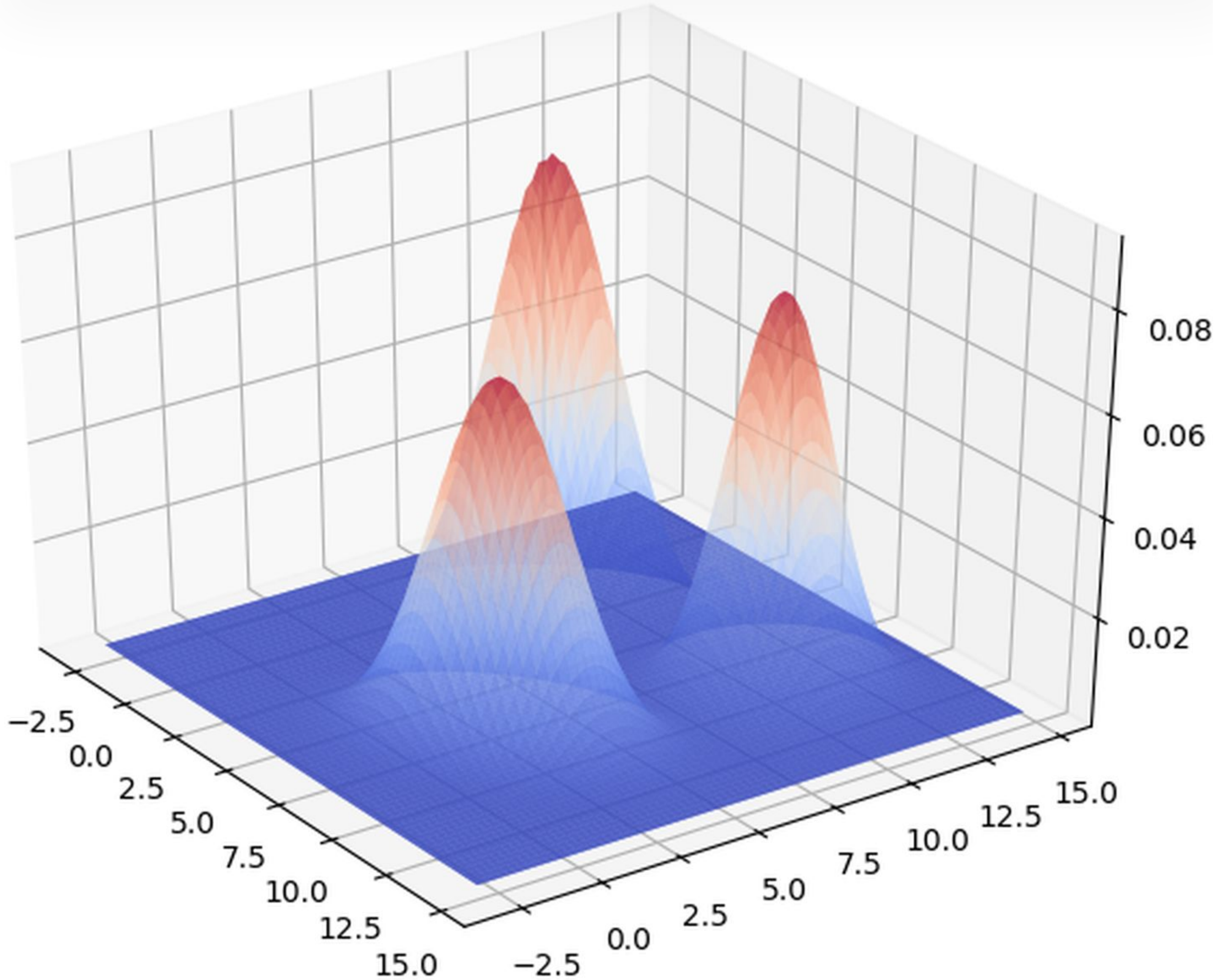


高斯混合模型（GMM）推导及实现

永远在你...
垃圾资乎，疯狂推新建的视频小号，这么急着赚钱，为了办后事吗

上一篇

永远在你身后：EM算法原理及推导
271 赞同 · 23 评论 文章

讨论的EM算法的推导过程，本问结合EM算法来推导高斯混合模型的原理，并对其进行简单的实现

而它与K-means相比的优点是，K-means只能将每个样本划分为一个类，GMM可以给出一个样本对于所有类别的概率。GMM不仅仅可以用于聚类，还可以用于概率密度的估计，也可以用于生成新的样本

GMM是一个生成模型，它假设数据是从多个高斯分布中生成的，可以这样理解生成流程：有 K 个高斯分布，赋予每一个分布一个权重，每当生成一个数据时，就按权重的比例随机选择一个分布，然后按照该分布生成数据

那么根据数据进行反推，可以假定一个样本有一个潜在的类别，而这个类别是无法观测到的，也就是隐变量，所以对于样本在给定参数 θ 的条件下边际概率为：

$$P(x|\theta)=\sum_{i}^{K}P(x,z=C_i|\theta)=\sum_{i}^{K}P(z=C_i|\theta)\,P(x|\theta,z=C_i)$$

其中 $z=C_i$ 表示样本属于某个类别， $P(z=C_i)$ 也就是隐变量的概率分布

z	C_1	C_2	...	C_K
P	p_1	p_2	...	p_K

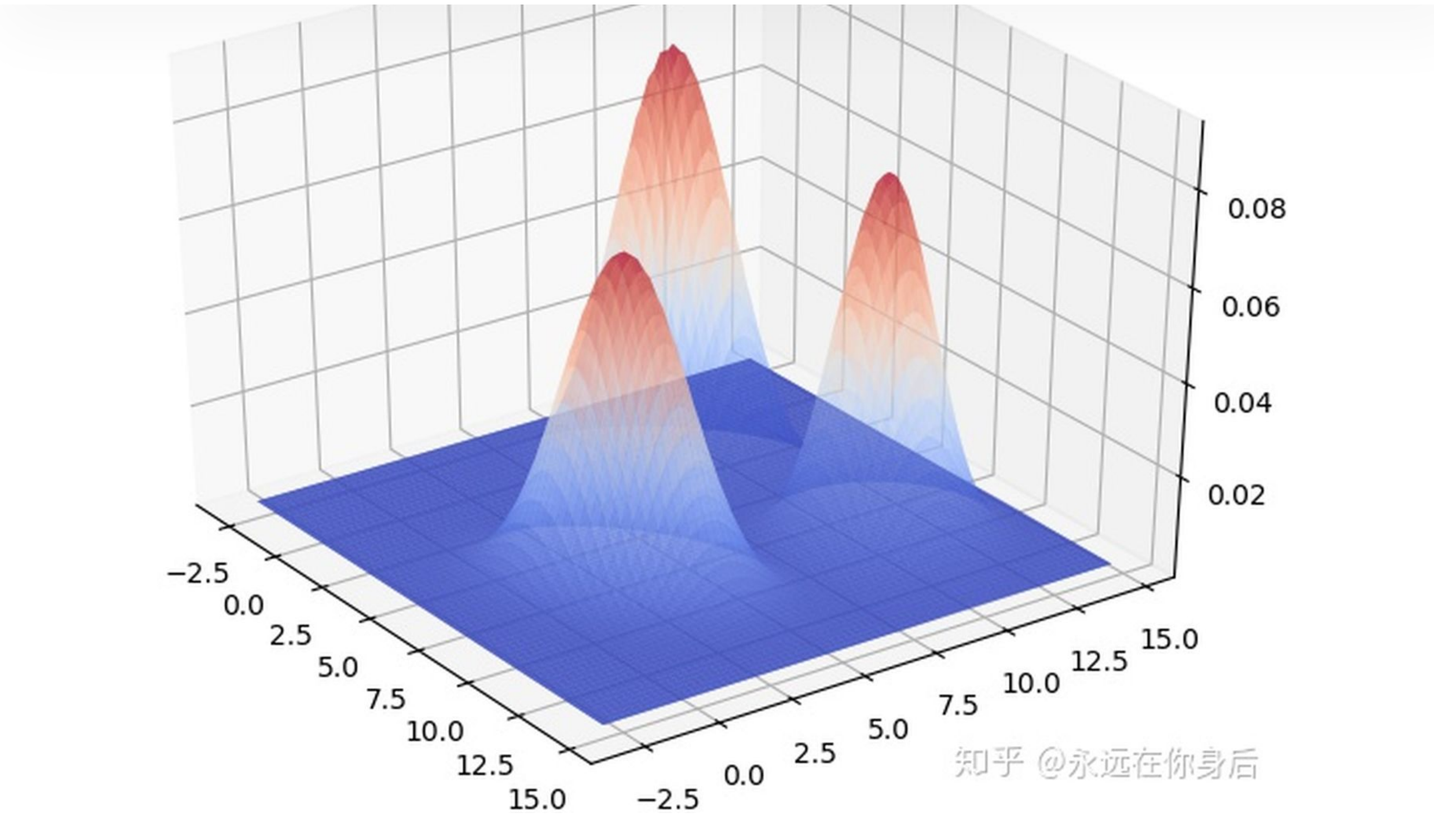
并且满足 $\sum_{i}^{K}p_i=1$ ，且样本的条件概率分布服从于（多元）高斯分布，即 $| = \sim (|, \Sigma)$ ：

$$\phi(x|\mu_i,\Sigma_i)=\frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_i|^{\frac{1}{2}}}\exp\left(-\frac{(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)}{2}\right)$$

上式假设样本含有 d 个属性。样本的边际概率为：

$$P(x|\theta)=\sum_{i}^{K}p_i\phi(x|\mu_i,\Sigma_i)$$

下面通过一个具体的栗子来对GMM进行推导，下图是三个二元高斯概率密度函数的图像



3个高斯

这三个高斯的参数分别是，均值：

$$\mu_1 = \begin{bmatrix} 2.5 \\ 8 \end{bmatrix}, \mu_2 = \begin{bmatrix} 8 \\ 2.5 \end{bmatrix}, \mu_3 = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$$

协方差矩阵：

$$\Sigma_1 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 3 & 2 \\ 1 & 2 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

下图是通过这三个高斯函数生成的数据

现在，我们通过这些数据，使用GMM将这三个高斯的参数给估计出来

首先，明确变量与参数

$$X=(x_1,x_2,\ldots,x_N)$$
$$Z=(z_1,z_2,\ldots,z_N)$$
$$\theta=(p,\mu,\Sigma)$$

其中参数 θ 包含隐变量 Z 的概率分布，各个高斯的均值与协方差矩阵：

$$p=(p_1,p_2,\ldots,p_K)$$
$$\mu=(\mu_1,\mu_2,\ldots,\mu_K)$$
$$\Sigma=(\Sigma_1,\Sigma_2,\ldots,\Sigma_K)$$

然后是EM算法中的E步，写出 Q 函数：

$$= \sum_{Z} \log P\left(X, Z|\theta\right) P\left(Z|X, \theta^{(t)}\right)$$

因为每个样本的独立同分布的，所以对数似然可以写成：

$$\log P\left(X, Z|\theta\right) = \log\left(\prod_{i=1}^{N} P\left(x_i, z_i|\theta\right)\right) = \sum_{i=1}^{N} \log P\left(x_i, z_i|\theta\right)$$

以及 Z 的后验可以写成

$$P\left(Z|X, \theta^{(t)}\right) = \prod_{i=1}^{N} P\left(z_i|x_i, \theta^{(t)}\right)$$

不过为了更加简洁，后验 $P\left(Z|X, \theta^{(t)}\right)$ 在下面的推导中先不展开

$$\begin{aligned}
&Q\left(\theta, \theta^{(t)}\right)\\
&= \sum_{Z}\left[\left(\sum_{i=1}^{N} \log P\left(x_i, z_i|\theta\right)\right) P\left(Z|X, \theta^{(t)}\right)\right]\\
&= \sum_{Z}\left[\log P\left(x_1, z_1|\theta\right) P\left(Z|X, \theta^{(t)}\right) + \ldots + \log P\left(x_N, z_N|\theta\right) P\left(Z|X, \theta^{(t)}\right)\right]\\
&= {\color{red}\sum_{Z}\left(\log P\left(x_1, z_1|\theta\right) P\left(Z|X, \theta^{(t)}\right)\right)} + \ldots + \sum_{Z}\left(\log P\left(x_N, z_N|\theta\right) P\left(Z|X, \theta^{(t)}\right)\right)
\end{aligned}$$

可以看到， Q 函数被分出了 N 个相似的部分，我们将其中标红部分提出来化简

$$\begin{aligned}
&\sum_{Z}\left(\log P\left(x_1, z_1|\theta\right) P\left(Z|X, \theta^{(t)}\right)\right)\\
&= \sum_{z_1, z_2, \ldots, z_N}\left(\log P\left(x_1, z_1|\theta\right)\left(\prod_{i=1}^{N} P\left(z_i|x_i, \theta^{(t)}\right)\right)\right)\\
&= \sum_{z_1, z_2, \ldots, z_N} \log P\left(x_1, z_1|\theta\right) P\left(z_1|x_1, \theta^{(t)}\right)\left(\prod_{i=2}^{N} P\left(z_i|x_i, \theta^{(t)}\right)\right)\\
&= \sum_{z_1} \sum_{z_2, \ldots, z_N} \log P\left(x_1, z_1|\theta\right) P\left(z_1|x_1, \theta^{(t)}\right)\left(\prod_{i=2}^{N} P\left(z_i|x_i, \theta^{(t)}\right)\right)\\
&= \sum_{z_1} \log P\left(x_1, z_1|\theta\right) P\left(z_1|x_1, \theta^{(t)}\right) {\color{blue}\sum_{z_2, \ldots, z_N}\left(\prod_{i=2}^{N} P\left(z_i|x_i, \theta^{(t)}\right)\right)}
\end{aligned}$$

再把上式中蓝色部分提出来展开

$$
\begin{aligned}
&= \sum_{z_2}\sum_{z_3}\cdots\sum_{z_N}\left(P\left(z_2|x_2,\theta^{(t)}\right)P\left(z_3|x_3,\theta^{(t)}\right)\ldots P\left(z_N|x_N,\theta^{(t)}\right)\right) \\
&= \sum_{z_2} P\left(z_2|x_2,\theta^{(t)}\right)\sum_{z_3} P\left(z_3|x_3,\theta^{(t)}\right)\ldots\sum_{z_N} P\left(z_N|x_N,\theta^{(t)}\right)
\end{aligned}
$$

可以看到， $P\left(z_i|x_i,\theta^{(t)}\right)$ 是关于 z_i 的概率分布，根据概率分布的性质有

$$
\sum_{z_i} P\left(z_i|x_i,\theta^{(t)}\right) = 1
$$

所以，整个这一（蓝色）部分的值等于1，代回前面的式子，得到

$$
\sum_{Z}\left(\log P(x_1,z_1|\theta)P\left(Z|X,\theta^{(t)}\right)\right) = \sum_{z_1}\log P(x_1,z_1|\theta)P\left(z_1|x_1,\theta^{(t)}\right)
$$

同理，可以得到

$$
\begin{aligned}
&\sum_{Z}\left(\log P(x_2,z_2|\theta)P\left(Z|X,\theta^{(t)}\right)\right) = \sum_{z_2}\log P(x_2,z_2|\theta)P\left(z_2|x_2,\theta^{(t)}\right) \\
&\vdots \\
&\sum_{Z}\left(\log P(x_N,z_N|\theta)P\left(Z|X,\theta^{(t)}\right)\right) = \sum_{z_N}\log P(x_N,z_N|\theta)P\left(z_N|x_N,\theta^{(t)}\right)
\end{aligned}
$$

将这些化简的结果代回到 Q 函数，可以得到

$$
\begin{aligned}
&Q\left(\theta,\theta^{(t)}\right) \\
&= \sum_{z_1}\log P(x_1,z_1|\theta)P\left(z_1|x_1,\theta^{(t)}\right) + \ldots + \sum_{z_N}\log P(x_N,z_N|\theta)P\left(z_N|x_N,\theta^{(t)}\right) \\
&= \sum_{i=1}^{N}\sum_{z_i}\log P(x_i,z_i|\theta)P\left(z_i|x_i,\theta^{(t)}\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K}\log P(x_i,z_i=C_j|\theta)P\left(z_i=C_j|x_i,\theta^{(t)}\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K}\log\left(p_k\phi(x_i|\mu_k,\Sigma_k)\right)P\left(z_i=C_k|x_i,\theta^{(t)}\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K}\left(\log p_k+\log\phi(x_i|\mu_k,\Sigma_k)\right)P\left(z_i=C_k|x_i,\theta^{(t)}\right)
\end{aligned}
$$

需要注意的是， Q 函数的两个参数中， $\theta^{(t)}$ 是当前时刻的参数值，是已经确定的，该参数所对应的部分： $P\left(z_i=C_k|x_i,\theta^{(t)}\right)$ ，是隐变量在当前参数下的后验概率，也是已经确定的，所以在后续的化简和求导中都可以当作常量处理

为了后面的推导更加简洁，先给该后验另取一个变量名 γ 作为表示：

$$
\begin{aligned}
\gamma_{ij} &= P\left(z_i=C_j|x_i,\theta^{(t)}\right) \\
&= \frac{P\left(x_i,z_i=C_j|\theta^{(t)}\right)}{\sum_{k=1}^{K}P\left(x_i,z_i=C_k|\theta^{(t)}\right)} \\
&= \frac{P\left(x_i|z_i=C_j,\theta^{(t)}\right)P\left(z_i=C_j|\theta^{(t)}\right)}{\sum_{k=1}^{K}P\left(x_i|z_i=C_k,\theta^{(t)}\right)P\left(z_i=C_k|\theta^{(t)}\right)} \\
&= \frac{p_j\phi\left(x_i|\mu_j,\Sigma_j\right)}{\sum_{k=1}^{K}p_k\phi\left(x_i|\mu_k,\Sigma_k\right)}
\end{aligned}
$$

下面，就是EM算法中的M步，根据确定的 Q 函数求下一时刻的参数，首先是 Z 的概率分布：

$$
\begin{aligned}
p^{(t+1)} &= \arg\max_{p}\sum_{i=1}^{N}\sum_{k=1}^{K}\left(\log p_k+\log\phi\left(x_i|\mu_k,\Sigma_k\right)\right)\gamma_{ik} \\
&= \arg\max_{p}\sum_{i=1}^{N}\sum_{k=1}^{K}\log p_k\gamma_{ik}
\end{aligned}
$$

不过，这里不能直接求导，因为概率分布有一个约束： $\sum_{i}^{K}p_i=1$ 。所以，通过拉格朗日乘数法来消除该约束。构造一个拉格朗日函数：

$$
L(p,\lambda)=\sum_{i=1}^{N}\sum_{k=1}^{K}\log p_k\gamma_{ik}+\lambda\left(\sum_{i}^{K}p_i-1\right)
$$

对该函数求导

$$
\frac{\partial L\left(p_k,\lambda\right)}{\partial p_j}=\sum_{i=1}^{N}\frac{1}{p_j}\gamma_{ij}+\lambda
$$

令偏导为0，并且两边同时乘上 p_j ，得到：

N

$$\frac{\partial L\left(p_k, \lambda\right)}{\partial p_1} \Longrightarrow \sum_{i=1}^{N} \gamma_{i1} = -\lambda\, p_1$$

$$\vdots$$

$$\frac{\partial L\left(p_k, \lambda\right)}{\partial p_K} \Longrightarrow \sum_{i=1}^{N} \gamma_{iK} = -\lambda\, p_K$$

将所有分量的结果相加，得到：

$$\sum_{j=1}^{K}\sum_{i=1}^{N} \gamma_{ij} = \sum_{j=1}^{K} -\lambda p_j$$

首先，利用之前的约束条件： $\sum_{i}^{K} p_i = 1$ ，所以上式右边结果为 $-\lambda$ 。又因为 γ_{ij} 也是一个概率分布，所以有： $\sum_{j}^{K} \gamma_{ij} = 1$ ，最终得到：

$$\lambda = -N$$

代入（1.1）式，得到：

$$p_j^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N} \gamma_{ij}$$

然后是各个高斯的均值：

$$\begin{aligned}
\mu^{(t+1)} &= \arg\max_{\mu} \sum_{i=1}^{N}\sum_{k=1}^{K} \left(\log p_k + \log \phi\left(x_i | \mu_k, \Sigma_k\right)\right) \gamma_{ik} \\
&= \arg\max_{\mu} \sum_{i=1}^{N}\sum_{k=1}^{K} \log \phi\left(x_i | \mu_k, \Sigma_k\right) \gamma_{ik} \\
&= \arg\max_{\mu} \sum_{i=1}^{N}\sum_{k=1}^{K} \left(\log \frac{1}{(2\pi)^{\frac{d}{2}}} - \frac{1}{2}\log\left|\Sigma_k\right| - \frac{\left(x_i - \mu_k\right)^T \Sigma_k^{-1}\left(x_i - \mu_k\right)}{2}\right) \gamma_{ik} \\
&= \arg\max_{\mu} \sum_{i=1}^{N}\sum_{k=1}^{K} \left(-\frac{\left(x_i - \mu_k\right)^T \Sigma_k^{-1}\left(x_i - \mu_k\right)}{2}\right) \gamma_{ik}
\end{aligned}$$

这里没有约束条件，直接对上式求导

$$
\begin{aligned}
&= \frac{\partial}{\partial \mu_j}\left(\sum_{i=1}^{N}\left(-\frac{(x_i-\mu_j)^T\Sigma_j^{-1}(x_i-\mu_j)}{2}\right)\gamma_{ij}\right) \\
&= \sum_{i=1}^{N}\left(-\frac{1}{2}\frac{\partial}{\partial \mu_j}\left((x_i-\mu_j)^T\Sigma_j^{-1}(x_i-\mu_j)\right)\right)\gamma_{ij} \\
&= \sum_{i=1}^{N}\Sigma_j^{-1}(x_i-\mu_j)\gamma_{ij} \\
&= \sum_{i=1}^{N}\Sigma_j^{-1}x_i\gamma_{ij} - \sum_{i=1}^{N}\Sigma_j^{-1}\mu_j\gamma_{ij}
\end{aligned}
$$

令偏导为0，并且两端同时乘上 Σ_j^2 ：

$$\sum_{i=1}^{N} x_i\gamma_{ij} = \mu_j\sum_{i=1}^{N}\gamma_{ij}$$

最后得到：

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} x_i\gamma_{ij}}{\sum_{i=1}^{N}\gamma_{ij}}$$

最后，就是协方差矩阵的更新了，首先化简 Q 函数

$$
\begin{aligned}
\Sigma^{(t+1)} &= \arg\max_{\Sigma}\sum_{i=1}^{N}\sum_{k=1}^{K}\left(\log\frac{1}{(2\pi)^{\frac{d}{2}}} - \frac{1}{2}\log|\Sigma_k| - \frac{(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)}{2}\right)\gamma_{ik} \\
&= \arg\min_{\Sigma}\sum_{i=1}^{N}\sum_{k=1}^{K}\left(\log|\Sigma_k| + (x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right)\gamma_{ik}
\end{aligned}
$$

这里，先把特征值的求导公式贴出来，对于一个矩阵 A ，有：

$$
\begin{aligned}
\frac{\partial|A|}{\partial A} &= |A|\,A^{-1} \\
\frac{\partial\log|A|}{\partial A} &= A^{-1}
\end{aligned}
$$

接着，求 Q 函数关于协方差的偏导：

$$
\begin{aligned}
&= \sum_{i=1}^{N} \frac{\partial}{\partial \Sigma_j} \left(\log |\Sigma_j| + (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j) \right) \gamma_{ij} \\
&= \sum_{i=1}^{N} \left(\Sigma_j^{-1} - (x_i - \mu_j)(x_i - \mu_j)^T \Sigma_j^{-2} \right) \gamma_{ij}
\end{aligned}
$$

下面对上式中标红部分的求导过程进行展开，因为 $(x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j)$ 的结果是一个标量，也可以视作是一个 1×1 的矩阵，所以它就迹就等于它自身。又因为

$$() = ()$$

所以，变换得到

$$
\begin{aligned}
(x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j) &= tr\left((x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j) \right) \\
&= tr\left(\Sigma_j^{-1} (x_i - \mu_j)(x_i - \mu_j)^T \right)
\end{aligned}
$$

同样，矩阵的迹求导公式为：

$$\frac{\partial tr(AB)}{\partial A} = B^T$$

所以

$$
\begin{aligned}
\frac{\partial tr\left(\Sigma_j^{-1} (x_i - \mu_j)(x_i - \mu_j)^T \right)}{\partial \Sigma_j} &= (x_i - \mu_j)(x_i - \mu_j)^T \frac{\partial \Sigma_j^{-1}}{\partial \Sigma_j} \\
&= -(x_i - \mu_j)(x_i - \mu_j)^T \Sigma_j^{-2}
\end{aligned}
$$

接着，令偏导为0，并且两端同时乘上 Σ_j^2 ：

$$
\begin{aligned}
&\sum_{i=1}^{N} \left(\Sigma_j - (x_i - \mu_j)(x_i - \mu_j)^T \right) \gamma_{ij} = 0 \\
&\Sigma_j \sum_{i=1}^{N} \gamma_{ij} = \sum_{i=1}^{N} (x_i - \mu_j)(x_i - \mu_j)^T \gamma_{ij}
\end{aligned}
$$

最后得到：

$$-^{(t+1)} \quad \sum_{i=1}^{N} (x_i - \mu_j)^T (x_i - \mu_j) \gamma_{ij}$$

以上，全部的公式已经推导完毕，现在整理一下

$$
\begin{aligned}
\gamma_{ij} &= \frac{p_j \phi\left(x_i | \mu_j, \Sigma_j\right)}{\sum_{k=1}^{K} p_k \phi\left(x_i | \mu_k, \Sigma_k\right)} \\
p_j^{(t+1)} &= \frac{1}{N} \sum_{i=1}^{N} \gamma_{ij} \\
\mu_j^{(t+1)} &= \frac{\sum_{i=1}^{N} x_i \gamma_{ij}}{\sum_{i=1}^{N} \gamma_{ij}} \\
\mu_j^{(t+1)} &= \frac{\sum_{i=1}^{N} x_i \gamma_{ij}}{\sum_{i=1}^{N} \gamma_{ij}}
\end{aligned}
$$

最后的最后，根据以上公式，写一个简单的实现

```python
import numpy as np
from scipy import stats

class GMM(object):
    def __init__(self, k: int, d: int):
        '''
        k: K值
        d: 样本属性的数量
        '''
        self.K = k
        # 初始化参数
        self.p = np.random.rand(k)
        self.p = self.p / self.p.sum()      # 保证所有p_k的和为1
        self.means = np.random.rand(k, d)
        self.covs = np.empty((k, d, d))
        for i in range(k):                  # 随机生成协方差矩阵，必须是半正定矩阵
            self.covs[i] = np.eye(d) * np.random.rand(1) * k

    def fit(self, data: np.ndarray):
        '''
        data: 数据矩阵，每一行是一个样本，shape = (N, d)
        '''
        for _ in range(100):
            density = np.empty((len(data), self.K))
            for i in range(self.K):
```

```
            posterior = density * self.p
            posterior = posterior / posterior.sum(axis=1, keepdims=True)
            # 计算下一时刻的参数值
            p_hat = posterior.sum(axis=0)
            mean_hat = np.tensordot(posterior, data, axes=[0, 0])
            # 计算协方差
            cov_hat = np.empty(self.covs.shape)
            for i in range(self.K):
                tmp = data - self.means[i]
                cov_hat[i] = np.dot(tmp.T*posterior[:,i], tmp) / p_hat[i]
            # 更新参数
            self.covs = cov_hat
            self.means = mean_hat / p_hat.reshape(-1,1)
            self.p = p_hat / len(data)

        print(self.p)
        print(self.means)
        print(self.covs)
```

随机生成了2000个样本，迭代100次后的结果如下：

```
p = np.array([0.3, 0.6, 0.1])
means = np.array([
    [2.5,8],
    [8,2.5],
    [10,10]
])
covs = np.array([
    [[2,1],[1,2]],
    [[3,2],[1,2]],
    [[2,0],[0,2]]
])

# 隐变量概率分布
[0.29775162 0.59836025 0.10388813]
# 均值
[[ 2.48575471  8.22122078]
 [ 7.95854299  2.46035662]
 [10.03840073  9.95829754]]
# 协方差
```

```
 [ 1.42297665  2.21976053]]

[[ 2.07091948 -0.03055896]
 [-0.03055896  2.07287524]]]
```

可以看到，基本上已经收敛的差不多了

编辑于 2019-10-07 14:22

机器学习　自然语言处理　人工智能

文章被以下专栏收录

技术部落联盟
涉及容器、大数据、人工智能、物联网等领域

一个大学生的日常笔记
你能找到的最通俗易懂的大学生数学与编程知识分享

推荐阅读

高斯混合模型(GMM)的两种详解及简化
磊磊

机器学习-白板推导系列(十一)高斯混合模型GMM（Gaussi...
龘龘丶

写下你的评论...

枫雪镜夜 2020-12-04
一定是看了B站的机器学习下白板推导

4

---

lucky 2020-03-26
cj是不是应该改成ck?

1

水库浪子 回复 **lucky** 2020-11-29
我觉得要改，应该是写混了

赞

六毛吧 2019-12-10
如果我是多维数据怎么办

1

学弱猹 2019-11-21
这篇文章可以投到我的专栏内嘛

1

永远在你身后 (作者) 回复 **学弱猹** 2019-11-22
恩，注明来源即可

1

学弱猹 回复 **永远在你身后** (作者) 2019-11-22
投稿需要宁帮忙投进来，我好像才能接收

赞

展开其他 2 条回复

信息门下溜狗 2020-12-17

warrior

这篇推导简直牛逼

赞

海川 11-06

从群里特地前来拜读

赞

ngc111 2020-06-01

您好可以转载一下吗，会标注好来源

赞

聃珠 2019-12-09

棒！学到很多

赞

ECHO 08-10

赞一个，白板推导没有期望和方差的EM，这篇文章都补齐了

赞

Haha 07-12

请问多维数据比如1000个变量，如何根据gmm聚类？

赞

angus zou 03-11

赞

三弦 03-04

博主你好 我想问一下gmm需要对训练数据进行标准化吗？

赞

我是狮子啊 01-15

为啥P(x|\theta) 称之为 边际概率？写成这个形式不应该是条件概率么？

赞

已赞同 384 ▼ 26 条评论 分享 喜欢 收藏 申请转载 ···

赞

我是狮子啊 回复 Reloaded 05-24

应该是笔误

赞

念念不忘必有回响 01-13

GMM的参数初始化，可以用kmeans来做吗

赞

Nights 2020-10-15

你好你的2000个数据是如何生成的，是按照GMM给定的参数生成的么

赞

喜欢一只雪梨 回复 Nights 04-02

请问你现在有这个的初始数据吗

赞

慕橙cC 2020-10-07

你好，先给你点赞！太强啦！另外想问下是否可以证明k-means是GMM在簇先验服从均匀分布、服从相等球形协方差矩阵时的特例呢?

赞

罗凡 2020-08-07

碰到好多次，后面写r还有写 diag的，这是什么意思

赞