ldots,\mu_k\}$，按照 EM 过程有：

1) 固定μ_k，将样本划分到距离最近的μ_k所属的簇中，用r_{nk}表示第 n 个样本所属的簇，则有：

$$r_{nk} = \begin{cases} 1 & if\ \left(k = \arg min_j \|x_n - \mu_j\|^2\right) \\ 0 & otherwise \end{cases}$$

2） 固定r_{nk}，根据上一步划分的结果重新计算每个簇的质心。由于我们的目标是最小化每个簇中样本与质心的距离，可将目标函数表示为：

$$J = \sum_{n=1}^{N} r_{nk} \left\| x_n - \mu_j \right\|^2$$

要最小化J则对μ_k求导，并令其为 0，解得μ_k即为簇中样本的均值向量，为：

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

上述两步分别更新了r_{nk}和μ_k，也就对应了 EM 算法中的 E 步和 M 步。和 EM 算法一样，K-means 每一步都最小化目标函数J，因而可以保证收敛到局部最优值，但在非凸目标函数的情况下不能保证收敛到全局最优值。另外，K-means 对每个样本进行“硬分类”，即只归类为一个簇，然而某些样本可能处于簇与簇的边界处，将这些样本强行分到其中一个簇可能并不能反映确信度。

2. 优缺点

优点：

1） K-means 算法原理简单明了，参数较少，易于实现；

2） K-means 的时间复杂度为$O(N \cdot K \cdot I)$，N 为样本数量，K 为簇数，I 为迭代次数，由于在实际使用场景中，N 的数量往往远大于 K 和 I，因此时间复杂度可近似为$O(N)$，效率比较高，收敛的速度比较快。

缺点：

1） 对噪声和奇异点敏感，如少数离群样本可能会对整个簇的分类产生影响；

2） 对 K 的初始化不容易把握，交叉验证不太合适，因为簇越多，目标函数J的值越小；

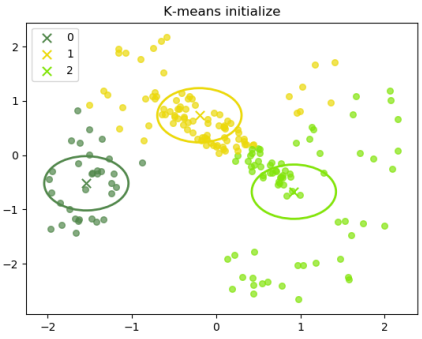
3） 对于非凸数据集或者规模差异太大的数据效果不好；

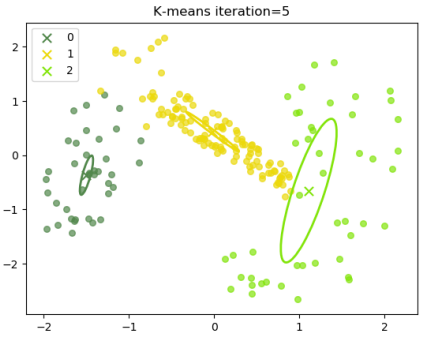
4） 无法保证收敛到全局最优值，常使用不同的初始值进行多次试验；

5） 采用欧氏距离作为质心选取判断，对于数据类型有局限性。

3. 实践测试

数据集及源代码见附件：





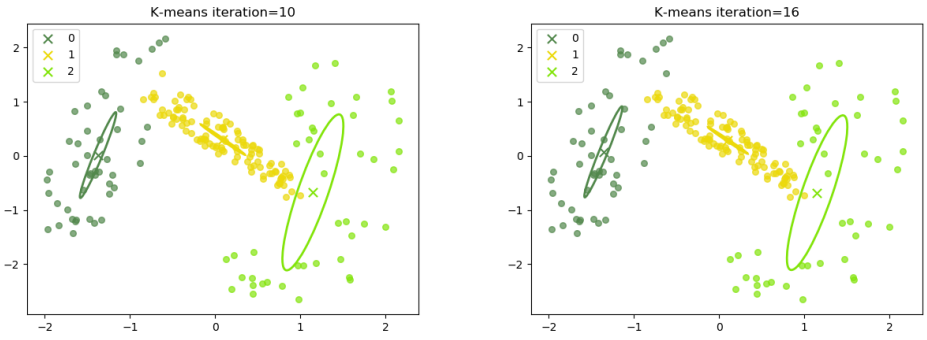


图 2.　K-means 迭代过程

根据随机初始化的质心μ的不同取值，K-means 的算法迭代次数不尽相同。如上图所示的数据集，经过 16 次迭代就能完成数据集的分类，结束了迭代过程。而在测试中，同样的数据集，也有经过迭代 30 多次才能稳定分类的情况。

（二）　高斯混合模型（GMM）

1.　原理

高斯混合模型同样也可以用于聚类，但与 K-mean 不同的是，高斯混合模型给出的是软分类的结果，即给出的是样本属于簇的概率。

对于高斯分布，有：

$$\mathrm{N}(x|\mu,\sigma^2)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

若 x 为 n 维随机向量，则多元高斯分布为：

$$\mathrm{N}(x|\mu,\Sigma)=\frac{1}{\sqrt{(2\pi)^n\cdot|\Sigma|^{\frac{1}{2}}}}e^{-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}}$$

其中μ为 n 维均值向量，Σ为 n×n 的协方差矩阵，$|\Sigma|$为Σ的行列式。

很多情况下，一个高斯分布并不能很好地体现数据的总体性质，例如图 2 中的数据集，用一个高斯分布来评价显然是不合理的，因此我们可以考虑使用多个高斯分布去拟合整个数据集，这便是高斯混合模型。设数据集由 k 个高斯分布线性组合，则数据样本的概率分布可以表示为：

$$p(x)=\sum_{k=1}^{K}\pi_k N(x|\mu_k,\Sigma_k)$$

其中π_k为每个高斯的混合系数，满足$0\le\pi_k\le1$，且$\sum_{k=1}^{K}\pi_k=1$。

我们要从高斯混合模型的分布中随机地取一个点的话，实际上可以分为两步：首先随机地在这 些高斯分模型之中选一个，每个高斯分模型被选中的概率实际上就是它的系数π_k，选中了高斯分模型之后，再单独地考虑从这个高斯分模型的分布中选取一个点就可以了——这里已经回到了普通的高斯分布，转化为了已知的问题。

实际上就是用一个 k 维的二元随机变量$\mathbf{z}$，表示每一个点属于哪一个高斯分模型的。定义一个 k 维随机变量$\mathbf{z}=[z_1,z_2,\dots,z_k]$，$\mathbf{z}$中只有一个$z_n$的值为 1，表示这个样本来自第 n 个高斯分模型，其余的z_i值都为 0。由于$p(x)=\sum_z p(x,z)=\sum_z p(z)p(x|z)$，对比上式，可以将$\pi_k$作为每个高斯分模型的先验概率，即$\pi_k=p(z_k=1)$，对应的有$N(x|\mu_k,\Sigma_k)=p(x|z_k=1)$。在得到观测数据$\{x_1,x_2,\cdots,x_n\}$后，每个 x_n都对应一个隐变量z_{nk}，利用贝叶斯定理，记 x_n的后验概率$\gamma(z_{nk})$，则有：

$$\gamma(z_{nk}) = p(z_{nk}=1|x_n) = \frac{p(z_{nk}=1)p(x_n|z_{nk}=1)}{\sum_{j=1}^{K} p(z_{nj}=1)p(x_n|z_{nj}=1)} = \frac{\pi_k N(x_n|\mu_k,\Sigma_k)}{\sum_{j=1}^{K}\pi_j N(x|\mu_j,\Sigma_j)}$$

因为我们并不知道参数，所以需要进行参数估计。一般采取极大似然函数去估计参数，为了方便计算通常取对数的似然估计函数：

$$L(\theta) = \ln p(X|\theta) = \ln p(X|\pi,\mu,\Sigma) = \ln[\prod_{n=1}^{N}(\sum_{k=1}^{K}\pi_k N(x|\mu_k,\Sigma_k))$$

$$= \sum_{n=1}^{N}\ln(\sum_{k=1}^{K}\pi_k \frac{1}{\sqrt{(2\pi)^n}\cdot|\Sigma|^{\frac{1}{2}}} e^{-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}})$$

由于上式过于复杂，无法直接求解，因此我们使用 EM 算法来简化计算：

$$Q(\theta,\theta^t) = \sum_{n=1}^{N}\sum_{z} p(z|x_n,\pi^{(t)},\mu^{(t)},\Sigma^{(t)})\ln(x_n,z|\pi,\mu,\Sigma)$$

式中第一项为 x_n 的后验概率，第二项为完全数据对数似然函数，因此可以将上式化简为：

$$Q(\theta,\theta^t) = \sum_{n=1}^{N}\sum_{z}\gamma(z_{nk})\ln(\pi_k \frac{1}{\sqrt{(2\pi)^n}\cdot|\Sigma|^{\frac{1}{2}}} e^{-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}})$$

上式分别对μ_k,Σ_k进行求导，得：

$$\mu_k = \frac{\sum_{n=1}^{N}\gamma(z_{nk})\cdot x_n}{\sum_{n=1}^{N}\gamma(z_{nk})}, \Sigma_k = \frac{\sum_{n=1}^{N}\gamma(z_{nk})\cdot(x_n-\mu_k)\cdot(x_n-\mu_k)^T}{\sum_{n=1}^{N}\gamma(z_{nk})}$$

又已知混合系数$\sum_{k=1}^{K}\pi_k = 1$，利用拉格朗日乘子法可将上式转化为：

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\left[\ln(\pi_k) + \ln\left(\frac{1}{\sqrt{(2\pi)^n}\cdot|\Sigma|^{\frac{1}{2}}} e^{-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}}\right)\right] + \lambda((\sum_{k=1}^{K}\pi_k) - 1)$$

对π_k求导，解得：

$$\pi_k = \frac{\sum_{n=1}^{N}\gamma(z_{nk})}{-\lambda}$$

带入$\sum_{k=1}^{K}\pi_k = 1$，最终解得：

$$\pi_k = \frac{\sum_{n=1}^{N}\gamma(z_{nk})}{N}$$

即每个分模型 k 的混合系数是属于 k 的样本的平均后验概率，由此运用 EM 算法能大大简化高斯混合模型的参数估计过程，在中间步只需计算$\gamma(z_{nk})$就行了。因此我们在计算高斯混合模型时就可以按照以下步骤进行快速求解:

对于样本集 X，初始化其分模型的参数为π_k,μ_k,Σ_k，然后按照 EM 迭代：

E 步：

依据当前模型参数，观测 x_n 属于各个分模型的后验概率：

$$\gamma(z_{nk}) = p(z_{nk}=1|x_n) = \frac{p(z_{nk}=1)p(x_n|z_{nk}=1)}{\sum_{j=1}^{K} p(z_{nj}=1)p(x_n|z_{nj}=1)} = \frac{\pi_k N(x_n|\mu_k,\Sigma_k)}{\sum_{j=1}^{K}\pi_j N(x|\mu_j,\Sigma_j)}$$

M 步：

根据 E 步求解出来的$\gamma(z_{nk})$计算新一轮的参数：

$$\mu_k^{new} = \frac{\sum_{n=1}^{N}\gamma(z_{nk})\cdot x_n}{\sum_{n=1}^{N}\gamma(z_{nk})}, \Sigma_k^{new} = \frac{\sum_{n=1}^{N}\gamma(z_{nk})\cdot(x_n-\mu_k)\cdot(x_n-\mu_k)^T}{\sum_{n=1}^{N}\gamma(z_{nk})}, \pi_k^{new} = \frac{\sum_{n=1}^{N}\gamma(z_{nk})}{N}$$

重复上述 E、M 两步，直到收敛。

2. 优缺点

优点：

1) 相比于 K-means，更具有一般性，对于数据集簇的形状要求没有那么高；
2) 在各种混合模型中运行的速度最快；
3) 算法的目标是最大化可能性，不会使得聚类偏向于特殊的结构。

缺点：

1) 当每个混合模型没有足够的点时，估算协方差变得困难起来，同时算法会发散并且找具有无穷大似然函数值的解，除非人为地对协方差进行正则化。
2) 分模型数量难以预先选择
3) 相比于 K-means 计算量较大，收敛较慢

3. 实践

源代码见附件：

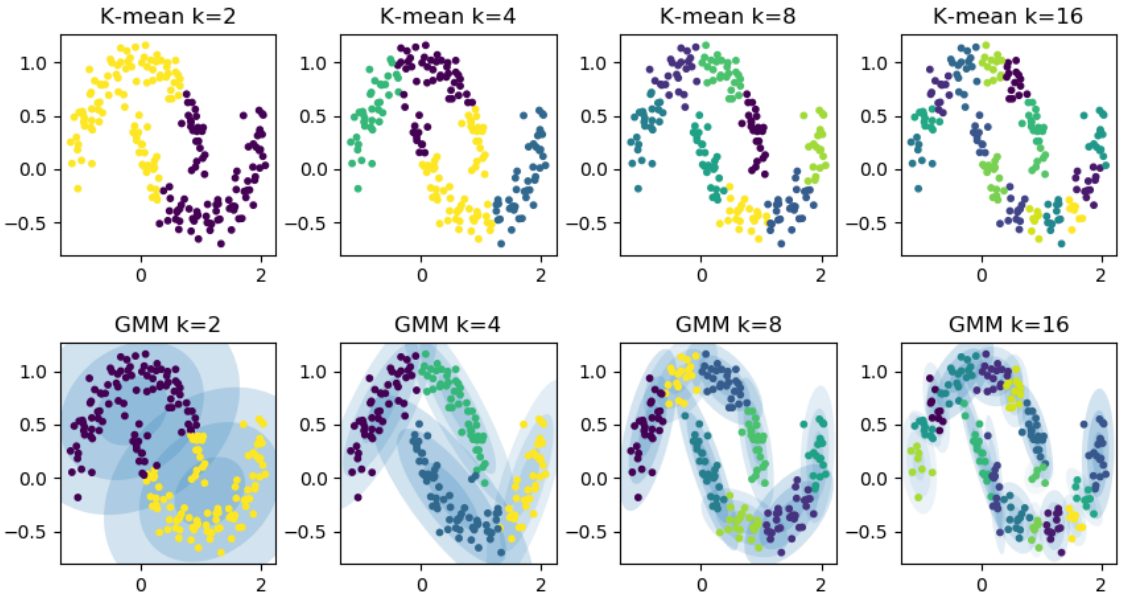


图 3. 不同 k 值对两种算法结果影响

如上图所示，对于一些数据集较为特别的分类来说，GMM 所得到的分类效果会比 K-means 更快得到一个相对合理的结果。同时，由于 GMM 模型生成的结果是一个概率模型，我们可以很轻松地利用这个概率模型生成一个新的随机数据，如下图所示：

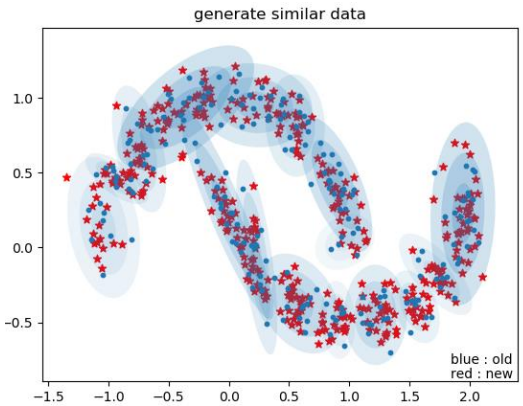


图 4. GMM 模型生成数据

五、 EM 算法拓展或改进

(一) 改进 E 步的算法(MCEM)

E 步的目标是在已知观测数据的条件下，求出缺失数据的条件期望，然后再求出完全数据下的期望对数似然。在有些情况下，无法通过简单的转化获得显式的期望，这时候 EM 算法就无法发挥作用。因此就有利用 Monte Carlo 模拟的方法来近似实现求解 E 步积分，即 MCEM 算法。

在 MCEM 中，第 k+1 步的 E 可以使用以下方式代替求解：

1） 在$p(y|x,\theta^{(k)})$中随机抽取$m(k)$个数，构成独立同分布的缺失数据集$\{y_1, y_2, \cdots, y_{m(k)}\}$，集合中每个 y_j是用来补充观测数据，构成一个完全数据集$z_j = (x, y_j)$；

2） 计算$\hat{Q}^{(k+1)}(\theta|\theta^{(k)}) = \frac{1}{m(k)}\sum_{j=1}^{m(k)} \log p(z_j|\theta)$，得到$\hat{Q}^{(k+1)}(\theta|\theta^{(k)})$就是$Q(\theta|\theta^{(k)})$的 Monte Carlo 估计。只要 m 足够大，我们就可以认为$\hat{Q}^{(k+1)}(\theta|\theta^{(k)})$和$Q(\theta|\theta^{(k)})$基本相等。

完成上述两步后就可以在 M 步中对$\hat{Q}^{(k+1)}(\theta|\theta^{(k)})$进行极大化求解，得到$\theta^{(k+1)}$，带回到 E 步中进行下一次迭代。

MCEM 算法的精度主要依赖于$m(k)$的选择。$m(k)$越大，MCEM 的精度越好，但是也会严重影响计算的速度。并且 MCEM 最终收敛的方式也与 EM 不同，MCEM 最终收敛值会在真实的最大值附近波动，因此需要通过额外的方式判断收敛。

(二) 改进 M 步的算法(ECM)

改进 M 步的一个方向就是减少或者避免出现迭代的 M 步，可以考虑在 M 步中每次都让 Q 函数增大，而不是求 Q 的极大化。常见的方法有 ECM 算法以及 ECM 的改进算法 ECME、ACEM 算法。

ECM 算法为了避免迭代 M 步，设计了一些简单的条件极大化步骤来简化 M 步，一般称之为 CM 循环，主要包括以下两个步骤：

1） 令 S 表示每个 CM 循环里 CM 步的个数，对 s=1,2,…,S，第 k 次迭代过程的第 k 次 CM 循环里，第 s 个 CM 步需要在约束条件$g_s(\theta) = g(\theta^{(k+\frac{s-1}{s})})$下求$Q(\theta|\theta^{(k)})$的最大化，其中$\theta^{(k+\frac{s-1}{s})}$是第 k 次 CM 循环的第 s-1 个 CM 步得到的估计值；

2） 当完成了 S 次 CM 步的循环，令$\theta^{(k+1)} = \theta^{(k+\frac{S}{S})}$，带入到 E 步的迭代中进行下一轮计算。

为了保证 ECM 算法的收敛性，需要确保每次的 CM 循环都是在θ向量空间的任意方向上搜索$Q(\theta|\theta^{(k)})$的最大值点，这样 ECM 算法才能像 EM 算法一样收敛到一个稳定点。因此 ECM 算法对θ的向量空间取值有一定的要求。

(三) 参数拓展 EM 算法(PX-EM)

为了提升 EM 算法的收敛速度，PX-EM 算法通过引进附加参数α，对参数及进行了扩充，使得新的参数空间为$\phi = (\theta^*, \alpha)$，此处$\theta^*$和 EM 算法中的$\theta$是同维的，且当$\alpha = \alpha_0$时，有$\theta^* = \theta$，此时即为原来的 EM 模型。使用 PX-EM 算法需要满足以下几个条件：

1） 存在某个已知变换R使得$\theta = R(\theta^*, \alpha)$

2） 当$\alpha = \alpha_0$时，在观测到的数据 X 上不存在α的信息

3） 在拓展的模型中，参数ϕ对于完全数据集$Z = (X, Y)$是可识别的

PX-EM 算法的迭代过程如下：

1） PX-E 步：

计算$Q(\phi|\phi^{(k)}) = E(\log p(z|\phi)|x,\phi^{(k)})$

2） PX-M 步：

求$\phi^{(k+1)} = \arg max_{\phi}\, Q(\phi|\phi^{(k)})$，然后令$\theta^{(k+1)} = R(\theta^{*(k+1)},\alpha)$

循环以上两步，直到收敛。

六、 总结

本文对 EM 算法的原理、实现步骤、收敛性讨论、实例应用、改进方向进行了研究与讨论。EM 算法作为很多算法中的重要组成以及理论基础，常常用于机器学习模型的参数求解，常见应用于 K-Means、高斯混合模型、隐马尔可夫模型等，具有简单且稳定的特点，算法原理较为明了，收敛稳定，适用性较为宽广，是一种被广泛应用于极大似然估计的迭代计算方法，在处理数据缺失等问题上优势明显。EM 算法的不足在于其可能会陷入局部最优，在高维数据的问题中，局部最优和全局最优可能有很大差异。

综上所述，EM 算法在当下机器学习算法中具有十分重要的地位，通过学习和理解 EM 算法，对于我们的模型的优化十分有帮助。EM 模型也在不断地改进优化，相信在未来能够有更快的收敛速度，帮助机器学习的模型更快地拟合。

参考文献

[1]. 李航.统计学习方法.北京：清华大学出版社，2012

[2]. 张宏东. EM 算法及其应用.山东大学

[3]. 房彦兵.α-EM 算法及其若干应用.上海交通大学

[4]. https://www.edureka.co/blog/em-algorithm-in-machine-learning/

[5]. https://blog.csdn.net/fennvde007/article/details/17734597

[6]. https://blog.csdn.net/qq_24519677/article/details/82383306

[7]. https://www.cnblogs.com/massquantity/p/9248482.html

[8]. https://www.cnblogs.com/long5683/p/11402875.html

[9]. https://brainbomb.org/Artificial-Intelligence/Machine-Learning/ML-Mixture-Models-EM-Algorithm-for-Gaussian-Mixtures/

[10]. https://wei2624.github.io/MachineLearning/usv_em/

[11]. https://psychology.wikia.org/wiki/Expectation%E2%80%93maximization_algorithm

[12]. https://www.cnblogs.com/jiangxinyang/p/9278608.html

[13]. https://www.jianshu.com/p/008025aaad25

[14]. https://blog.csdn.net/jasonzhoujx/article/details/81947663

[15]. 夏筱筠,张笑东,王帅等.一种半监督机器学习的 EM 算法改进方法.《小型微型计算机系统》2020 年 2 月第 2 期