基于期望最大化EM算法估计混合高斯模型GMM参数

SY1906206 王阁元，SY1906407 罗薇，SY1906202 寄家豪

2020 年 6 月 7 日

1 理论基础

EM算法也称作期望最大化（Expectation-Maximization，简称EM）算法，它是一种迭代算法，用于含有隐变量的概率模型**参数**的极大似然估计或极大后验概率估计。

1.1 极大似然估计

极大似然估计（Maximum likelihood estimation，简称MLE）就是利用已知的样本结果（数据）信息，反推最**具有可能（最大概率）**导致这些样本结果（数据）出现的模型参数值。

考虑图1，红色叉号表示数据点，这组数据上方有三个高斯分布。现在假设这组数据全部来自于同一个分布，那最有可能是哪一个分布呢？我们都知道，高斯分布的参数为$\theta = \{\mu, \sigma\}$，那么问题其实可以表述为：图1 中三个分布对应的三组参数里，哪组参数能够更好的解释数据？即哪组参数让这些数据样本出现的可能性最大？

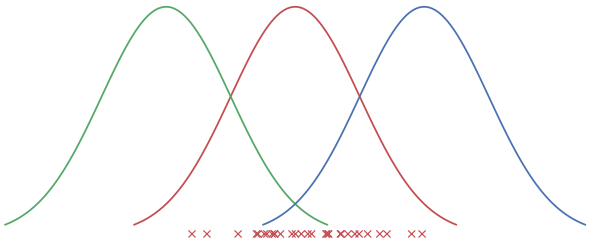

图 1: MLE估计高斯分布参数

假设数据点表述为$\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$，且每个数据点之间都满足独立同分布条件，其概率密度函数为$P(X = x|\theta)$，那么所有数据出现的概率就是

$$P(\mathcal{X}|\theta) = \prod_{i=1}^{n} P(x_i|\theta). \tag{1}$$

可以看到公式中数据$\mathcal{X}$是已知的，所以$P(\mathcal{X}|\theta)$是一个关于θ的函数，通常被称作**似然函数**。

接下来就要寻找能够更好地解释这组数据的参数θ，即使得函数$P(\mathcal{X}|\theta)$的值最大的θ，写作

$$\hat{\theta} = \arg\max_{\theta} \prod_{i=1}^{n} P(x_i|\theta). \tag{2}$$

要最大化一个函数的值，我们首先想到的一定是一阶导数为零。但是观察公式 (1)，对于连乘的函数求导并不是一件容易的事情，于是考虑取对数简化运算。

这里有两点考虑：一方面，对数函数能够保持原函数的增减性，所以取对数前后函数的极值点保持一致；另一方面，对数函数能够变乘法为加法，大大降低了运算难度，也提高了数值的可识别性，比如概率累乘会出现数值非常小的情况（像1e-30这样的数值），很容易超出计算机的精度产生溢出错误，而取对数之后，计算机就很容易识别了（对1e-30取以10为底的对数得到-30）。于是**对数似然函数**便产生了，如公式 (3)，

$$\begin{aligned}\mathcal{L}(\theta|\mathcal{X}) &= \log P(\mathcal{X}|\theta) \\ &= \log \prod_{i=1}^{n} P(x_i|\theta) \\ &= \sum_{i=1}^{n} \log P(x_i|\theta).\end{aligned} \tag{3}$$

注意这里的对数似然函数写成$\mathcal{L}(\theta|\mathcal{X})$而不是$\mathcal{L}(\theta)$，是因为待估计量（参数）$\theta$是随着观测数据$\mathcal{X}$的变化而变化的。因此优化目标变为：

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log P(x_i|\theta). \tag{4}$$

接下来便是对多维参数求偏导（若是一维参数则直接求导），然后令一阶导数为零，最后一一解出参数的估计值即可。

这里我们以高斯分布为例，进行参数的极大似然估计。

首先高斯分布的概率密度函数为

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{5}$$

假设数据为$X = (x_1, x_2, \cdots, x_n)^T$，　$x_i \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$，可以得到对数似然函数

$$\begin{aligned}\mathcal{L}(\mu, \sigma|X) &= \sum_{i=1}^{n} \log P(x_i|\mu, \sigma) \\ &= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \\ &= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}} + \log \frac{1}{\sigma} - \frac{(x_i-\mu)^2}{2\sigma^2}.\end{aligned} \tag{6}$$

然后用$\mathcal{L}(\mu, \sigma|X)$分别对参数$\mu$和$\sigma$求偏导并令其为零，因为比较容易，所以这里省略化简过程，最后可以得到参数的估计值：

$$\begin{aligned}\hat{\mu}_{MLE} &= \frac{1}{n} \sum_{i=1}^{n} x_i \\ \hat{\sigma}^2_{MLE} &= \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{MLE})^2.\end{aligned} \tag{7}$$

不难发现，用极大似然估计的高斯分布$\hat{\mu}$为所有样本数据点的均值，$\hat{\sigma}^2$为所有样本数据点方差。

最后，简单总结一下极大似然估计参数的过程：

i. 根据概率密度函数写出似然函数

ii. 对似然函数取对数，整理表达式

iii. 对表达式求一阶导，令导数为零，得到似然方程

iv. 解似然方程，得到参数的估计值

1.2 隐变量

在EM算法的学习过程中，经常会看到一个词：**隐变量**（latent variable），它是相对于**观测变量**（obeservable variable）而言的，观测变量一般就指的是数据本身，那么隐变量到底是什么呢？其实就是未观测到的但是影响观测数据的变量。

下面我们举例进行解释。刚刚在极大似然估计的过程中存在一个假设：所有的数据点来自于同一个分布。当然这个假设在直观上也是符合认知的，因为图1 中的那组数据看起来确实像是从同一个高斯分布中抽取出来的。

但是并不是所有问题都符合这种假设，更多的是图2 中的数据分布，如果我们依旧用单个高斯分布去拟合，根据章节1.1 中讨论过的极大似然估计的结果，$\hat{\mu}$（μ的估计值）为样本均值，$\hat{\sigma}^2$（σ^2的估计值）为样本方差，因此就会出现图2 中的情况。显然，这不是我们想要的。

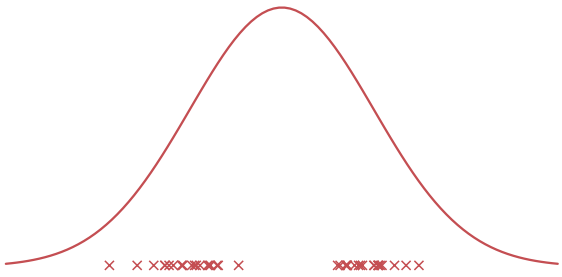

图 2: 单高斯拟合含有隐变量的数据

很容易想到：我们可以用两个高斯去拟合。这其实就是混合高斯模型的雏形，其模型思想很简单，如图3 所示，当给出的样本是绿色的点时，就用绿色的高斯分布去拟合，当给出的样本是红色的点时，就用红色的高斯分布去拟合。

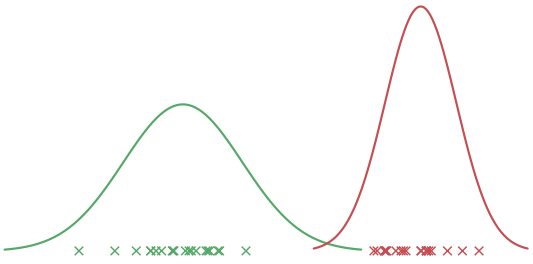

图 3: 混合高斯拟合含有隐变量的数据

于是，在这个模型中，每一个样本点的解释就分为两步：第一，这个样本来自于哪个高斯分布，第二，这个高斯分布的参数是什么。此时**隐变量**就出现了，它存在于第一步中，用于决定样本来自于哪个高斯分布。为了让模型更具一般性，假设有K个高斯分布，用$z_k = 1, 2, \cdots, K$来表示样本来自于哪个高斯分布，参数$\theta = \{\mu_1, \cdots, \mu_K; \sigma_1, \cdots, \sigma_K\}$表示每个高斯分布的参数，那么模型就可以写作：

$$\begin{aligned} p(x|\theta) &= \sum_{i=1}^{K} p(z_k)\mathcal{N}(\mu_k, \sigma_k) \\ \text{s.t.} \quad & \sum_{i=1}^{K} p(z_k) = 1, \end{aligned} \tag{8}$$

即任意一个样本产生的概率既与决定该样本属于哪个分布的隐变量（此处为z_k）有关，也与产生该样本的分布（此处$\mathcal{N}(\mu_k,\sigma_k)$）有关。

如果将观测数据表示为$X=(x_1,x_2,\cdots,x_n)$，则观测数据的对数似然函数为：

$$\begin{aligned}\mathcal{L}(\theta|X)&=\sum_{i=1}^{n}\log p(x|\theta)\\&=\sum_{i=1}^{n}\log\sum_{k=1}^{K}p(z_k)\mathcal{N}(\mu_k,\sigma_k).\end{aligned}\tag{9}$$

考虑求模型参数θ的极大似然估计，即

$$\hat{\theta}=\arg\max_{\theta}\mathcal{L}(\theta|X).\tag{10}$$

按照极大似然估计的步骤，我们应该给$\mathcal{L}(\theta|X)$求偏导并令其为零，然后求解方程得到参数的估计值。

但是从公式(9) 可以看到，由于**隐变量**的引入， $\mathcal{L}(\theta|X)$的表达实里存在求和之后取对数的情况，这时想要求导并得到解析解几乎不可能，因此只有通过迭代的方法求解，而EM算法就是可以用于求解这个问题的一种迭代算法。

1.3 EM算法

对于一个迭代算法而言，必须有一个迭代变量，同时也要建立起一个迭代关系，如公式(11)，从而让计算机发挥运算速度快、适合做重复运算的优势来进行求解。

$$\theta^{(g+1)}=f\left(\theta^{(g)}\right).\tag{11}$$

既然EM算法是一种迭代算法，那么它的迭代变量和迭代关系分别是什么呢？通常会看到教科书或者网课上给出如公式(12) 所示的迭代关系，其中，X代表所有数据样本，Z 代表隐变量，θ是模型参数，其右上角的角标代表迭代轮数。

$$\theta^{(g+1)}=\arg\max_{\theta}\int_{Z}\log P(X,Z|\theta)P\left(Z|X,\theta^{(g)}\right)dZ.\tag{12}$$

但是为什么是这样的迭代关系呢？我们一步步来分析。

首先，假设迭代关系是对数似然函数，既然我们想要最大化它的值，那么就需要保证每一步迭代的结果都比上一次更好，即证明

$$\mathcal{L}\left(\theta^{(g+1)}|X\right)\geq\mathcal{L}\left(\theta^{(g)}|X\right),\tag{13}$$

也可以将其展开，

$$\log P\left(X|\theta^{(g+1)}\right)\geq\log P\left(X|\theta^{(g)}\right).\tag{14}$$

可以看到，目前为止隐变量还没有出现，下面来看对数似然函数本身，我们结合贝叶斯公式引入隐变量Z，试图简化计算，

$$\begin{aligned}\log P(X|\theta)&=\log\left\{\frac{P(X,Z|\theta)}{P(Z|X,\theta)}\right\}\\&=\log P(X,Z|\theta)-\log P(Z|X,\theta),\end{aligned}\tag{15}$$

然后对其两边依照概率$P(Z|X,\theta^{(g)})$求期望，

$$\mathbb{E}_{P(Z|X,\theta^{(g)})}\left[\log P(X|\theta)\right]=\mathbb{E}_{P(Z|X,\theta^{(g)})}\left[\log P(X,Z|\theta)\right]-\mathbb{E}_{P(Z|X,\theta^{(g)})}\left[\log P(Z|X,\theta)\right],\tag{16}$$

将其写为积分的形式，由于$\log P(X|\theta)$中不含积分变量Z，所以先将其提到积分运算之外，

$$\begin{aligned}\int_Z \log P(X|\theta) P\left(Z|X,\theta^{(g)}\right) dZ &= \int_Z \log P(X,Z|\theta) P\left(Z|X,\theta^{(g)}\right) dZ - \int_Z \log P(Z|X,\theta) P\left(Z|X,\theta^{(g)}\right) dZ \\ \log P(X|\theta) \int_Z P\left(Z|X,\theta^{(g)}\right) dZ &= \int_Z \log P(X,Z|\theta) P\left(Z|X,\theta^{(g)}\right) dZ - \int_Z \log P(Z|X,\theta) P\left(Z|X,\theta^{(g)}\right) dZ,\end{aligned} \tag{17}$$

因为变量空间中所有事件的概率和为1，即$\int_Z P(Z|X,\theta^{(g)})dZ = 1$，因此可以将上式化简为

$$\log P(X|\theta) = \underbrace{\int_Z \log P(X,Z|\theta) P\left(Z|X,\theta^{(g)}\right) dZ}_{Q\left(\theta,\theta^{(g)}\right)} - \underbrace{\int_Z \log P(Z|X,\theta) P\left(Z|X,\theta^{(g)}\right) dZ}_{H\left(\theta,\theta^{(g)}\right)}. \tag{18}$$

至此，我们有一个问题还没有考虑：为什么要依照概率$P(Z|X,\theta^{(g)})$求期望呢？ $P(Z|X,\theta^{(g)})$代表了给定观测数据X和第g轮参数估计$\theta^{(g)}$下隐变量数据Z的条件概率分布，由于Z是未观测数据，是用于简化计算的辅助变量，所以必须保证它不能影响结果，也就是说在求第$g+1$轮的参数估计$\theta^{(g+1)}$时，必须保证在给定数据X的情况下， θ是唯一影响对数似然函数取值的因素，这就要求剔除Z的影响，因而对公式(15) 依照概率$P(Z|X,\theta^{(g)})$求期望从而将Z消掉。

回到证明EM算法的收敛性上来，即证明公式(14) 成立，结合公式(18) 可将问题转化为证明下式成立：

$$Q\left(\theta^{(g+1)},\theta^{(g)}\right) - H\left(\theta^{(g+1)},\theta^{(g)}\right) \geq Q\left(\theta^{(g)},\theta^{(g)}\right) - H\left(\theta^{(g)},\theta^{(g)}\right). \tag{19}$$

不难发现，公式(18) 中的$Q\left(\theta,\theta^{(g)}\right)$其实就是EM算法迭代函数中最大化的对象，结合它将公式(12)改写一下可以得到：

$$\forall\theta, Q\left(\theta^{(g+1)},\theta^{(g)}\right) \geq Q\left(\theta,\theta^{(g)}\right). \tag{20}$$

也就是说，左边式子是个值，右边式子是个函数，而且不论右边式子中的变量θ取任何值，都不会大于左边式子的值，因而左边式子的值是右边函数的最大值，所以当$\theta = \theta^{(g)}$时，上式依然成立，即

$$Q\left(\theta^{(g+1)},\theta^{(g)}\right) \geq Q\left(\theta^{(g)},\theta^{(g)}\right). \tag{21}$$

接下来证明H项，既然Q项已经满足公式(21) 了，那么如果H项能满足

$$H\left(\theta^{(g+1)},\theta^{(g)}\right) \leq H\left(\theta^{(g)},\theta^{(g)}\right). \tag{22}$$

我们就可以完成公式(19)证明了。要证明上式成立，我们也可以构造类似公式(20) 的不等式，即

$$\forall\theta, H\left(\theta,\theta^{(g)}\right) \leq H\left(\theta^{(g)},\theta^{(g)}\right). \tag{23}$$

如果这个式子成立，那么就可以同样的令$\theta = \theta^{(g+1)}$从而证明公式(22) 成立。要证这个式子成立，需要参

考公式(18)和Jensen不等式，有

$$
\begin{aligned}
& H\left(\theta, \theta^{(g)}\right)-H\left(\theta^{(g)}, \theta^{(g)}\right) \\
=& \int_{Z} \log P(Z|X, \theta) P\left(Z|X, \theta^{(g)}\right) dZ-\int_{Z} \log P(Z|X, \theta^{(g)}) P\left(Z|X, \theta^{(g)}\right) dZ \\
=& \int_{Z} \log \frac{P(Z|X, \theta)}{P\left(Z|X, \theta^{(g)}\right)} P\left(Z|X, \theta^{(g)}\right) dZ \\
\leq & \log \int_{Z} \frac{P(Z|X, \theta)}{P\left(Z|X, \theta^{(g)}\right)} P\left(Z|X, \theta^{(g)}\right) dZ \\
=& \log \int_{Z} P(Z|X, \theta) dZ \\
=& \log 1 \\
=& 0
\end{aligned}
\tag{24}
$$

因而公式(23) 是成立的，所以公式(19) 得证。综上所述，EM算法是会随着迭代的进行一步步收敛到极值的。

在公式(24) 的证明中，出现不等式的那一步利用了所谓的Jensen不等式，简单来讲，Jensen不等式描述了：在凸函数中（这里指下凸函数），函数的期望不小于期望的函数，或者说在凹函数中，函数的期望不大于期望的函数。

其实它很容易理解，如图4 所示，可以看到，对于点$x1$和$x2$，函数的期望为$p1*f(x1)+p2*f(x2)$，期望的函数为$f(p1*x1+p2*x2)$，显然，假设约束$p1+p2=1, p1>0, p2>0$一直满足，那么无论$p1$和$p2$如何变化， $p1*f(x1)+p2*f(x2) \leq f(p1*x1+p2*x2)$总是成立，即函数的期望总是不大于期望的函数。

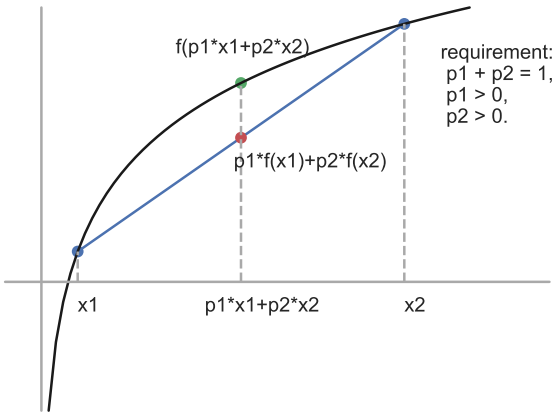

图 4: Jensen不等式图示

结合公式(24) 来说，这里的函数指的是log(x)，而对于连续的变量通常使用积分来替代期望，所以下式很容易成立，

$$
\int_{Z} \log \frac{P(Z|X, \theta)}{P\left(Z|X, \theta^{(g)}\right)} P\left(Z|X, \theta^{(g)}\right) dZ \leq \log \int_{Z} \frac{P(Z|X, \theta)}{P\left(Z|X, \theta^{(g)}\right)} P\left(Z|X, \theta^{(g)}\right) dZ. \tag{25}
$$

这样一来，公式(24) 中后半部分的证明就顺利成章了。

其实，以上的推导也在一定程度上解答了另外一个问题：为什么不直接优化对数似然函数而是选择优化Q函数。因为在迭代时，要保证优化过程的收敛，就必须保证对数似然函数在逐步最大化，而对数

似然函数由两部分组成，

$$\mathcal{L}(\theta|X) = Q\left(\theta, \theta^{(g)}\right) - H\left(\theta, \theta^{(g)}\right), \tag{26}$$

其中，可以证明H函数是逐步变小的，那么$-H$就是逐步变大的，所以要保证对数似然函数逐步变大，只要保证Q函数逐步变大即可，因此，既然可以通过优化形式相对简单的Q函数达到目的，那么就不必要在EM算法中优化整个对数似然函数了。

到这里，EM算法的推导与收敛证明就告一段落了，下面给出EM算法的一般流程。

算法 1 EM算法

输入： 观测变量数据X，隐变量数据Z，联合分布$P(X, Z|\theta)$，条件分布$P(Z|X, \theta)$；

输出： 模型参数θ。

1: 选择参数的初始值$\theta^{(0)}$，开始迭代；

2: E步：记$\theta^{(g)}$为第g次迭代参数θ的估计值，在第$g+1$次迭代的E步，计算

$$\begin{aligned} Q\left(\theta, \theta^{(g)}\right) &= \mathbb{E}_{P(Z|X,\theta^{(g)})}\left[\log P(X, Z|\theta)\right] \\ &= \int_Z \log P(X, Z|\theta) P\left(Z|X, \theta^{(g)}\right) dZ. \end{aligned} \tag{27}$$

这一步主要目的是计算$P(Z|X, \theta^{(g)})$，它代表了给定观测数据X和第g轮参数估计$\theta^{(g)}$下隐变量数据Z的条件概率分布；

3: M步：求使$Q\left(\theta, \theta^{(g)}\right)$极大化的$\theta$，确定第$g+1$轮参数的估计值$\theta^{(g+1)}$

$$\theta^{(g+1)} = \arg\max_{\theta} Q\left(\theta, \theta^{(g)}\right). \tag{28}$$

4: 重复第2 步和第3 步，直到收敛。

算法1 中提到了收敛，那么到底什么时候才算收敛呢？通常会这么做，对于较小的正数$\varepsilon_1, \varepsilon_2$，若满足

$$\left\|\theta^{(g+1)} - \theta^{(g)}\right\| \leq \varepsilon_1 \quad or \quad \left\|Q\left(\theta^{(g+1)}, \theta^{(g)}\right) - Q\left(\theta^{(g)}, \theta^{(g)}\right)\right\| \leq \varepsilon_2, \tag{29}$$

则迭代停止。

1.4 高斯混合模型

高斯混合模型（Guassian Mixture Model, 简称GMM），为单一高斯概率密度函数的延伸，用多个高斯概率密度函数（正态分布曲线）精确地量化变量分布，是将变量分布分解为若干基于高斯概率密度函数（正态分布曲线）分布的统计模型。

以下分别用单高斯模型和混合高斯模型分析同一组样本点。图5 使用单个二维高斯分布来描述数据，椭圆即为二倍标准差的正态分布椭圆。图6 使用两个二维高斯分布来描述数据，分别记为$\mathcal{N}(\mu_1, \Sigma_1)$和$\mathcal{N}(\mu_2, \Sigma_2)$，两个椭圆分别是这两个高斯分布的二倍标准差椭圆。

可以看到使用两个二维高斯分布来描述图中的数据显然更合理。实际上图中的两个聚类的中的点是通过两个不同的正态分布随机生成而来。如果将两个二维高斯分布为$\mathcal{N}(\mu_1, \Sigma_1)$和$\mathcal{N}(\mu_2, \Sigma_2)$合成一个二维的分布，那么就可以用合成后的分布来描述图6 中的所有点。最直观的方法就是对这两个二维高斯分布做线性组合，用线性组合后的分布来描述整个集合中的数据。

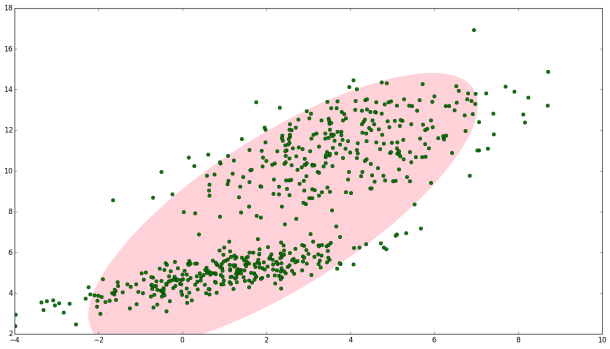

图 5: 单个二维高斯分布拟合数据

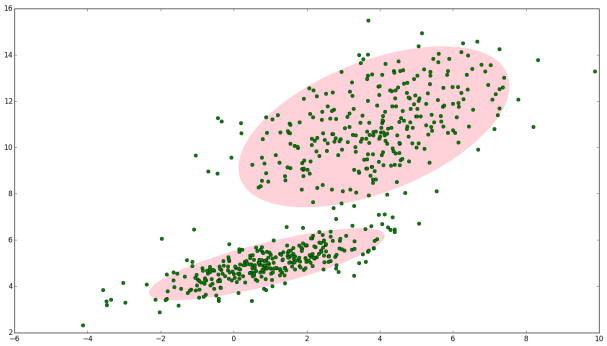

图 6: 两个二维高斯分布拟合数据

提出混合模型主要是为了能更好地近似一些较复杂的样本分布，通过不断增加单个分布的个数，可以任意地逼近任何连续的概率分布，所以我们认为任何样本分布都可以用混合模型来建模。为什么我们要假设数据是由若干个高斯分布组合而成的，而不假设是其他分布呢？实际上不管是什么分布，只要数量取得足够大，这个混合模型就会变得足够复杂，就可以用来逼近任意连续的概率密度分布。只是因为高斯函数具有良好的计算性能，所以GMM被广泛地应用。

GMM与k-means类似，也是常见的聚类算法。不同的是，k-means把每个样本点分配到其中某一个cluster，而GMM是学习出一些概率密度函数，给出这些样本点分配到每个cluster的概率，每个单分布就是一个聚类中心，也称为软聚类。所以GMM 不仅仅可以用于聚类，还可以用于概率密度的估计。

设有随机变量X，则GMM的概率密度函数可以用下式表示：

$$\begin{aligned} p(x) &= \sum_{k=1}^{K} p(k)p(x|k) \\ &= \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k). \end{aligned} \tag{30}$$

其中$\mathcal{N}(x|\mu_k, \Sigma_k)$称为混合模型中的第$k$个组件（component）。$\pi_k$是混合系数（mixture coefficient），且满足$\sum_{k=1}^{K} \pi_k = 1 (0 \leq \pi_k \leq 1)$。实际上，可以认为$\pi_k$就是每个分量$\mathcal{N}(x|\mu_k, \Sigma_k)$的权重。

根据式(30) ，如果我们要从GMM分布中随机地取一个点，需要两步：第一，随机地在这K个组件之中选一个，每个组件被选中的概率实际上就是它的系数π_k；第二，选中了组件之后，再单独地考虑从这个组件的分布中选取一个点。

对一个样本集建立高斯混合模型的过程，就是根据已知样本集X反推高斯混合模型的参数(μ, σ, π)，

这是一个参数估计问题。首先想到用最大似然的方法求解，也就是，要确定参数μ, σ, π使得它所确定的概率分布生成这些样本点的概率最大，这个概率也就是似然函数，如下：

$$p(x) = \prod_{i=1}^{N} x_k. \tag{31}$$

而一般对于单个样本点其概率较小，多个相乘后更小，容易造成浮点数下溢，所以一般是对似然函数求对数，变成加和形式：$\sum_{i=1}^{N} \ln p(x_i)$。这个叫做log似然函数，目标是要最大化它。用log似然函数对参数分别求偏导，令偏导等于0，可求解得参数。然而，GMM的log似然函数是如下形式：

$$\ln L(\mu, \sigma, \pi) = \sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(y_n | \mu_k, \sigma_k). \tag{32}$$

可以看到对数中有求和，直接求导求解将导致一系列复杂的运算，无法求出这个对数似然函数的最大值，故考虑使用EM算法。

那么极大似然估计与EM算法分别适用于什么问题呢？如果我们已经清楚了某个变量服从的高斯分布，而且通过采样得到了这个变量的样本数据，想求高斯分布的参数，这时候极大似然估计可以胜任这个任务；而如果我们要求解的是一个混合模型，只知道混合模型中各个类的分布模型（譬如都是高斯分布）和对应的采样数据，而不知道这些采样数据分别来源于哪一类（隐变量），那这时候就可以借鉴EM算法。EM算法可以用于解决数据缺失的参数估计问题（隐变量的存在实际上就是数据缺失问题，缺失了各个样本来源于哪一类的记录）。

EM算法分两步，第一步先求出要估计参数的粗略值，第二步使用第一步的值最大化似然函数。因此要先求出GMM的似然函数。

考虑GMM生成一个样本点的过程，这里对每个$\mathbf{x}_i$引入隐变量$\mathbf{z}$，$\mathbf{z}$是一个K维向量，如果生成$\mathbf{x}_i$时选择了第k个component，则$\mathbf{z}_k = 1$，其他元素都为0，$\sum_{k=1}^{K} z_k = 1$.

假设$\mathbf{z}$是已知的，则样本集变成了X, Z，要求解的似然函数变成了：

$$p(X, Z | \mu, \sum, \pi) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n | \mu_k, \sum_k)^{z_{nk}}. \tag{33}$$

log似然函数为：

$$\ln p(X, Z | \mu, \sum, \pi) = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbf{z}_{nk} [\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \mu_k, \sum_k)]. \tag{34}$$

可以看到，这次 ln 直接对Gaussian作用，求和在 ln 外面，所以可以直接求最大似然解了。

EM算法估计GMM参数的过程如下：

1．初始化一组参数μ^0, Σ^0, π^0

2．E-step

然而，事实上$\mathbf{z}$是不知道的，我们只是假设$\mathbf{z}$已知。而$\mathbf{z}$的值是通过后验概率观测，所以这里考虑用$\mathbf{z}$值的期望在上述似然函数中代替$\mathbf{z}$。

对于一个样本点$\mathbf{x}$：

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}. \tag{35}$$

$$p(\mathbf{x} | \mathbf{z}_k = 1) = \mathcal{N}(x | \mu_k, \sum_k). \tag{36}$$

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\mu_k, \sum_k)^{z_k}. \tag{37}$$

$$\begin{aligned} p(\mathbf{x}) &= \sum_z p(\mathbf{z}) p(\mathbf{x}|\mathbf{z}) \\ &= \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \sum_k). \end{aligned} \tag{38}$$

后验概率（固定$\mu, \sum, \pi$):

$$p(\mathbf{z}|\mathbf{x}, \mu, \sum, \pi) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} \text{正比于} \prod_{n=1}^{N}\prod_{k=1}^{K} [\pi_k \mathcal{N}(x_n|\mu_k, \sum_k)]^{z_{nk}}. \tag{39}$$

因为$\mathbf{z}_n$之间是相互独立的。

计算$\mathbf{z}$期望$\gamma(\mathbf{z}_{nk})$（z向量只有一个值取1，其余为0）：

$$\begin{aligned} \gamma(\mathbf{z}_{nk}) &= E[\mathbf{z}_{nk}] \\ &= 0 * p(\mathbf{z}_{nk} = 0|\mathbf{x}_n) + 1 * p(\mathbf{z}_{nk} = 1|\mathbf{x}_n) \\ &= p(\mathbf{z}_{nk} = 1|\mathbf{x}_n) \\ &= \frac{p(\mathbf{z}_{nk} = 1)p(\mathbf{x}_n|\mathbf{z}_{nk} = 1)}{p(\mathbf{x}_n)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}|\mu_k, \sum_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\mu_j, \sum_j)}. \end{aligned} \tag{40}$$

将$\mathbf{z}$值用期望代替，则待求解的log似然函数(34)式变为：

$$E_z[\ln p(X, Z|\mu, \sum, \pi)] = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(\mathbf{z}_{nk})[\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\mu_k, \sum_k)]. \tag{41}$$

3. M-step

现在可以最大化似然函数求解参数了，首先对μ求偏导，令偏导等于0，可得：

$$\sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(\mathbf{z}_{nk}) \sum_k (\mathbf{x}_n - \mu_k) = 0. \tag{42}$$

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(\mathbf{z}_{nk}) \mathbf{x}_n, \text{其中} N_k = \sum_{n=1}^{N} \gamma(\mathbf{z}_{nk}). \tag{43}$$

再对$\sum_k$求偏导，令偏导等于0，可得：

$$\sum_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(\mathbf{z}_{nk})(\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^T. \tag{44}$$

接下来还需求解π，注意到π需满足$\sum_{k=1}^{K} \pi_k = 1$，所以这是一个带等式约束的最大值问题，使用拉格朗日乘数法。

构造拉格朗日函数：

$$L = \ln p(X|\pi, \mu, \sum) + \lambda(\sum_{k=1}^{K} \pi_k - 1). \tag{45}$$

对π求导，令导数为0：

$$\sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}|\mu_k, \sum_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\mu_j, \sum_j)} + \lambda = 0. \tag{46}$$

两边同乘π_k得：

$$\sum_{n=1}^{N} \gamma(\mathbf{z}_{nk}) + \lambda \pi_k = 0. \tag{47}$$

$$N_k + \lambda \pi_k = 0. \tag{48}$$

两边对k求和：

$$\sum_{k=1}^{K} N_k + \sum_{k=1}^{K} \lambda \pi_k = 0. \tag{49}$$

$$N + \lambda = 0. \tag{50}$$

可得：$\lambda = -N$

代入可得：$\pi_k = \frac{N_k}{N}$.

4. 检查是否收敛

重复E-step和M-step两步，直到收敛，即可求得一个局部最优解。

2 应用问题描述

2.1 GMM的应用场景

GMM在实际应用中十分广泛，比如语音识别、图像生成、以及各种聚类。

在语音识别方面，早于1995年，Douglas A. Reynolds和R.C.Rose就提出论文，基于GMM实现了独立于文本的语音识别。构成高斯混合的各个高斯分量用于对说话者的频谱从不同角度建模。他们在一个有49名讲话者的电话会议的语音上进行了测试，使用五秒的清晰语音即可获得96.8%的识别准确度，在当时取得了极佳的效果。同年，Douglas A. Reynolds将GMM在更大的测试集上进行了测试。在TIMIT和NTIMIT数据库上（630位说话人）的闭集识别精度分别为99.5%和60.7%，在Switchboard数据集（113位说话人）上的识别准确率为82.8%。2000年，Douglas A. Reynolds和Thomas F.Quatieri以及Robert B.Dunn发表了论文。对已经在几个NIST语音识别评估（NIST Speaker Recognition Evaluations，SREs）中取得良好表现的语音识别模型的主要结构进行了描述，该模型由麻省理工学院林肯实验室开发，基于GMM。

在计算机视觉方面，2004年 Z. Zivkovic 提出了基于GMM的一种高效的自适应算法来进行背景提取，该模型可以不断更新参数，同时为每个像素选择适当数量的成分（component）。2005年Dar-Shyang Lee试图提高自适应高斯混合的收敛速度而不影响模型的稳定性。他将全局静态保留因子（global static retention factor）替换为在每帧处为每个高斯分布计算的自适应学习速率。结果显示该方法在合成视频数据和真实视频数据上都有更好的表现。该方法还可以与背景提取的统计框架结合，得到更好的图像分割性能。

在聚类方面，在《基于高斯混合模型的层次聚类算法》一文中，作者提到计算高斯混合分布中每两个组成成分的重叠度，然后根据重叠的程度，即重叠率是否大于一个阈值，决定是否将两个分布合并。如果合并了，就重新更新均值和方差。再具体的生产应用中，GMM算法可用于图像聚类以及文本聚类等任务。

2.2 实验创想

由于聚类任务是GMM算法的一大重要应用，在此我们选取图像聚类为导向，进行GMM算法的实验验证。我们先计划使用比较简单的MNIST数据集，验证GMM算法在的聚类任务中的有效性。然后我们再选取一些彩色图像进行图像分割任务，从而让GMM的实用性得以体现。

3 实验

3.1 数据集与评价指标

3.1.1 数据集

本实验采用的MNIST数据库是由Yann LeCun教授提供的手写数字数据库文件，该数据集包含了60000张训练图像和10000张测试图像。这些数字已经过尺寸标准化并位于图像中心，且图像是28*28大小的灰度图像，每个像素是一个八位字节。数据集共包括四个文件，一个训练图片集train-images-idx3-ubyte.gz，一个训练标签集train-labels-idx1-ubyte.gz，一个测试图片集t10k-images-idx3-ubyte.gz，一个测试标签集t10k-labels-idx1-ubyte.gz。上述四个文件直接解压就可以使用了。

3.1.2 评价指标

在本实验中，手写数字识别的评价指标采用了调整兰德系数(Adjusted rand index)，正确率(accuracy)，查准率(precision)，查全率(recall)，F1分数(F1-Score)。

兰德系数(Rand index)需要给定实际类别信息 C，假设K是聚类结果，a表示在C与K中都是同类别的元素对数，b表示在C与K中都是不同类别的元素对数，则兰德系数为：

$$\mathrm{RI} = \frac{a+b}{C_2^{n_{samples}}}. \tag{51}$$

其中$C_2^{n_{samples}}$为数据集中可以组成的总元素对数，RI取值范围为$[0,1]$，值越大意味着聚类结果与真实情况越吻合。调整兰德系数(Adjusted rand index)具有更高的区分度：

$$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\max(\mathrm{RI}) - E[\mathrm{RI}]}. \tag{52}$$

ARI取值范围为$[-1,1]$，值越大意味着聚类结果与真实情况越吻合。从广义的角度来讲，ARI衡量的是两个数据分布的吻合程度。

对于二分类问题，可将样例根据其真实类别和分类器预测类别划分为：

- TP：真实类别为正例，预测类别为正例。
- FP：真实类别为负例，预测类别为正例。
- FN：真实类别为正例，预测类别为负例。
- TN：真实类别为负例，预测类别为负例。

正确率是我们最常见的评价指标，表示被分对的样本数在所有的样本数中所占比例，即：

$$accuracy = \frac{TP+TF}{TP+FP+FN+TN}. \tag{53}$$

通常来说，正确率越高，分类器越好。

查准率是精确性的度量，表示被分为正例的示例中实际为正例的比例，即：

$$precision = \frac{TP}{TP + FP}. \tag{54}$$

查全率也叫做敏感度，表示正例被分类器正确探测出的比率，即：

$$recall = \frac{TP}{TP + FN}. \tag{55}$$

F1分数是统计学中用来衡量二分类模型精确度的一种指标，它同时兼顾了分类模型的查准率和查全率。F1分数可以看作是模型查准率和查全率的一种加权平均，它的最大值是1，最小值是0。数学定义如下：

$$F1 = \frac{2 \times precision \times recall}{precision + recall}. \tag{56}$$

对于多分类问题，我们可以看作是n个二分类问题。正确率的计算方式显然与二分类相同，对于其他的评价指标，分为宏平均(Macro-averaging)和微平均(Micro-averaging)两种情况来进行综合考察。

宏平均先分别计算各类的precision和recall，得到各类的F1，然后再计算F1的平均值。由此可以得到宏查准率(macro-P)、宏查全率(macro-R)、宏F1(macro-F1)：

$$marcoP = \frac{1}{n}\sum_{1}^{n} P_i. \tag{57}$$

$$marcoR = \frac{1}{n}\sum_{1}^{n} R_i. \tag{58}$$

$$marcoF1 = \frac{1}{n}\sum_{1}^{n} F1_i. \tag{59}$$

微平均将n个二分类评价的TP、FP、FN对应相加，计算出所有类别总的Precision和Recall，再计算得到F1。经过推导可知，此时满足：

$$microP = microR = microF1 = accuracy \tag{60}$$

3.2 实验设置

3.3 实验环境

3.4 实验结果与讨论

4 展望

xxx

5 大作业总结

本学期已经接近尾声，这份大作业也贯穿了《数据科学基础》课程的始终。三个月以来，我们小组分工合作、齐心协力，一起查阅资料，分工完成任务，对难以理解之处进行小组讨论，在这个过程中我们都从中学习了很多。完成本次大作业需要对基础理论有充分的理解，进而在此基础上进行实际的应用，

对于我们的理论和实际技能都是一次很好的提升机会。课程大作业参照毕业设计的方式，按照开题、中期、答辩设置三个节点，这样有利于我们在时间上对大作业有一个整体的把握，更好地协调小组的工作进度，让大作业有条不紊地完成。

由于疫情的原因，我们的上课方式改成了线上进行，虽然不能在课堂上进行面对面的交流，但整体感觉也是很扎实的。每一章的知识点都在PPT和学习指导中详细介绍，在每周二的课堂时间老师也会细细梳理一遍，然后我们在总结学习笔记的过程中再进行查漏补缺，基础知识掌握得很牢固。如今作为研究生的我们尚未返校，学习具有极大的自主性，自由支配的时间和空间变得更多，学习形式也更为灵活多样，在自学的过程中我们也应该注重知识结构的把握，有意识地锻炼自己的科学研究能力，不断提升理论水准，并将知识与能力结合。学习是一个长期积累的过程，在以后的工作、生活中都就应不断地学习，努力提高自我知识和综合素质。

本学期以来，我们都从《数据科学基础》课程中学习到了很多知识，对于这门课程我们还有一些小小的建议。由于线上的特殊授课方式，老师和同学之间的交流互动似乎过少了。如果在课堂上增加一些互动的环节，或许同学们能更加积极地参与其中，老师也能更准确地掌握学生的学习状况。另外，学习笔记中大部分都是定义、原理、公式推导等理论型的题目，且大多都能在PPT的对应章节找到答案，这样可能会导致学生在学习时只是机械地做一些“知识搬运”的工作。可以考虑在作业中也增加一些开放性较强、偏应用型的题目，鼓励学生多思考，灵活运用所学的理论知识。

感谢老师和助教同学一学期以来的辛苦付出，你们辛苦了！

6 小组同学分工说明

xxx

参考文献

[1] Wu M, Zhang Z. Handwritten digit classification using the mnist data set[J]. Course project CSE802: Pattern Classification & Analysis, 2010.

[2] Gupta L, Sortrakul T. A Gaussian-mixture-based image segmentation algorithm[J]. Pattern Recognition, 1998, 31(3): 315-325.

[3] McLachlan G J, Krishnan T. The EM algorithm and extensions[M]. John Wiley & Sons, 2007.

[4] Dempster A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm[J]. Journal of the Royal Statistical Society: Series B (Methodological), 1977, 39(1): 1-22.

[5] Goodfellow I, Bengio Y, Courville A. Deep learning[M]. MIT press, 2016.

[6] 周志华. 机器学习[M]. 北京:清华大学出版社, 2016.

[7] 李航. 统计学习方法[M]. 北京:清华大学出版社, 2012.

[8] Bishop C M. Pattern recognition and machine learning[M]. springer, 2006.

[9] XIAO Z, WANG J. Image classification algorithm based on PCA and GMM[J]. Computer Engineering and Design, 2006, 27(11): 1951-1953.

[10] Bouman C A, Shapiro M, Cook G W, et al. Cluster: An unsupervised algorithm for modeling Gaussian mixtures[J]. 1997.

[11] 陈雪峰. 图像高斯混合模型的判别学习方法[D].北京理工大学,2009.

[12] Xuan G, Zhang W, Chai P. EM algorithms of Gaussian mixture model and hidden Markov model[C]//Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205). IEEE, 2001, 1: 145-148.

[13] Reynolds D A. Gaussian Mixture Models[J]. Encyclopedia of biometrics, 2009, 741.