GMM

GMM和k-mean的区别

“我不能创造的，我就无法理解。” —Richard Feynman

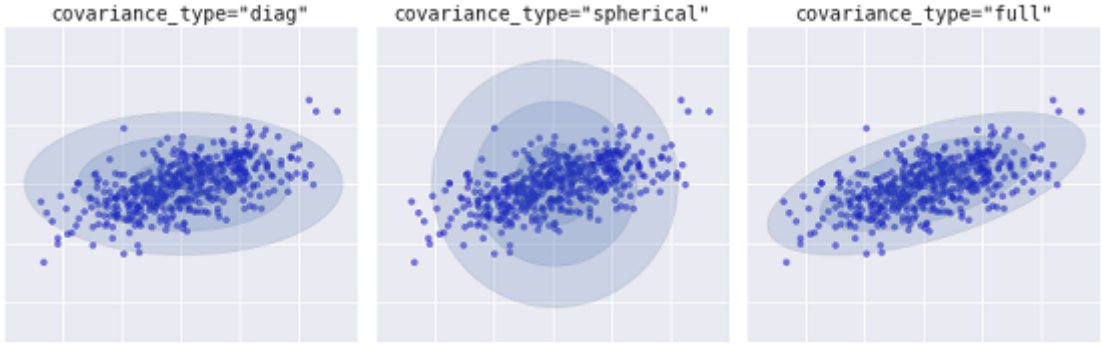
从原理上考虑“理解”模型。

k-means 聚类方法相对简单，易于理解。但是，它有许多缺点，不适合我们的情况。

- 特别是，由于它不是概率的，所以在许多实际情况下它的性能较差。
- 假设这种方法将圆（或超球体）放置在给定数量的质心周围，其半径由簇的最外点确定。此半径严格限制每个簇的点集。因此，所有的簇只能用圆和超球体来描述，而真正的簇并不总是满足这个标准（因为它们可以是椭圆形或椭圆的形式）。这将导致不同簇值的重叠。

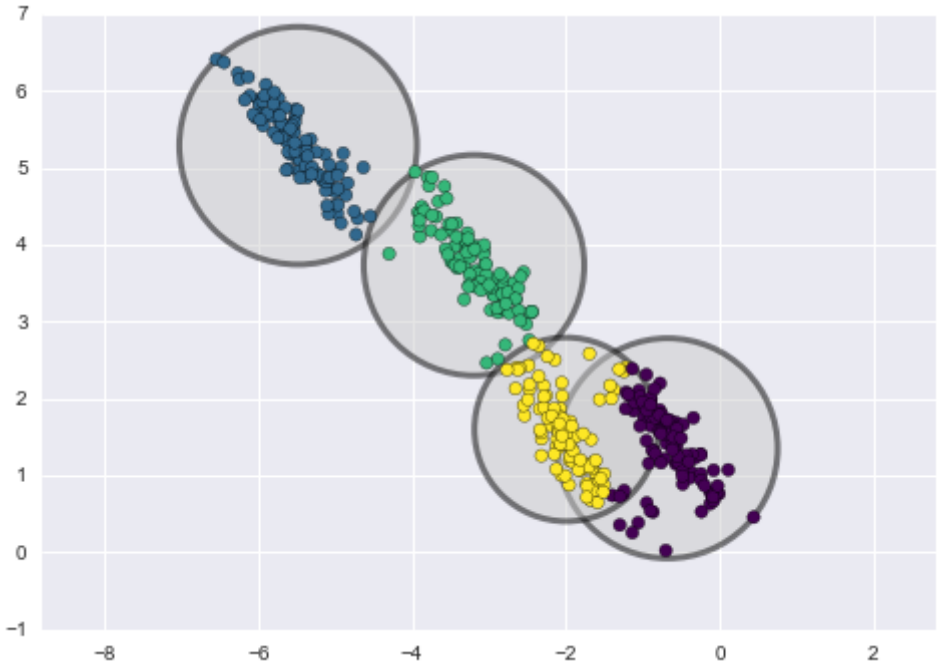
更先进的算法是高斯混合模型。

- 该模型搜索多元高斯概率分布的混合，从而对数据集进行最佳建模。因为模型是概率的，所以这会输出一个实例被分类为特定集群的概率。
- 另外，每个簇不是与一个严格定义的球体相关联，而是与一个光滑的高斯模型相关联，该模型不仅可以表示为圆，还可以表示为在空间中任意定向的椭圆。

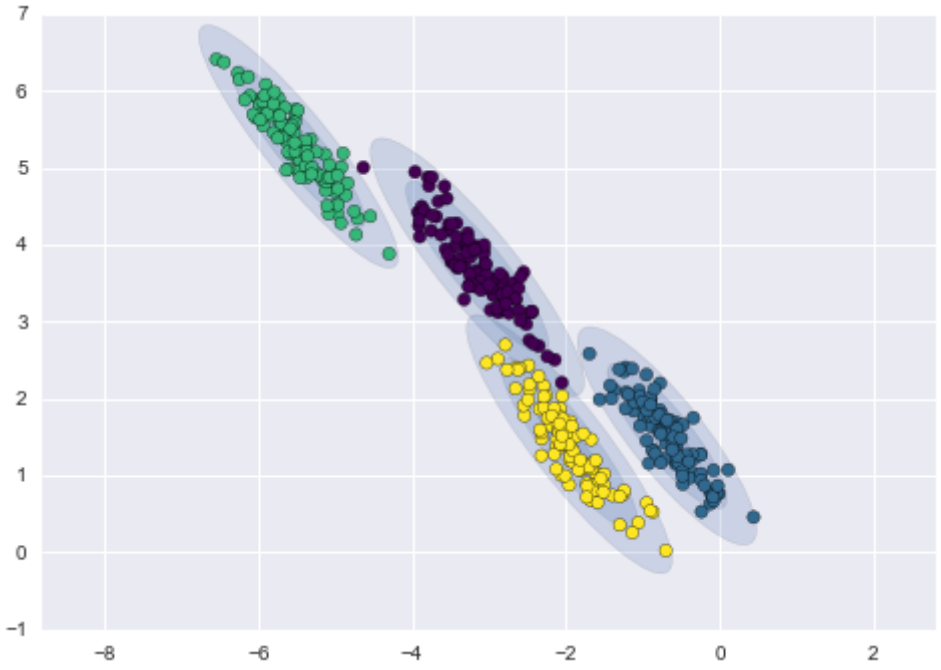


不同类型的概率模型，取决于 covariance_type (协变量类型)

下面是通过k-means和GMM获得的聚类的比较



K-means 聚类



GMM 聚类

如何改进模型?

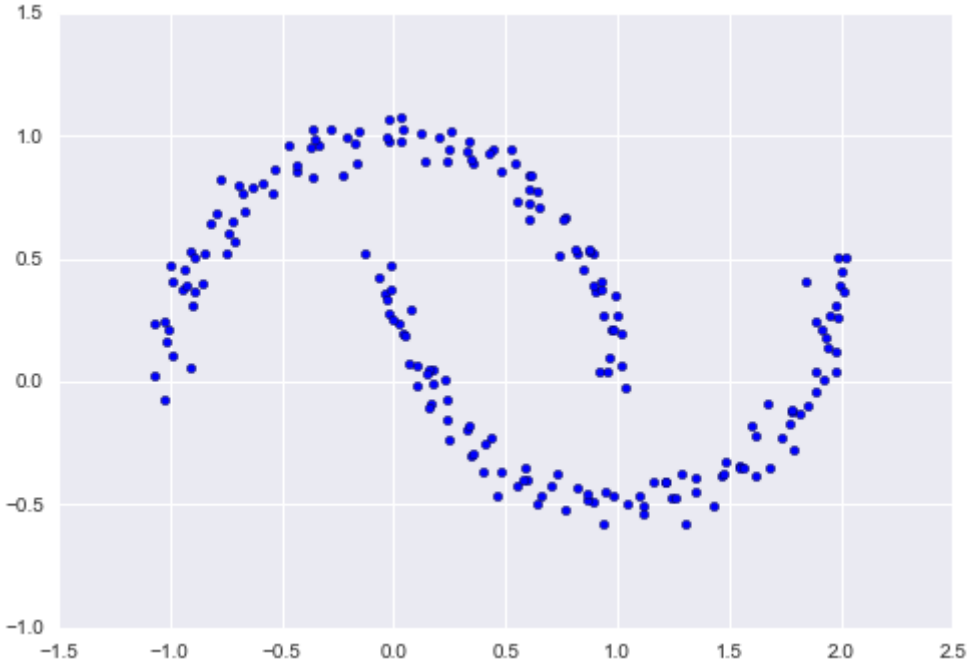
模型训练包含许多随机成分，这些成分每次都是不同的。例如，交易的随机抽样、GMM训练（也有随机性因素）、后验GMM分布的随机抽样和 CatBoost 训练（也有随机性因素）。因此，可以重新启动几次以获得最佳结果。如果无法获得稳定的模型，则应调整 LOOK_BACK 参数和移动平均数及其周期数。您还可以更改从GMM接收的样本数，以及训练和测试间隔。

GMM 作为密度估计

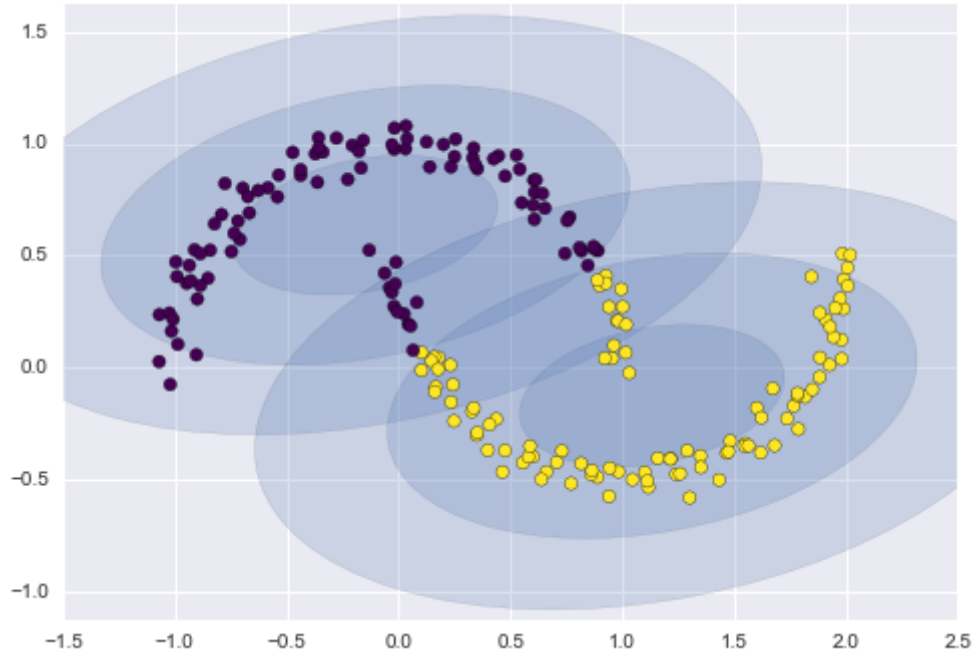
事实上，高斯混合模型（GMM）算法并不是真正的聚类算法，因为它的主要任务是估计概率密度。该模型中的聚类表示为从描述该数据的概率分布生成的数据。因此，在估计每个聚类的概率密度之后，可以从这些分布生成新的数据集。这些数据集将与原始数据相似，但它们具有或多或少的可变性，并且具有较少的异常值。此外，在许多情况下，数据集的相关性会降低。

密度估计的作用

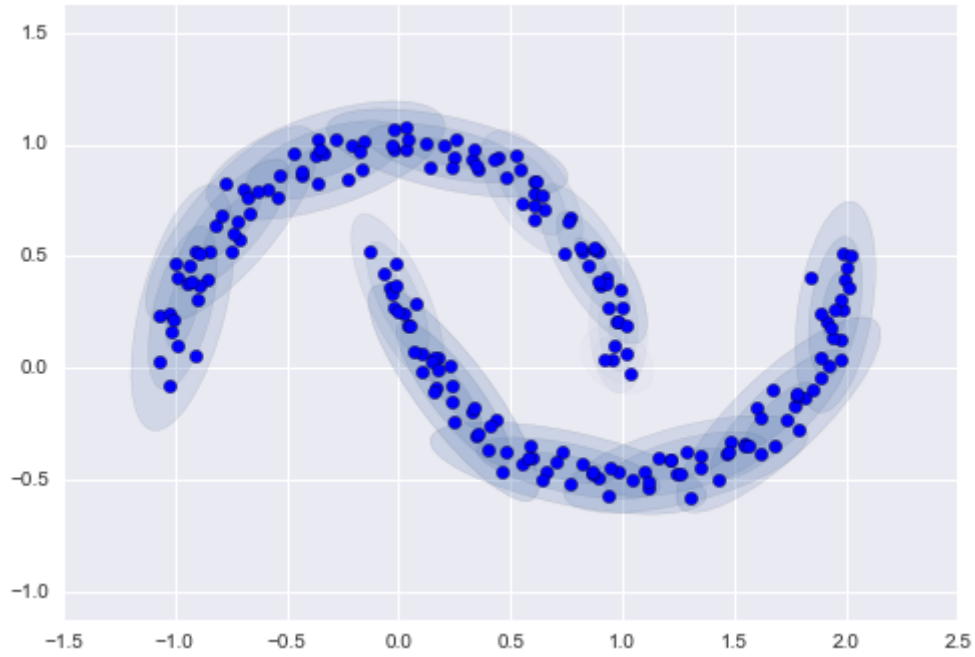
如下数据中，K-means 聚类必然无法成功聚类。



如果我们尝试将其与被视为聚类模型的双组分 GMM 拟合，结果并不是特别有用：



但是，如果我们使用更多的组件并忽略集群标签，我们会发现更接近输入数据的拟合：



在这里，16 个高斯的混合不是用于查找单独的数据簇，而是用于对输入数据的整体*分布*进行建模。 这是一个分布的生成模型，这意味着 GMM 为我们提供了生成与输入分布类似的新随机数据的配方。
GMM 是一种方便的灵活方法，可以对任意的多维数据分布进行建模。

数学原理

高斯混合模型(Gaussian Mixture Model)是机器学习中一种常用的聚类算法，以下推导了其参数估计的过程。主要参考Christopher M. Bishop的《Pattern Recognition and Machine

Learning》。
以粗体小写字母表示向量，粗体大写字母表示矩阵；标量不加粗，大写表示常数。

1. 高斯分布

高斯分布(Gaussian distribution)，也称为正态分布(normal distribution)，是一种常用的连续变量分布的模型。若单个随机变量 x 服从均值为 μ，方差为 σ^2 的高斯分布，记为 $x \sim N\left(\mu, \sigma^2\right)$，则其概率密度函数为：

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

对于一个 D 维的向量 $\mathbf{x}$，若其各元素服从均值为向量 μ，协方差矩阵为 $\mathbf{\Sigma}$ 的多元高斯分布，记为 $\mathbf{x} \sim N\left(\mu, \mathbf{\Sigma}\right)$，则概率密度为：

$$p(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^{\mathbf{T}}\mathbf{\Sigma^{-1}}(\mathbf{x}-\mu)\right\} \tag{1}$$

其中 μ 为 D 维均值向量，$\mathbf{\Sigma}$ 为 $D \times D$ 的协方差矩阵，$|\mathbf{\Sigma}|$ 表示 $\mathbf{\Sigma}$ 的行列式。

(1)式中，指数部分的二次型 $\Delta^2 = (\mathbf{x}-\mu)^{\mathbf{T}}\mathbf{\Sigma^{-1}}(\mathbf{x}-\mu)$ 称为 $\mathbf{x}$ 到 μ 的马哈拉诺比斯距离(马氏距离，Mahalanobis distance)；当 $\mathbf{\Sigma}$ 为单位矩阵时退化为欧几里得距离(Euclidean distance)。多元高斯分布密度函数的等高线即 Δ^2 为常数时 $\mathbf{x}$ 的方程，是椭球方程(Ellipsoid - Wikipedia)。

2. 高斯混合模型(Gaussian Mixture Model)

多个高斯分布的线性叠加能拟合非常复杂的密度函数；通过足够多的高斯分布叠加，并调节它们的均值，协方差矩阵，以及线性组合的系数，可以精确地逼近任意连续密度([1], Section 2.3.9, p111)。

我们考虑 K 个高斯分布的线性叠加，这个高斯混合分布(Gaussian mixture distiburion)的概率密度函数为：

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k p\left(\mathbf{x}|\mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right) \tag{2}$$

其中，$p\left(\mathbf{x}|\mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right)$ 表示参数为 $\mu_{\mathbf{k}}, \mathbf{\Sigma_k}$ 的高斯分布的概率密度。

我们称(2)式为一个高斯混合(Mixture of Gaussians, Gaussian Mixture)。其中每个高斯密度函数称为混合的一个分模型(component)，有自己的均值 $\mu_{\mathbf{k}}$ 和协方差矩阵 $\mathbf{\Sigma_k}$。

(2)式中的参数 π_k 是模型的混合系数(mixing coefficients)。将(2)式左右两侧对 $\mathbf{x}$ 积分，得到

$$\sum_{k=1}^{K} \pi_k = 1$$

此外，由于 $p(\mathbf{x}) \geq 0$ ， $p\left(\mathbf{x}|\mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right) \geq 0$ ，所以 $\pi_k \geq 0$ 。即混合系数应满足 $0 \leq \pi_k \leq 1$ 。
(更一般地，混合模型也可以是其他任意分布的叠加。)

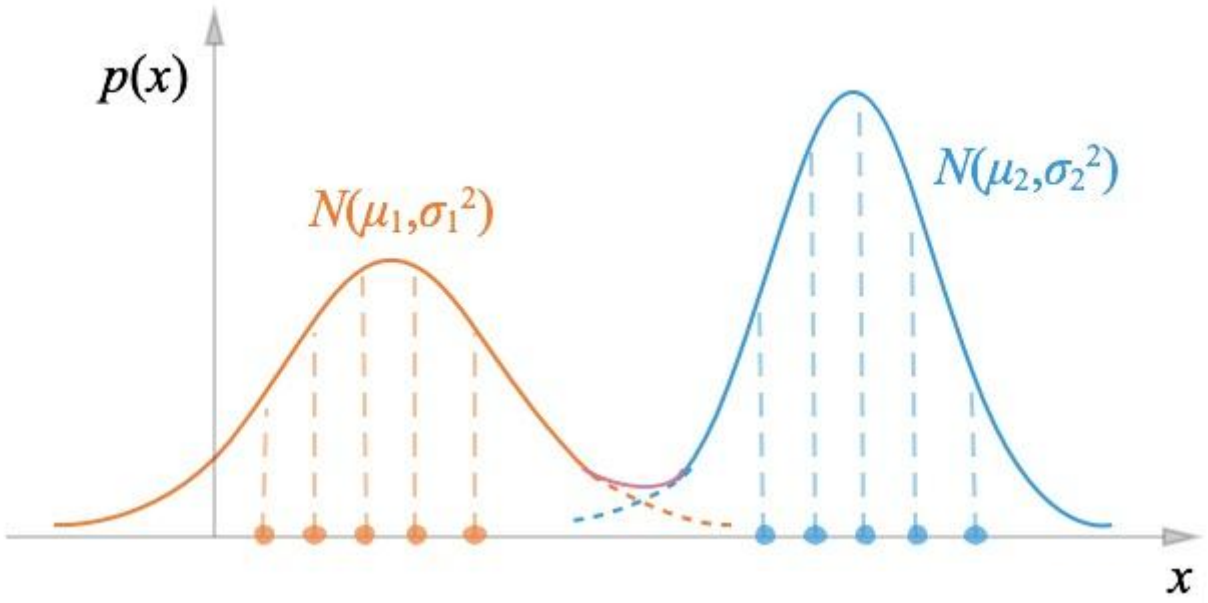
由全概率公式， $\mathbf{x}$ 的边缘分布(marginal distribution)的概率密度为：

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k)p(\mathbf{x}|k)$$

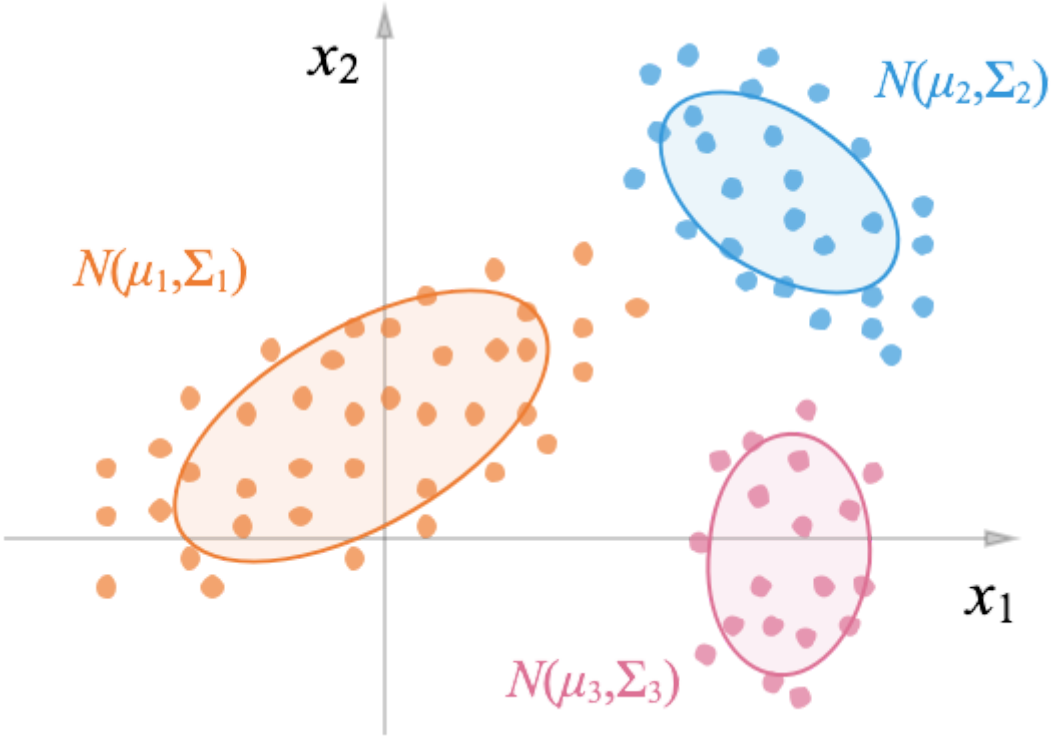
上式与(2)式等价，其中 $p(k) = \pi_k$ 可以看作选择第 k 个分模型的先验概率(prior probability)，$p(\mathbf{x}|k)$ 是 $\mathbf{x}$ 对 k 的条件概率密度。

在观测到 $\mathbf{x}$ 后，其来自第 k 个分模型的后验概率(posterior probability) $p(k|\mathbf{x})$ 称为第 k 个分模型的响应度(responsibility)。

下图所示为包含两个一维分模型的高斯混合：



两个一维分布模型的高斯混合。



如上是三个二维分布模型的高斯混合。

3. 隐变量 & 完全数据

引入一个 K 维的二值型随机变量 $\mathbf{z}=[z_1,\ldots,z_K]$，来表示样本 $\mathbf{x}$ 由哪一分模型（子模型）产生。 $\mathbf{z}$ 满足这样的条件：$z_k\in\{0,1\}$，且 $\sum_{k=1}^{K}z_k=1$，即其 K 个元素中，有且只有一个为1，其余为0。 $z_k=1$ 表示样本 $\mathbf{x}$ 由分模型 k 抽样得到。可以看出， $\mathbf{z}$ 一共有 K 种可能的状态。

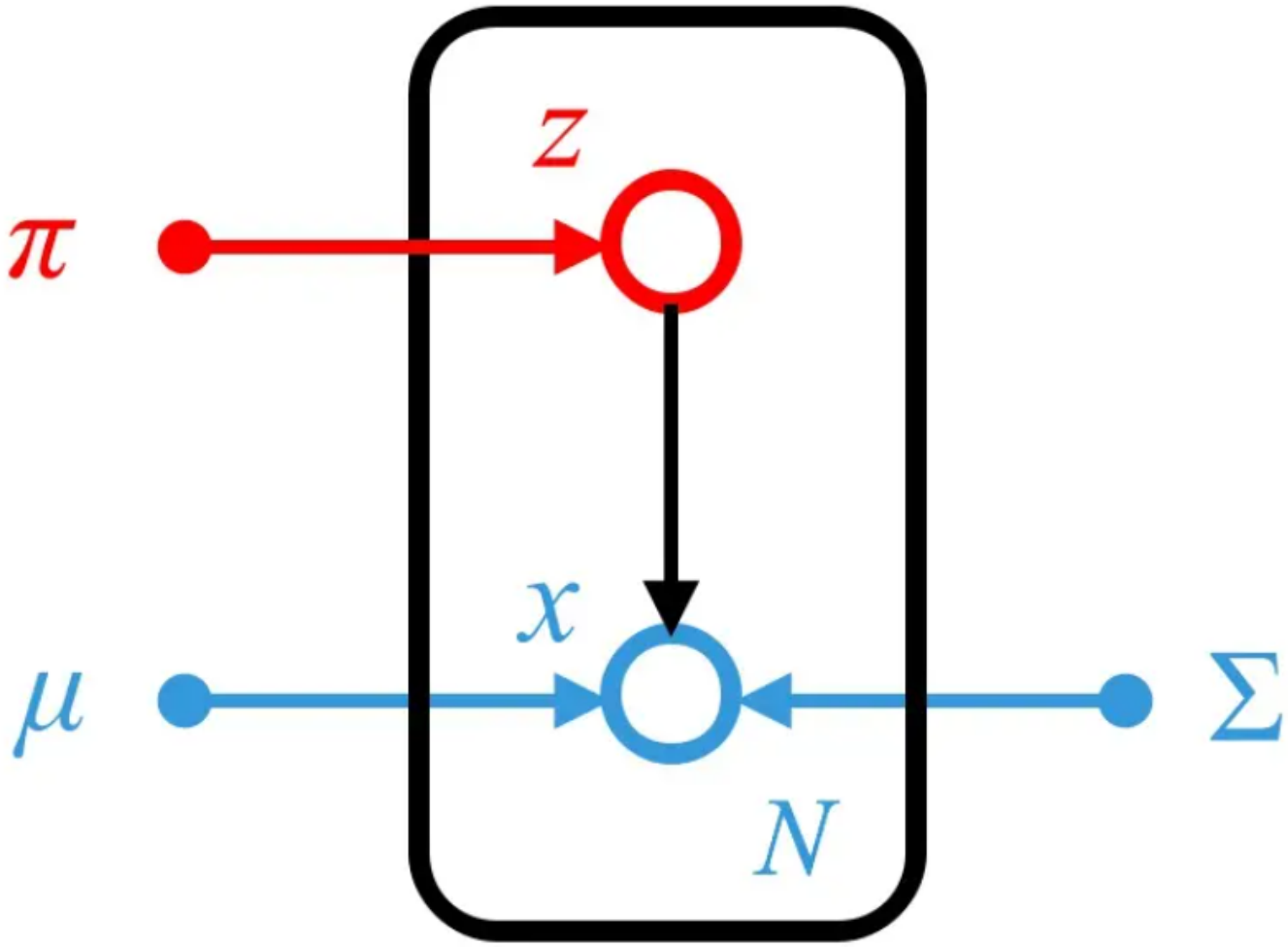
$\mathbf{z}$ 的边缘分布由混合系数给出： $p\left(z_k=1\right)=\pi_k$ 。也可写成如下形式：

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

考虑由以下方式产生样本 $\mathbf{x}$：

1. 先以离散分布 $p(\mathbf{z})$ 抽样得到变量 $\mathbf{z}$；
2. 设根据 $\mathbf{z}$ 的取值选择了第 k 个分模型，则以高斯分布 $p\left(\mathbf{x}|\mu_{\mathbf{k}},\mathbf{\Sigma_k}\right)$ 抽样得到 $\mathbf{x}$ 。

记高斯混合模型的参数为 $\pi\equiv\{\pi_1,\ldots,\pi_K\}$， $\mu\equiv\{\mu_{\mathbf{1}},\ldots,\mu_{\mathbf{K}}\}$， $\mathbf{\Sigma}\equiv\{\mathbf{\Sigma_1},\ldots,\mathbf{\Sigma_K}\}$，则这个过程可由如下的graphical model表示[1]：



高斯混合模型的graphical model

变量 $\mathbf{z}$ 称为隐变量(latent variable)，包含 $(\mathbf{x}, \mathbf{z})$ 取值的样本称为完全数据(complete data)，只含有 $\mathbf{x}$ 取值的样本称为不完全数据(incomplete data)，

给定 $\mathbf{z}$ 后 $\mathbf{x}$ 的条件概率密度为：

$$p(\mathbf{x}|z_k = 1) = p\left(\mathbf{x}|\mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right)$$

或者写成如下形式：

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} p(\mathbf{x}|\mu_{\mathbf{k}}, \mathbf{\Sigma_k})^{z_k}$$

$\mathbf{x}$ 的边缘分布为联合概率分布 $p(\mathbf{x}, \mathbf{z})$ 对 $\mathbf{z}$ 的所有可能状态求和：

$$
\begin{aligned}
p(\mathbf{x}) &= \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) = \sum_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x}|\mathbf{z}) \\
&= \sum_{\mathbf{z}} \left(\prod_{k=1}^{K} \pi_k^{z_k} \prod_{k=1}^{K} p(\mathbf{x}|\mu_\mathbf{k}, \mathbf{\Sigma_k})^{z_k} \right) \\
&= \sum_{\mathbf{z}} \left(\prod_{k=1}^{K} \left[\pi_k p\left(\mathbf{x}|\mu_\mathbf{k}, \mathbf{\Sigma_k}\right) \right]^{z_k} \right) \\
&= \sum_{k=1}^{K} \pi_k p\left(\mathbf{x}|\mu_\mathbf{k}, \mathbf{\Sigma_k}\right)
\end{aligned}
$$

也可由下面的式子得到：

$$
p(\mathbf{x}) = \sum_{j=1}^{K} p(z_j = 1) p(\mathbf{x}|z_j = 1) = \sum_{j=1}^{K} \pi_j p\left(\mathbf{x}|\mu_\mathbf{j}, \mathbf{\Sigma_j}\right)
$$

这表明 $\mathbf{x}$ 的边缘分布就是高斯混合的形式。

4. 后验概率 & 响应度

根据贝叶斯定理(Bayes' theorem - Wikipedia)，观测到 $\mathbf{x}$ 后，其来自第 k 个分模型的后验概率(posterior responsibility)为：

$$
p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1) p(\mathbf{x}|z_k = 1)}{p(\mathbf{x})} \qquad = \frac{\pi_k p\left(\mathbf{x}|\mu_\mathbf{k}, \mathbf{\Sigma_k}\right)}{\sum_{j=1}^{K} \pi_j p\left(\mathbf{x}|\mu_\mathbf{j}, \mathbf{\Sigma_j}\right)} \tag{3}
$$

上式中， $p(z_k = 1|\mathbf{x})$ 和 $p(z_k = 1)$ 为概率， $p(\mathbf{x}|z_k = 1)$ 和 $p(\mathbf{x})$ 为概率密度。

将上式定义为： $\gamma(z_k) \equiv p(z_k = 1|\mathbf{x})$ ，称为第 k 个分模型对 $\mathbf{x}$ 的响应度(responsibility)[2]。

对于样本集 $\mathbf{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$ ，记 $\mathbf{x_n}$ 对应的隐变量为 $\mathbf{z_n} = [z_{n1}, \ldots, z_{nK}]$ ， $n = 1, \ldots, N$ ，则第 k 个分模型对 $\mathbf{x_n}$ 的响应度为：

$$
\gamma(z_{nk}) \equiv p(z_{nk} = 1|\mathbf{x_n}) = \frac{\pi_k p\left(\mathbf{x_n}|\mu_\mathbf{k}, \mathbf{\Sigma_k}\right)}{\sum_{j=1}^{K} \pi_j p\left(\mathbf{x_n}|\mu_\mathbf{j}, \mathbf{\Sigma_j}\right)} \tag{4}
$$

5. 对数似然函数 & 最大似然估计

现在有一个样本集 $\mathbf{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$ ，我们假设 $\mathbf{X}$ 是由一个高斯混合模型产生的，要估计这个模型的参数： $\pi \equiv \{\pi_1, \ldots, \pi_K\}$ ， $\mu \equiv \{\mu_\mathbf{1}, \ldots, \mu_\mathbf{K}\}$ ， $\mathbf{\Sigma} \equiv \{\mathbf{\Sigma_1}, \ldots, \mathbf{\Sigma_K}\}$ 。

样本集 $\mathbf{X}$ (不完全数据)的似然函数(likelihood function)为：

$$
p(\mathbf{X}|\pi, \mu, \mathbf{\Sigma}) = \prod_{n=1}^{N} \left[\sum_{k=1}^{K} \pi_k p\left(\mathbf{x_n}|\mu_\mathbf{k}, \mathbf{\Sigma_k}\right) \right]
$$

似然函数中的连乘求导比较麻烦，取自然对数将其转换成对数的和，得到对数似然函数(the log of the likelihood function)：

$$\ln p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma}) = \sum_{n=1}^{N} \ln \left[\sum_{k=1}^{K} \pi_k p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right) \right] \tag{5}$$

其中，

$$p(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma_k}|^{1/2}} \exp\left\{ -\frac{1}{2} (\mathbf{x_n}-\mu_\mathbf{k})^\mathbf{T} \mathbf{\Sigma_k^{-1}} (\mathbf{x_n}-\mu_\mathbf{k}) \right\}$$

最大似然估计(maximum likelihood estimation)，即通过求似然函数的最大值来估计概率模型的参数。用最大似然估计来计算高斯混合模型的参数，形成如下的优化问题：
$\max_{\pi,\mu,\mathbf{\Sigma}} \ln p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma})$
$s.t. \sum_{k=1}^{K} \pi_k = 1$

采用拉格朗日乘子法来求极值，拉格朗日函数为：

$$L(\pi,\mu,\mathbf{\Sigma}) = \ln p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

先将 $L(\pi,\mu,\mathbf{\Sigma})$ 对 $\mu_\mathbf{k}$ 求梯度。因为

$$\frac{\partial p(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k})}{\partial \mu_\mathbf{k}} = p(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}) \left[\Sigma_k^{-1} (\mathbf{x_n}-\mu_\mathbf{k}) \right]$$

所以，

$$\begin{aligned}
\frac{\partial L}{\partial \mu_\mathbf{k}} &= \frac{\partial \ln p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma})}{\partial \mu_\mathbf{k}} \\
&= \frac{\partial \left(\sum_{n=1}^{N} \ln \left[\sum_{k=1}^{K} \pi_k p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right) \right] \right)}{\partial \mu_\mathbf{k}} \\
&= \sum_{n=1}^{N} \frac{\partial \ln \left[\sum_{k=1}^{K} \pi_k p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right) \right]}{\partial \mu_\mathbf{k}} \\
&= \sum_{n=1}^{N} \left[\frac{\pi_k p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right)}{\sum_{j=1}^{K} \pi_j p\left(\mathbf{x_n}|\mu_\mathbf{j},\mathbf{\Sigma_j}\right)} \left[\mathbf{\Sigma_k}^{-1} (\mathbf{x_n}-\mu_\mathbf{k}) \right] \right]
\end{aligned}$$

注意上式中 $\frac{\pi_k p(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k})}{\sum_{j=1}^{K} \pi_j p(\mathbf{x_n}|\mu_\mathbf{j},\mathbf{\Sigma_j})}$ 即为响应度 $\gamma(z_{nk})$ ，所以有：

$$\frac{\partial L}{\partial \mu_\mathbf{k}} = \sum_{n=1}^{N} \left[\gamma(z_{nk}) \mathbf{\Sigma_k}^{-1} (\mathbf{x_n}-\mu_\mathbf{k}) \right]$$

令 $\frac{\partial L}{\partial \mu_\mathbf{k}} = \mathbf{0}$ ，则 $\sum_{n=1}^{N} \left[\gamma(z_{nk}) \mathbf{\Sigma_k}^{-1} (\mathbf{x_n}-\mu_\mathbf{k}) \right] = \mathbf{0}$ ，

左乘 $\mathbf{\Sigma_k}$ ，整理得

$$\sum_{n=1}^{N}\left[\gamma(z_{nk})\mathbf{x_n}\right]=\sum_{n=1}^{N}\left[\gamma(z_{nk})\mu_{\mathbf{k}}\right]=\mu_{\mathbf{k}}\sum_{n=1}^{N}\gamma(z_{nk})$$

定义 $N_k=\sum_{n=1}^{N}\gamma(z_{nk})$ ，则

$$\mu_{\mathbf{k}}=\frac{1}{N_k}\sum_{n=1}^{N}\left[\gamma(z_{nk})\mathbf{x_n}\right]$$

N_k 可理解为被分配到第 k 个分模型(聚类)的“有效“的样本数。

对 $\mathbf{\Sigma_k}$ 中各元素求偏导：

$$\begin{aligned}\frac{\partial L}{\partial\mathbf{\Sigma_k}}&=\frac{\partial\ln p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma})}{\partial\mathbf{\Sigma_k}}\\&=\frac{\partial\left(\sum_{n=1}^{N}\ln\left[\sum_{k=1}^{K}\pi_k p\left(\mathbf{x_n}|\mu_{\mathbf{k}},\mathbf{\Sigma_k}\right)\right]\right)}{\partial\mathbf{\Sigma_k}}\\&=\sum_{n=1}^{N}\frac{\partial\ln\left[\sum_{k=1}^{K}\pi_k p\left(\mathbf{x_n}|\mu_{\mathbf{k}},\mathbf{\Sigma_k}\right)\right]}{\partial\mathbf{\Sigma_k}}\\&=\sum_{n=1}^{N}\left[\frac{\pi_k}{\sum_{j=1}^{K}\pi_j p\left(\mathbf{x_n}|\mu_{\mathbf{j}},\mathbf{\Sigma_j}\right)}\frac{\partial p\left(\mathbf{x_n}|\mu_{\mathbf{k}},\mathbf{\Sigma_k}\right)}{\partial\mathbf{\Sigma_k}}\right]\end{aligned}$$

接下来涉及矩阵求导(Matrix calculus - Wikipedia)，要复杂一些，这里不做推导，按参考文献[1]Chapter 9的公式(9.19)，给出 $\frac{\partial L}{\partial\mathbf{\Sigma_k}}=\mathbf{0}$ 的结果：

$$\mathbf{\Sigma_k}=\frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})(\mathbf{x_n}-\mu_{\mathbf{k}})(\mathbf{x_n}-\mu_{\mathbf{k}})^T$$

对 π_k 求偏导并令其为0：

$$\begin{aligned}\frac{\partial L}{\partial\pi_k}&=\frac{\partial\ln p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma})}{\partial\pi_k}\\&=\frac{\partial\left[\sum_{n=1}^{N}\ln\left[\sum_{k=1}^{K}\pi_k p\left(\mathbf{x_n}|\mu_{\mathbf{k}},\mathbf{\Sigma_k}\right)\right]+\lambda\left(\sum_{k=1}^{K}\pi_k-1\right)\right]}{\partial\pi_k}\\&=\sum_{n=1}^{N}\frac{p\left(\mathbf{x_n}|\mu_{\mathbf{k}},\mathbf{\Sigma_k}\right)}{\sum_{j=1}^{K}\pi_j p\left(\mathbf{x_n}|\mu_{\mathbf{j}},\mathbf{\Sigma_j}\right)}+\lambda=0\end{aligned}$$

注意到上式左右两边乘以 π_k 可凑出 $\gamma(z_{nk})$ ，所以有

$$\sum_{n=1}^{N}\frac{\pi_k p\left(\mathbf{x_n}|\mu_{\mathbf{k}},\mathbf{\Sigma_k}\right)}{\sum_{j=1}^{K}\pi_j p\left(\mathbf{x_n}|\mu_{\mathbf{j}},\mathbf{\Sigma_j}\right)}+\lambda\pi_k=\sum_{n=1}^{N}\gamma(z_{nk})+\lambda\pi_k\&=N_k+\lambda\pi_k=0$$

上式对 k 求和得

$$\sum_{k=1}^{K}(N_k+\lambda\pi_k)=\sum_{k=1}^{K}N_k+\lambda\sum_{k=1}^{K}\pi_k=\sum_{k=1}^{K}N_k+\lambda=0$$

又

$$\begin{aligned}\sum_{k=1}^{K}N_k&=\sum_{k=1}^{K}\sum_{n=1}^{N}\gamma(z_{nk})=\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\\&=\sum_{n=1}^{N}\sum_{k=1}^{K}\frac{\pi_k p\left(\mathbf{x}|\mu_{\mathbf{k}},\mathbf{\Sigma_k}\right)}{\sum_{j=1}^{K}\pi_j p\left(\mathbf{x}|\mu_{\mathbf{j}},\mathbf{\Sigma_j}\right)}\\&=\sum_{n=1}^{N}1=N\end{aligned}$$

所以 $\lambda=-N$ ，进而 $\pi_k=-\frac{N_k}{\lambda}=\frac{N_k}{N}$ 。

综上，对数似然函数 $\ln p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma})$ 的极值点满足的条件为：

$$\begin{aligned}\mu_{\mathbf{k}}&=\frac{1}{N_k}\sum_{n=1}^{N}\left[\gamma(z_{nk})\mathbf{x_n}\right]\mathbf{\Sigma_k}\\&=\frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})(\mathbf{x_n}-\mu_{\mathbf{k}})(\mathbf{x_n}-\mu_{\mathbf{k}})^T\pi_k\\&=\frac{N_k}{N}\end{aligned}\tag{6}$$

需要注意的是，上式并未给出高斯混合模型的解析解/闭式解(analytical soluiton/closed-form solution)，因为响应度$\gamma(z_{nk})$ 由式(4)给出，而参数 $\mu_{\mathbf{k}},\mathbf{\Sigma_k}$ 未知，故 $\gamma(z_{nk}),N_k$ 无法计算。

不过，根据(6)式可使用迭代算法来计算模型参数。

6. EM算法计算高斯混合模型

采用*EM算法*进行最大似然估计的快速计算。

6.1 高斯混合模型（Gaussian mixture model, GMM)

假设数据是由 K 个高斯概率密度函数混合而成，则高斯混合分布如下所示：

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

其中 π_k 表示第 k 个高斯密度函数在整个高斯混合分布中所占的比例，区间为 $[0,1]$ 且 $\sum_{k=1}^{K}\pi_k=1$ ， $N(X,\mu_k,\Sigma_k)$ 表示第 k 个高斯概率密度函数：

$$N(X,\mu_k,\Sigma_k)=\frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}}exp\{-\frac{1}{2}(X-\mu_k)^T\Sigma^{-1}(X-\mu_k)\}$$

高斯混合模型要做的事情，是通过一组训练数据 $\{X_i\}_{i=1}^n$ ，估计 K 个三元组参数$\{\pi_k, \mu_k, \Sigma_k\}$的值。我们即不知道每一个数据属于哪一个高斯密度函数，也不知道三元组参数的具体取值，那么如何进行求解呢?

这里就要搬出EM算法（Expectation-maximization），固定一个求另一个，不断循环迭代。

使用EM算法求解高斯混合模型的流程如下:

>> 输入： $\{X_i\}_{i=1}^n$
>> 输出：K 个三元组$\{\pi_k, \mu_k, \Sigma_k\}$

具体流程:
Step 1. 随机选取 k 个三元组$\{\pi_k, \mu_k, \Sigma_k\}$;

Step 2. 求 $\gamma_{ik} = \frac{\pi_k N(X_i, \mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k N(X_i, \mu_k, \Sigma_k)}$ ，表示第 i 个数据 X_i 属于第 k 个高斯密度函数的概率;

Step 3: 重新计算 k 个三元组$\{\pi_k, \mu_k, \Sigma_k\}$:

$$\pi_k = \frac{1}{n} \sum_{i=1}^n \gamma_{ik}$$

$$\mu_k = \frac{\sum_{i=1}^n \gamma_{ik} X_i}{\sum_{i=1}^n \gamma_{ik}}$$

$$\Sigma_k = \frac{\sum_{i=1}^n \gamma_{ik} (X_i - \mu_k)(X_i - \mu_k)^T}{\sum_{i=1}^n \gamma_{ik}}$$

> 相比高斯概率分布的 $\mu = \frac{1}{n} \sum_{i=1}^n X_i$ 和 $\Sigma = \frac{1}{n-1} \sum_{i=1}^n (X_i - \mu)(X_i - \mu)^T$，高斯混合模型用 γ_{ik} 对 K 个概率密度函数进行加权的归一化。

Step 4. 回到2循环直至收敛。

6.2 高斯混合模型的EM算法

不完全数据集 $\mathbf{X}$ 的对数似然函数
$\ln p(\mathbf{X}|\pi, \mu, \mathbf{\Sigma}) = \sum_{n=1}^N \ln \left[\sum_{k=1}^K \pi_k p\left(\mathbf{x_n}|\mu_\mathbf{k}, \mathbf{\Sigma_k}\right) \right]$ 的极大似然估计得到参数:

$$\begin{aligned} \mu_\mathbf{k} &= \frac{1}{N_k} \sum_{n=1}^N \left[\gamma(z_{nk}) \mathbf{x_n}\right] \\ \mathbf{\Sigma_k} &= \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x_n} - \mu_\mathbf{k})(\mathbf{x_n} - \mu_\mathbf{k})^T \\ \pi_k &= \frac{N_k}{N} \end{aligned} \tag{1}$$

其中，响应度 $\gamma(z_{nk}) = \frac{\pi_k p(\mathbf{x_n}|\mu_\mathbf{k}, \mathbf{\Sigma_k})}{\sum_{j=1}^K \pi_j p(\mathbf{x_n}|\mu_\mathbf{j}, \mathbf{\Sigma_j})}$ ， $N_k = \sum_{n=1}^N \gamma(z_{nk})$ 。

虽然(1)式不是闭式解，但是我们可以根据(1)式，通过迭代的方法来计算参数 $\mu_{\mathbf{k}}, \mathbf{\Sigma_k}, \pi_k$ 。

EM算法(Expectation-Maximization algorithm) 是就是这样一种用于计算含有隐变量的模型的极大似然解的强大方法。

下面简述用EM算法来计算高斯混合模型参数的步骤：

1. 初始化：给参数均值向量 $\mu_{\mathbf{k}}$ ，协方差矩阵 $\mathbf{\Sigma_k}$ 和混合系数 π_k 赋初值；并计算对数似然函数的初值；

* 注：可以用K均值聚类(K-means clustering)算法来得到初始参数。

2. **E步(Expectation step)**：用当前参数值 $\mu_{\mathbf{k}}, \mathbf{\Sigma_k}, \pi_k$ 计算后验概率(响应度) $\gamma(z_{nk})$ ：

$$\gamma(z_{nk}) \leftarrow \frac{\pi_k p\left(\mathbf{x_n}|\mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right)}{\sum_{j=1}^{K} \pi_j p\left(\mathbf{x_n}|\mu_{\mathbf{j}}, \mathbf{\Sigma_j}\right)}$$

3. **M步(Maximization step)**：用当前响应度 $\gamma(z_{nk})$ ，根据(1)式重新估计参数：

$$\begin{aligned}
\mu_{\mathbf{k}}^{\mathbf{new}} &\leftarrow \frac{1}{N_k} \sum_{n=1}^{N} \left[\gamma(z_{nk}) \mathbf{x_n}\right] \\
\mathbf{\Sigma}_{\mathbf{k}}^{\mathbf{new}} &\leftarrow \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x_n} - \mu_{\mathbf{k}}^{\mathbf{new}}) (\mathbf{x_n} - \mu_{\mathbf{k}}^{\mathbf{new}})^T \\
\pi_k^{new} &\leftarrow \frac{N_k}{N}
\end{aligned}$$

其中， $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$ 。

在M步中，先计算 $\mu_{\mathbf{k}}^{\mathbf{new}}$ 的值，然后用 $\mu_{\mathbf{k}}^{\mathbf{new}}$ 来计算新的协方差矩阵 $\mathbf{\Sigma}_{\mathbf{k}}^{\mathbf{new}}$ 。

4. 用新的参数 $\mu_{\mathbf{k}}^{\mathbf{new}}, \mathbf{\Sigma}_{\mathbf{k}}^{\mathbf{new}}, \pi_k^{new}$ 计算对数似然函数：

$$\ln p(\mathbf{X}|\pi, \mu, \mathbf{\Sigma}) = \sum_{n=1}^{N} \ln \left[\sum_{k=1}^{K} \pi_k p\left(\mathbf{x_n}|\mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right) \right]$$

然后检查是否收敛。(当然，也可直接检查参数是否收敛。)

如果收敛，则得到了模型参数；如果还没收敛，则返回第2步继续迭代。

参考文献[1], Section9.2, Page437指出，E步+M步可以保证对数似然函数的上升。不过需要注意的是，作为一种迭代算法，EM算法有可能收敛到局部最大值(local maximum)。

7. 换个角度看高斯混合模型的EM算法

在前面的极大似然估计中，我们的目标函数是不完全数据集 $\mathbf{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$ 的对数似然函数 $\ln p(\mathbf{X}|\pi, \mu, \mathbf{\Sigma})$ 。

现在我们换个角度来看：假如我们也有隐变量的观测值 $\mathbf{Z}=\{\mathbf{z_1},\ldots,\mathbf{z_N}\}$，即知道 $\mathbf{x_n}$ 来自高斯混合中的哪一个分模型，换句话说，我们有完全数据集 $\{\mathbf{X},\mathbf{Z}\}$，那么我们可以计算完全数据的对数似然函数 $\ln p(\mathbf{X},\mathbf{Z}|\pi,\mu,\mathbf{\Sigma})$。

首先，一组完全数据 $\{\mathbf{x_n},\mathbf{z_n}\}$ 的似然函数为：

$$p(\mathbf{x_n},\mathbf{z_n}|\pi,\mu,\mathbf{\Sigma})=\prod_{k=1}^{K}\pi_k^{z_{nk}}p(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k})^{z_{nk}}$$

所以，整个完全数据集 $\{\mathbf{X},\mathbf{Z}\}$ 的似然函数为：

$$p(\mathbf{X},\mathbf{Z}|\pi,\mu,\mathbf{\Sigma})=\prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}p(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k})^{z_{nk}}$$

(当然，我们假设每组数据相互独立。)

取对数得

$$\ln p(\mathbf{X},\mathbf{Z}|\pi,\mu,\mathbf{\Sigma})=\sum_{n=1}^{N}\sum_{k=1}^{K}\{z_{nk}\left[\ln\pi_k+\ln p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right)\right]\}\tag{2}$$

可以看出，和不完全数据的对数似然函数
$\ln p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma})=\sum_{n=1}^{N}\ln\left[\sum_{k=1}^{K}\pi_k p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right)\right]$相比，(2)式中 $\sum_{k=1}^{K}$ 和 $\ln$ 换了顺序。

然而实际情况是，我们并不知道隐变量的观测值，即(2)式中的 z_{nk}，所以我们无法直接计算 $\ln p(\mathbf{X},\mathbf{Z}|\pi,\mu,\mathbf{\Sigma})$。因此，我们考虑对 $\ln p(\mathbf{X},\mathbf{Z}|\pi,\mu,\mathbf{\Sigma})$ 在隐变量 z_{nk} 的后验分布下求期望，利用求期望的过程，将 z_{nk} 消掉。

$\ln p(\mathbf{X},\mathbf{Z}|\pi,\mu,\mathbf{\Sigma})$ 在隐变量 z_{nk} 的后验分布下的期望：

$$\begin{aligned}E_\mathbf{Z}\ln p(\mathbf{X},\mathbf{Z}|\pi,\mu,\mathbf{\Sigma})&=E\sum_{n=1}^{N}\sum_{k=1}^{K}\{z_{nk}\left[\ln\pi_k+\ln p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right)\right]\}\\&=\sum_{n=1}^{N}\sum_{k=1}^{K}\{E(z_{nk})\left[\ln\pi_k+\ln p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right)\right]\}\end{aligned}$$

其中，$E(z_{nk})$ 表示 z_{nk} 在后验分布下的期望 $E(z_{nk}|\mathbf{X},\theta)$。因 $z_{nk}\in\{0,1\}$，所以

$$\begin{aligned}E(z_{nk}|\mathbf{X},\theta)&=0\times p(z_{nk}=0|\mathbf{X},\theta)+1\times p(z_{nk}=1|\mathbf{X},\theta)\\&=p(z_{nk}=1|\mathbf{X},\theta)\\&=\gamma(z_{nk})\end{aligned}$$

也就是说，期望 $E(z_{nk})$ 就是响应度 $\gamma(z_{nk})$。进而，

$$E_\mathbf{Z}\ln p(\mathbf{X},\mathbf{Z}|\pi,\mu,\mathbf{\Sigma})=\sum_{n=1}^{N}\sum_{k=1}^{K}\{\gamma(z_{nk})\left[\ln\pi_k+\ln p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right)\right]\}$$

将上式简记为 $E_{\mathbf{Z}}$ ，现在将 $\gamma(z_{nk})$ 当作常数，求参数 $\pi, \mu, \mathbf{\Sigma}$ ，使 $E_{\mathbf{Z}}$ 最大化，即
$\max_{\pi,\mu,\mathbf{\Sigma}} E_{\mathbf{Z}}$
$s.t. \sum_{k=1}^{K} \pi_k = 1$

采用拉格朗日乘子法，拉格朗日函数为：

$$L_E = E + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

先对 $\mu_{\mathbf{k}}$ 求梯度，因为

$$\begin{aligned} \frac{\partial \ln p\left(\mathbf{x_n} | \mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right)}{\partial \mu_{\mathbf{k}}} &= \frac{1}{p\left(\mathbf{x_n} | \mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right)} p\left(\mathbf{x_n} | \mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right) \left[\mathbf{\Sigma^{-1}} (\mathbf{x_n} - \mu_{\mathbf{k}})\right] \\ &= \mathbf{\Sigma^{-1}} (\mathbf{x_n} - \mu_{\mathbf{k}}) \end{aligned}$$

所以，

$$\begin{aligned} \frac{\partial L_E}{\partial \mu_{\mathbf{k}}} &= \sum_{n=1}^{N} \left[\frac{\partial \sum_{k=1}^{K} \gamma(z_{nk}) \ln p\left(\mathbf{x_n} | \mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right)}{\partial \mu_{\mathbf{k}}} \right] \\ &= \sum_{n=1}^{N} \left[\gamma(z_{nk}) \frac{\partial \ln p\left(\mathbf{x_n} | \mu_{\mathbf{k}}, \mathbf{\Sigma_k}\right)}{\partial \mu_{\mathbf{k}}} \right] \\ &= \sum_{n=1}^{N} \gamma(z_{nk}) \left[\mathbf{\Sigma_k^{-1}} (\mathbf{x_n} - \mu_{\mathbf{k}})\right] \end{aligned}$$

进而，

$$\begin{aligned} \frac{\partial L_E}{\partial \mu_{\mathbf{k}}} = \mathbf{0} &\Rightarrow \sum_{n=1}^{N} \gamma(z_{nk}) \left[\mathbf{\Sigma_k^{-1}} (\mathbf{x_n} - \mu_{\mathbf{k}})\right] = \mathbf{0} \\ &\Rightarrow \sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x_n} - \mu_{\mathbf{k}}) = \mathbf{0} \\ &\Rightarrow \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x_n} = \sum_{n=1}^{N} \gamma(z_{nk}) \mu_{\mathbf{k}} \end{aligned}$$

同之前极大化 $\ln p(\mathbf{X} | \pi, \mu, \mathbf{\Sigma})$ 一样，可推得

$$\mu_{\mathbf{k}} = \frac{1}{N_k} \sum_{n=1}^{N} \left[\gamma(z_{nk}) \mathbf{x_n} \right]$$

然后是对 $\mathbf{\Sigma_k}$ 求梯度，

$$
\begin{aligned}
\frac{\partial L_E}{\partial \mathbf{\Sigma_k}} &= \sum_{n=1}^{N}\left[\frac{\partial \sum_{k=1}^{K}\gamma(z_{nk})\ln p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right)}{\partial \mathbf{\Sigma_k}}\right] \\
&= \sum_{n=1}^{N}\left[\gamma(z_{nk})\frac{\partial \ln p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right)}{\partial \mathbf{\Sigma_k}}\right] \\
&= \sum_{n=1}^{N}\left[\frac{\gamma(z_{nk})}{p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right)}\frac{\partial p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right)}{\partial \mathbf{\Sigma_k}}\right]
\end{aligned} \tag{3}
$$

回忆在文(一)中，我们求 $\ln p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma})$ 的极值点时，有

$$
\frac{\partial L}{\partial \mathbf{\Sigma_k}} = \sum_{n=1}^{N}\left[\frac{\pi_k}{\sum_{j=1}^{K}\pi_j p\left(\mathbf{x_n}|\mu_\mathbf{j},\mathbf{\Sigma_j}\right)}\frac{\partial p\left(\mathbf{x_n}|\mu_\mathbf{k},\mathbf{\Sigma_k}\right)}{\partial \mathbf{\Sigma_k}}\right] \tag{4}
$$

对比(3)(4)两式，结合响应度 $\gamma(z_{nk})$ 的公式，可发现两式是一致的，这意味着两种方法计算出的 $\mathbf{\Sigma_k}$ 也相同。

再来看 π_k ，

$$
\begin{aligned}
\frac{\partial L_E}{\partial \pi_k} &= \sum_{n=1}^{N}\left[\frac{\partial \sum_{k=1}^{K}\gamma(z_{nk})\ln \pi_k}{\partial \pi_k}\right] + \lambda \\
&= \sum_{n=1}^{N}\left[\gamma(z_{nk})\frac{\partial \ln \pi_k}{\partial \pi_k}\right] + \lambda \\
&= \sum_{n=1}^{N}\left[\gamma(z_{nk})\frac{1}{\pi_k}\right] + \lambda = \frac{1}{\pi_k}\sum_{n=1}^{N}\gamma(z_{nk}) + \lambda \\
&= \frac{1}{\pi_k}N_k + \lambda
\end{aligned}
$$

对比前文，可以发现 π_k 也和 $\ln p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma})$ 极值点的参数一致。

综上，通过极大化 $E_\mathbf{Z}\ln p(\mathbf{X},\mathbf{Z}|\pi,\mu,\mathbf{\Sigma})$ 得到的模型参数，和 $\ln p(\mathbf{X}|\pi,\mu,\mathbf{\Sigma})$ 的极大似然估计结果是一致的。在EM算法的每一次迭代中，我们都在对 $E_\mathbf{Z}$ 求极大值。

这给了我们另一个角度来看EM算法在高斯混合模型中的应用，也有助于我们理解EM算法的一般形式。

8. EM算法的一般形式

对于含有隐变量的概率模型，记变量和(离散的)隐变量分别为 $\mathbf{x},\mathbf{z}$ ，模型参数为θ ，(比如高斯混合模型的参数 $\theta=\{\pi,\mu,\mathbf{\Sigma}\}$)， $\mathbf{X}=\{\mathbf{x_1},\ldots,\mathbf{x_N}\}$ 为变量 $\mathbf{x}$ 的 N 个观测值的集合，即不完全数据集，$\mathbf{Z}=\{\mathbf{z_1},\ldots,\mathbf{z_N}\}$ 为相应的隐变量值的集合。

要计算 θ 极大化对数似然函数 $\ln p(\mathbf{X}|\theta)$ ，*EM算法*的思路是：考虑完全数据集 $\{\mathbf{X},\mathbf{Z}\}$ 的对数似然函数 $\ln p(\mathbf{X},\mathbf{Z}|\theta)$ 在隐变量的后验分布 $p(\mathbf{Z}|\mathbf{X},\theta^{\mathbf{old}})$ 下的期望，反复迭代极大化这个期望，最

终得到模型的参数值。其中 $\theta^{\mathbf{old}}$ 为当前参数。(若 $\mathbf{x_1},\ldots,\mathbf{x_N}$ 相互独立，则计算 $p(\mathbf{Z}|\mathbf{X},\theta^{\mathbf{old}})$ 实际上是计算 $p(\mathbf{z_n}|\mathbf{x_n},\theta^{\mathbf{old}}),n=1,\ldots,N$)

下面简述EM算法的一般形式：

1. 初始化模型参数 $\theta^{\mathbf{old}}$ ；

2. **E步(Expectation step)**：计算隐变量的后验分布 $p(\mathbf{Z}|\mathbf{X},\theta^{\mathbf{old}})$ ；

* 对于高斯混合模型，这一步实际上就是计算响应度 $\gamma(z_{nk})$ 。

3. **M步(Maximization step)**：极大化完全数据的对数似然函数 $\ln p(\mathbf{X},\mathbf{Z}|\theta)$ 在隐变量的后验分布 $p(\mathbf{Z}|\mathbf{X},\theta^{\mathbf{old}})$ 下的期望，即极大化函数:

$$Q\left(\theta,\theta^{\mathbf{old}}\right)=\sum_{\mathbf{Z}}p(\mathbf{Z}|\mathbf{X},\theta^{\mathbf{old}})\ln p(\mathbf{X},\mathbf{Z}|\theta)$$

得到相应的参数值 $\theta^{\mathbf{new}}=\arg\max_\theta Q\left(\theta,\theta^{\mathbf{old}}\right)$ 。

4. 检查对数似然函数或参数是否收敛，如果没收敛，则

$$\theta^{\mathbf{old}}\leftarrow\theta^{\mathbf{new}}$$

回到第2步继续迭代；

如果已收敛，则得到了模型参数。

EM算法的每次循环可以使不完全数据集的对数似然函数上升，除非其已收敛到局部最大值(参考文献[1], Section9.3, Page440)。

代码实现

此刻构建函数：

```
import numpy as np
import matplotlib.pyplot as plt

#高斯概率密度函数
def gaussian(x, Mean, Cov_matrix):
    dim = np.shape(cov)[0]  # 维度
    # 之所以加入单位矩阵是为了防止行列式为0的情况
    covdet = np.linalg.det(Cov_matrix + np.eye(dim) * 0.01)  # 协方差矩阵的行列式
    covinv = np.linalg.inv(Cov_matrix + np.eye(dim) * 0.01)  # 协方差矩阵的逆
    xdiff = x - Mean
    # 概率密度
    prob = 1.0 / np.power(2 * np.pi, 1.0 * dim / 2) / np.sqrt(np.abs(covdet)) *
np.exp(
```

```
        -0.5 * xdiff.T @ covinv @ xdiff)
    return prob

def GMM(X, K, iter_num=10):
    global  imgnum
    m, n = X.shape
    # print('m,n:',m,n)
    # print('K:',K)
    # 初始化参数
    oldpi = pi = np.full(K, 1.0 / K)
    oldmu = mu = [X[i] for i in np.roll(np.arange(K), np.random.choice(m))]
    olsigma = sigma = [np.eye(n) for i in range(K)]
    # print(oldpi)
    # print(oldmu)
    # print(olsigma)
    Q = np.zeros((m, K))
    errorlost = 0
    for itn in range(iter_num):
        plt.close()
        plt.plot(X[:, 0], X[:, 1], '.')
        # E_step
        for i in range(m):
            pxz = [pi[j] * gaussian(X[i], mu[j], sigma[j]) for j in range(K)]
            pxzsum = np.sum(pxz)
            Q[i] = pxz / pxzsum
        Qcol = np.sum(Q, axis=0)
        pi = Qcol / m
        mu = [np.sum(Q[:, j:j + 1] * X, axis=0) / Qcol[j] for j in range(K)]
        for j in range(K):
            xdif = X - mu[j]
            sigma[j] = 1.0 / Qcol[j] * np.sum(
                [Q[i][j] * (xdif[i:i + 1].T @ xdif[i:i + 1]) for i in range(m)],
axis=0)
        # 画出每个高斯分布的均值
        for j in range(K):
            plt.plot(mu[j][0], mu[j][1], marker='o', c='r')
        # plt.savefig(str(imgnum)+".png")
        imgnum += 1
        plt.show()
        curerror = sum(np.power((oldpi - pi).tolist(), 2)) +
sum(np.power(np.ravel(oldmu) - np.ravel(mu), 2)) + sum(np.power(np.ravel(olsigma) -
np.ravel(sigma), 2))
        # print(np.power((oldpi - pi).tolist(), 2))
        # print(np.power(np.ravel(oldmu) - np.ravel(mu), 2))
        # print(np.power(np.ravel(olsigma) - np.ravel(sigma), 2))
        if abs(curerror - errorlost) < 0.0001:#参数变化量小于0.0001时，就跳出循环
```

```
            break
        errorlost = curerror
        # print('mu:',mu)
        # print('sigma:',sigma)
        print("第{:d}次迭代与第一次的参数平方损失{:6f}: ".format(itn, errorlost))
    return pi, mu, sigma
```

主要的函数如下：

```
if __name__ == '__main__':
    # 初始化参数
    mean = [2, 2]
    cov = [[1, 0], [0, 1]]
    s1 = np.random.multivariate_normal(mean, cov, 80, "raise")
    mean = [10, 8]
    cov = [[1, 0], [0, 4]]
    s2 = np.random.multivariate_normal(mean, cov, 100, "raise")
    mean = [4, 13]
    cov = [[2, 1], [1, 3]]
    s3 = np.random.multivariate_normal(mean, cov, 120, "raise")

    sall = np.vstack((s1, s2, s3))
    np.random.shuffle(sall)
    # 高斯分量的个数
    K = 3
    imgnum = 0
    # 最大迭代次数
    iter_numbers = 60
    pi, mu, sigma = GMM(sall, K, iter_numbers)
    pointnum = 100
    xrange = np.linspace(sall[:, 0].min(), sall[:, 0].max(), pointnum)
    yrange = np.linspace(sall[:, 1].min(), sall[:, 1].max(), pointnum)
    XX, YY = np.meshgrid(xrange, yrange)
    plt.close()
    plt.scatter(s1[:, 0], s1[:, 1], marker='.', c='r')
    plt.scatter(s2[:, 0], s2[:, 1], marker='o', c='b')
    plt.scatter(s3[:, 0], s3[:, 1], marker='x', c='g')
    plt_data = np.dstack((XX, YY))
    for j in range(K):
        Z = [[gaussian(plt_data[q][p], mu[j], sigma[j]) for p in range(pointnum)]
for q in range(pointnum)]
        cs = plt.contour(XX, YY, Z)
        plt.clabel(cs)
    # plt.savefig(str(imgnum) + ".png")
    plt.show()
```

效果如下：