高斯混合模型与EM算法的数学原理及应用实例

SIGAI特约作者
张凌寒

微信扫一扫
关注公众号

摘要

GMM(Gaussian Mixture Model, 高斯混合模型)被誉为万能分布近似器, 其拥有强悍的数据建模能力. GMM使用若干个高斯分布的加权和作为对观测数据集进行建模的基础分布, 而由中心极限定理我们知道, 大量独立同分布的随机变量的均值在做适当标准化之后会依分布收敛于高斯分布, 这使得高斯分布具有普适性的建模能力, 继而奠定了使用高斯分布作为主要构成部件的GMM进行数据建模的理论基础. GMM是典型的概率图模型 ^{[8][9]} , GMM与其变种k-means(k均值)算法都是工业实践中经常使用的聚类工具. 由于GMM在建模时引入了隐变量的概念, 致使我们无法直接使用MLE(Maximum Likelihood Estimate, 极大似然估计)进行参数估计, 进而引入了EM(Expectation Maximization algorithm, 最大期望算法)算法来对含有隐变量的模型进行训练. EM算法通过迭代地构造似然函数下限的方式不断地提升似然函数的取值, 从而完成对含有隐变量模型的参数估计, 其典型的应用包括GMM、HMM(Hidden Markov Model, 隐马尔可夫模型) ^{[11]} 的参数估计. 本文将从一个问题实例出发, 引出使用GMM和EM算法对问题进行解决的思路, 阐述其背后的数学原理, 最后给出完整的解决方案. 本文组织如下:

阐述一个不完全数据的问题实例;
使用GMM模型对不完全数据的分布进行建模;
使用EM算法对带隐变量的模型进行参数估计;
使用EM算法对GMM模型进行求解的具体过程;
求解不完全数据问题实例的概率分布;
阐述k-means算法与GMM模型的关系;

总结.

关键词: 高斯混合模型, EM算法, 概率图模型, 机器学习

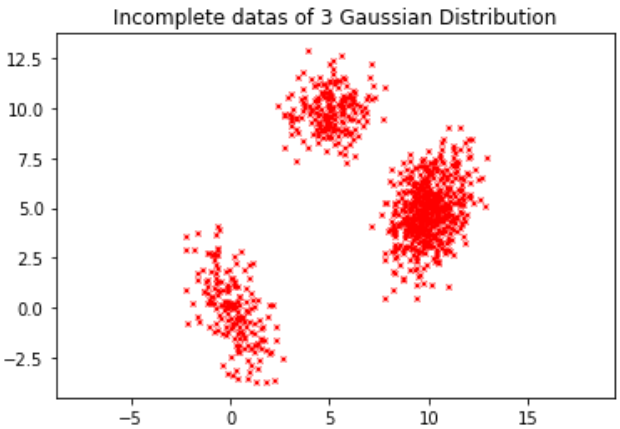
不完全数据的问题实例

n是样本个数

假设我们有数据集

$$\mathcal{D} = \{x_i\}_{i=1}^{n}$$

, 数据集 $\mathcal{D}$ 中的每个样本 $x_i, i = 1, \cdots,$ n 分别是从 K 个高斯分布[1][7]中的某一个采样得到的, 但具体是哪一个高斯分布我们不得而知, 即我们无法观察采样的过程. 如下图所示, 图中含有1000个样本点, 每个样本点是从3个高斯分布中的某一个进行采样得到的. 由于我们无法观测到采样的具体过程, 所以某一个样本点 $x_i, i = 1, \cdots, n$ 归属于哪个高斯分布我们并不知晓, 故在图中所有样本点均使用同一颜色进行表示. 如果我们现在想要对产生数据的分布进行建模, 估计每个高斯分布的参数, 并对每个点属于哪一个高斯分布进行预测, 我们应该如何操作呢? 为了解决这个问题, 我们需要引进一些额外的变量.



三个高斯分布采样得到的数据集

为了能清晰地描述数据集 $\mathcal{D}$ 的生成过程, 我们引入一个随机变量 Z 来进行辅助, 这个随机变量 Z 服从概率分布 $\mathcal{Q}$, 其取值为 $k \in 1, \cdots, K$, 但分布 $\mathcal{Q}$ 的具体参数我们并不知晓, 即概率 $\alpha_k = \mathcal{Q}(Z = k), k \in 1, \cdots, K$ 是未知的. 有了随机变量 Z 的辅助, 数据集 $\mathcal{D}$ 的生成便可分为两个步骤: 首先, 我们从分布 $Z \sim \mathcal{Q}$ 采样得到一个值 $z_i = k, k \in 1, \cdots, K$ 用来确定样本 x_i 将从哪一个高斯分布进行采样产生; 然后, 我们对第 k 个高斯分布 $\mathcal{N}(x|Z = k; \vec{u}_k, \Sigma_k)$ 进行采样, 从而产生我们的样本 $x_i, i = 1, \cdots, n,$ 其中均值向量 $\vec{u}_k$, 协方差矩阵 Σ_k 是第 k 个高斯分布的待估参数.

因为我们无法直接观察到数据集 $\mathcal{D}$ 的整个产生过程, 观测值只有 $x < em > i, i = 1, \cdots, n,$ 而无法得知每个样本具体从哪一个高斯分布采样得到, 即 $< spanclass = "editormd - tex" > z_i, i = 1, \cdots, n < /span >$ 的取值无从知晓, 所以我们无法轻易地获得分布 $< spanclass = "editormd - tex" > \mathcal{Q} < /span >$ 和分布 $< spanclass = "editormd - tex" > \mathcal{N}(x|Z = k; \vec{u}_k, \Sigma_k), k = 1, \cdots, K < /span >$ 的参数估计值.我们将 $< spanclass = "editormd - tex" > (x_i, z_i) < /span >$ $i = 1^n$ 称为完全数据, 而将 $x_{i i=1}^{n}$ 称为不完全数据. 由于随机变量 Z 不可被直接观察, 所以我们称之为隐变量. 对于这种含有隐变量的不完全数据, 我们该如何来对其分布进行建模呢? 答案便是GMM模型.

把隐变量考虑进去的叫完全数据
不把隐变量考虑进去的叫不完全数据

GMM模型对不完全数据的分布进行建模

GMM模型使用 K 个高斯分布的加权和作为其概率密度函数, 具体地

$$p(x; \Theta) = \sum_{k=1}^{K} \alpha_k \cdot \mathcal{N}(x; \vec{u}_k, \Sigma_k) \quad (1)$$

(1)中, 模型的参数

$$\Theta = \{\alpha_k, \vec{u}_k, \Sigma_k\}_{k=1}^{K}$$

, 为了保证概率密度函数在区间 $(-\infty, +\infty)$ 上的积分为1, 我们有

$$\int_{-\infty}^{+\infty} p(x; \Theta) dx = \sum_{k=1}^{K} \left[\alpha_k \int_{-\infty}^{+\infty} \mathcal{N}(x; \vec{u}_k, \Sigma_k) \, dx \right] = \sum_{k=1}^{K} \alpha_k = 1 \quad (2)$$

样本来自第k个高斯的概率

所以对于(1), 参数 $\alpha_k, k = 1, \cdots, K$ 需要满足 $\sum_{k=1}^{K} \alpha_k = 1,$ 此处参数 $\alpha_k, k = 1, \cdots, K$ 即为隐变量 Z 所服从的分布 $\mathcal{Q}$ 的参数值, 即 $\alpha_k = \mathcal{Q}(Z = k), k \in 1, \cdots, K$. 由(1)可以看出, GMM模型对不完全数据分布的建模是通过求其边缘分布 $p(x; \Theta) = \sum < em > zp(x, z; \Theta) = \sum < /em > zp(z; \Theta)p(x|z; \Theta)$ 得到的.

假如我们能观察到数据集 $\mathcal{D}$ 的整个产生过程, 即样本 x_i 是由第 z_i 个高斯分布采样得到是可以观察到的, 此时观测值为完全数据

$$\{(x_i, z_i)\}_{i=1}^{n}$$

那我们就能很轻松地对分布 $\mathcal{Q}$ 和分布 $\mathcal{N}(x|Z = k; \vec{u}_k, \Sigma_k), k = 1, \cdots, K$ 进行参数估计了, 具体地, 对于分布 $\mathcal{N}(x|Z = k; \vec{u}_k, \Sigma_k)$

$$\vec{u}_k^{\star}, \Sigma_k^{*} = \underset{\vec{u}_k, \Sigma_k}{\operatorname{argmin}} - \sum_{i \in \{i|z_i = k\}}^{n} \log \mathcal{N}(x_i|Z = z_i; \vec{u}_k, \Sigma_k), k = 1, \cdots, K \quad (3)$$

(3)中使用来自同一个高斯分布的样本 $x_i, i \in i|z_i = k, k = 1, \cdots, K$ 构造出了第 k 个高斯分布的NLL(Negative Log Likelihood, 负对数似然) [12] 函数并进行最小化以得到参数

$$\overline{u}\_k^{\star}, \sum\_k^{*}$$

. 而对于分布 $\mathcal{Q}$, 我们可通过统计不同高斯分布所产生的样本个数来对其参数 $\alpha_k = \mathcal{Q}(Z = k), k \in 1, \cdots, K$ 进行估计. 至此, 所有的未知参数都得到了很好的估计. 样本 x_i 生成的概率可由

$$
\begin{aligned}
p\left(x_i;\Theta\right) &= \sum_{z_i} p\left(x_i,z_i;\Theta\right) \\
&= \sum_{k=1}^{K} \mathcal{N}\left(x_i|Z=k;\vec{u}_k,\Sigma_k\right)\mathcal{Q}(Z=k) \\
&= \sum_{k=1}^{K} \alpha_k \cdot \mathcal{N}\left(x_i|Z=k;\vec{u}_k,\Sigma_k\right)
\end{aligned} \tag{4}
$$

微信扫一扫
关注公众号

得到.

但是, 由于事实上观测数据只有不完全数据

$$\mathcal{D}=\{x_i\}_{i=1}^{n}$$

, 所以(3)(4)的参数估计方法无法使用. 我们尝试直接对数据集 $\mathcal{D}$ 使用(1)来构造NLL函数并使用MLE来指导参数的估计, 具体地

$$
\begin{aligned}
\mathcal{NLL}(\Theta) &= -\log\prod_{i=1}^{n} p\left(x_i;\Theta\right) \\
&= -\sum_{i=1}^{n} \log p\left(x_i;\Theta\right) \\
&= -\sum_{i=1}^{n} \log\left[\sum_{z_i} p\left(x_i,z_i;\Theta\right)\right] \\
&= -\sum_{i=1}^{n} \log\left[\sum_{k=1}^{K} \alpha_k \cdot p\left(x_i|z_i=k;\Theta\right)\right]
\end{aligned} \tag{5}
$$

通过求解带约束的最优化问题

$$
\begin{aligned}
&\min \mathcal{NLL}(\Theta) \\
&\text{s.t. } \textstyle\sum_{k=1}^{K}\alpha_k = 1
\end{aligned} \tag{6}
$$

求解(6)这个最优化问题相对比较困难, 原因有两个: 一NLL函数中, 对数函数的自变量带有连加操作; 二带有约束. 那么我们该如何对(5)进行参数估计呢? 答案便是EM算法 [3][4][5][6].

EM算法对带隐变量的模型进行参数估计

虽然直接求解(6)的最优化问题比较困难, 但是EM算法并没有摒弃使用MLE来指导NLL函数的优化以获得参数估计值的思路, 相反, EM算法延续了MLE的思路, 通过不断地构造对数似然函数的下界函数, 并对这个较为容易求解的下界函数进行最优化, 以增大对数似然函数取值的下界, 使得在不断的迭代操作后, 对数似然函数的取值能逼近最大值, 从而完成参数的估计.

要厘清EM算法的流程, 我们需要先了解Jensen’ s inequality [2],

In the context of probability theory, it is generally stated in the following form: if X is a random variable and φ is a convex function, then \phi[\mathbf{E}(X)] \leq \mathbf{E}[\phi(X)] , notice that the equality holds if X is constant (degenerate random variable) or if φ is linear.

受Jensen’ s inequality的启发, 如果我们能在(5)中对数函数的连加操作里面构造出一个关于 z_i 的期望, 那我们就能将连加操作移至对数函数的外面了, 具体地, 对于对数似然函数我们有

$$\begin{aligned}\mathcal{LL}(\Theta) &= \sum_{i=1}^{n}\log\left[\sum_{z_i}p\left(x_i,z_i;\Theta\right)\right]\\ &= \sum_{i=1}^{n}\log\left[\sum_{z_i}p\left(z_i|x_i;\Theta_{t-1}\right)\cdot\frac{p\left(x_i,z_i;\Theta\right)}{p\left(z_i|x_i;\Theta_{t-1}\right)}\right] \quad (7)\\ &\geq \sum_{i=1}^{n}\sum_{z_i}p\left(z_i|x_i;\Theta_{t-1}\right)\cdot\log\left[\frac{p\left(x_i,z_i;\Theta\right)}{p\left(z_i|x_i;\Theta_{t-1}\right)}\right]\end{aligned}$$

微信扫一扫
关注公众号

在(7)中, 我们借助在第 t-1 次迭代中得到的参数估计值 Θ_{t-1} 来获得 z_i 关于 x_i 的后验分布

$$p\left(z_i|x_i;\Theta_{t-1}\right) = \frac{p(x_i|z_i;\Theta_{t-1})\cdot p(z_i;\Theta_{t-1})}{\sum_{z_i}p(x_i|z_i;\Theta_{t-1})\cdot p(z_i;\Theta_{t-1})} \quad (8)$$

高斯 alpha 贝叶斯

此时的后验概率 $p\left(z_i|x_i;\Theta_{t-1}\right)$ 是一个可计算的常数. 再使用这个关于 z_i 的后验分布构造期望

$$\mathbf{E}_{z_i}\left[\frac{p(x_i,z_i;\Theta)}{p(z_i|x_i;\Theta_{t-1})}\right] = \sum_{z_i}p\left(z_i|x_i;\Theta_{t-1}\right)\cdot\frac{p(x_i,z_i;\Theta)}{p(z_i|x_i;\Theta_{t-1})} \quad (9)$$

而后由Jensen’s inequality即可得到

$$\begin{aligned}\log\mathbf{E}_{z_i}\left[\frac{p(x_i,z_i;\Theta)}{p(z_i|x_i;\Theta_{t-1})}\right] &\geq \mathbf{E}_{z_i}\left[\log\frac{p(x_i,z_i;\Theta)}{p(z_i|x_i;\Theta_{t-1})}\right]\\ \log\left[\sum_{z_i}p\left(z_i|x_i;\Theta_{t-1}\right)\cdot\frac{p(x_i,z_i;\Theta)}{p(z_i|x_i;\Theta_{t-1})}\right] &\geq \sum_{z_i}p\left(z_i|x_i;\Theta_{t-1}\right)\cdot\log\left[\frac{p(x_i,z_i;\Theta)}{p(z_i|x_i;\Theta_{t-1})}\right]\end{aligned} \quad (10)$$

(10)对于 $i=1,\cdots,n$ 均成立, 所以由(8)(9)(10)我们最终可得(7).

我们不妨将(7)中的下界函数记为

$$\begin{aligned}\mathcal{B}\left(\Theta,\Theta_{t-1}\right) &= \sum_{i=1}^{n}\mathbf{E}_{z_i}\left[\log\frac{p\left(x_i,z_i;\Theta\right)}{p\left(z_i|x_i;\Theta_{t-1}\right)}\right]\\ &= \sum_{i=1}^{n}\sum_{z_i}p\left(z_i|x_i;\Theta_{t-1}\right)\cdot\log\left[\frac{p\left(x_i,z_i;\Theta\right)}{p\left(z_i|x_i;\Theta_{t-1}\right)}\right]\end{aligned} \quad (11)$$

则对数似然函数与下界函数有

$$\mathcal{LL}(\Theta)\geq\mathcal{B}\left(\Theta,\Theta_{t-1}\right) \quad (12)$$

由(12)我们可知, 借助Jensen’s inequality我们构造出了对数似然函数 $\mathcal{LL}(\Theta)$ 的一个下界函数 $\mathcal{B}(\Theta,\Theta_{t-1})$, 而对这个下界函数进行最优化是较为简单的事情, 所以我们可对下界函数进行求解, 以提升对数似然函数的函数值, 这便是EM算法的核心内容. 具体地, 我们将EM算法分为两步:

对于第 t 次迭代,

借助第 t-1 次迭代的参数估计值 Θ_{t-1}, 构造对数似然函数的下界函数

$$\mathcal{B}\left(\Theta,\Theta_{t-1}\right) = \sum_{i=1}^{n}\sum_{z_i}p\left(z_i|x_i;\Theta_{t-1}\right)\cdot\log\left[\frac{p(x_i,z_i;\Theta)}{p(z_i|x_i;\Theta_{t-1})}\right] \quad (13)$$

(13)的构成部件亦既是EM算法中的Expectation;

1. 对(13)进行最优化, 得到当前的参数估计值

$$\Theta_t$$

$$\Theta_t = \underset{\Theta}{\operatorname{argmax}}\mathcal{B}\left(\Theta,\Theta_{t-1}\right) \quad (14)$$

(14)亦既是EM算法中的Maximum. 通过不断地迭代, 直至对数似然函数收敛.

当然, 要想以上述流程完成参数估计, 我们还需要保证对数似然函数是能收敛的, 即

$$\mathcal{LL}\left(\Theta_t\right) \geq \mathcal{LL}\left(\Theta_{t-1}\right), t \in [1, +\infty) (15)$$

对于(10), 当 $\Theta = \Theta_{t-1}$ 时有

$$\frac{p(x_i, z_i; \Theta)}{p(z_i|x_i; \Theta_{t-1})} = \frac{p(z_i|x_i; \Theta_{t-1}) \cdot p(x_i|\Theta_{t-1})}{p(z_i|x_i; \Theta_{t-1})} = p(x_i|\Theta_{t-1})$$

微信扫一扫
关注公众号

不受 z_i 取值影响为常数, 此时Jensen’ s inequality等号成立, 故有

$$\sum_{i=1}^{n} \log \left[\sum_{z_i} p\left(z_i|x_i; \Theta_{t-1}\right) \cdot \frac{p\left(x_i, z_i; \Theta_{t-1}\right)}{p\left(z_i|x_i; \Theta_{t-1}\right)} \right] = \sum_{i=1}^{n} \sum_{z_i} p\left(z_i|x_i; \Theta_{t-1}\right) \cdot \log \left[\frac{p\left(x_i, z_i; \Theta_{t-1}\right)}{p\left(z_i|x_i; \Theta_{t-1}\right)} \right]$$

$$\mathcal{B}\left(\Theta_{t-1}, \Theta_{t-1}\right) = \mathcal{LL}\left(\Theta_{t-1}\right)$$

而对于 $\Theta = \Theta_t$, 由(7)我们有

$$\begin{aligned} \mathcal{LL}\left(\Theta_t\right) &= \sum_{i=1}^{n} \log \left[\sum_{z_i} p\left(z_i|x_i; \Theta_{t-1}\right) \cdot \frac{p\left(x_i, z_i; \Theta_t\right)}{p\left(z_i|x_i; \Theta_{t-1}\right)} \right] \\ &\geq \sum_{i=1}^{n} \sum_{z_i} p\left(z_i|x_i; \Theta_{t-1}\right) \cdot \log \left[\frac{p\left(x_i, z_i; \Theta_t\right)}{p\left(z_i|x_i; \Theta_{t-1}\right)} \right] (17) \\ &= \mathcal{B}\left(\Theta_t, \Theta_{t-1}\right) \end{aligned}$$

由(14)(16)(17)可得

$$\mathcal{LL}\left(\Theta_t\right) \geq \mathcal{B}\left(\Theta_t, \Theta_{t-1}\right) \geq \mathcal{B}\left(\Theta_{t-1}, \Theta_{t-1}\right) = \mathcal{LL}\left(\Theta_{t-1}\right) (18)$$

由(18)EM算法的收敛性得以保证, 我们可以使用其来对GMM模型进行参数估计了.

EM算法对GMM模型进行求解的具体过程

由上一节我们知道, 使用EM算法来对GMM模型进行参数估计的核心是要构造出其期望函数作为下界函数, 并对下界函数进行最优化. 假设我们已经得到GMM模型第 t-1 次迭代的参数估计值为

$$\Theta_{t-1} = \left\{\alpha_k^{t-1}, \vec{u}_k^{t-1}, \Sigma_k^{t-1}\right\}_{k=1}^{K}$$

, 由(8)我们可获得 z_i 关于 x_i 的后验分布为

贝叶斯

$$\begin{aligned} p\left(z_i = k|x_i; \Theta_{t-1}\right) &= \frac{p\left(x_i|z_i = k; \Theta_{t-1}\right) \cdot p\left(z_i = k; \Theta_{t-1}\right)}{\sum_{z_i} p\left(x_i|z_i = k; \Theta_{t-1}\right) \cdot p\left(z_i = k; \Theta_{t-1}\right)} \\ &= \frac{\alpha_k^{t-1} \cdot \mathcal{N}\left(x_i; \vec{u}_k^{t-1}, \Sigma_k^{t-1}\right)}{\sum_{k=1}^{K} \alpha_k^{t-1} \cdot \mathcal{N}\left(x_i; \vec{u}_k^{t-1}, \Sigma_k^{t-1}\right)}, k = 1, \cdots, K (19) \end{aligned} (19)$$

(19)是可以被直接计算出来的常数, 我们将其记为 q_{ik} . 由(19)所表示的 z_i 的后验分布与(11), 我们可得GMM模型的对数似然函数的下界函数为

$$\begin{aligned}\mathcal{B}\left(\Theta,\Theta_{t-1}\right)&=\sum_{i=1}^{n}\mathbf{E}_{z_i}\left[\log\frac{p\left(x_i,z_i;\Theta\right)}{p\left(z_i|x_i;\Theta_{t-1}\right)}\right]\\&=\sum_{i=1}^{n}\sum_{z_i}p\left(z_i|x_i;\Theta_{t-1}\right)\cdot\log\left[\frac{p\left(x_i,z_i;\Theta\right)}{p\left(z_i|x_i;\Theta_{t-1}\right)}\right]\\&=\sum_{i=1}^{n}\sum_{k=1}^{K}q_{ik}\cdot\log\left[\frac{p\left(x_i|z_i=k;\Theta\right)\cdot p\left(z_i=k;\Theta\right)}{q_{ik}}\right]\\&=\sum_{i=1}^{n}\sum_{k=1}^{K}\left[q_{ik}\cdot\log p\left(x_i|z_i=k;\Theta\right)+q_{ik}\log\alpha_k-q_{ik}\log q_{ik}\right]\end{aligned}\tag{20}$$

微信扫一扫
关注公众号

而(20)中

$$\begin{aligned}\log p\left(x_i|z_i=k;\Theta\right)&=\log\left[\frac{1}{\left(2\pi\right)^{\frac{n}{2}}\left|\Sigma_k\right|^{\frac{1}{2}}}\cdot e^{-\frac{1}{2}\cdot\left[\left(x_i-\vec{\mu}_k\right)^{\top}\Sigma_k^{-1}\left(x_i-\vec{\mu}_k\right)\right]}\right]\\&=-\frac{n}{2}\log 2\pi-\frac{1}{2}\log\left|\Sigma_k\right|-\frac{1}{2}\cdot\left[\left(x_i-\vec{\mu}_k\right)^{\top}\Sigma_k^{-1}\left(x_i-\vec{\mu}_k\right)\right]\end{aligned}\tag{21}$$

整合(20)(21)并去掉与优化无关的项, 可得下界函数的最终形式

$$\begin{aligned}\mathcal{B}\left(\Theta,\Theta_{t-1}\right)&=\sum_{i=1}^{n}\sum_{k=1}^{K}q_{ik}\left[-\frac{n}{2}\log 2\pi-\frac{1}{2}\log\left|\Sigma_k\right|-\frac{1}{2}\cdot\left[\left(x_i-\vec{\mu}_k\right)^{\top}\Sigma_k^{-1}\left(x_i-\vec{\mu}_k\right)\right]+\log\alpha\right.\\&=\sum_{i=1}^{n}\sum_{k=1}^{K}q_{ik}\left[-\frac{1}{2}\log\left|\Sigma_k\right|-\frac{1}{2}\cdot\left[\left(x_i-\vec{\mu}_k\right)^{\top}\Sigma_k^{-1}\left(x_i-\vec{\mu}_k\right)\right]+\log\alpha_k\right]\end{aligned}$$

有了下界函数(22), 我们就可以来求得第 t 次迭代的参数估计值了.

对于均值向量

$$\vec{\mu}_k,k=1,\cdots,K$$

结果见下一页

, 令其偏导数为零

对于协方差矩阵

$$\Sigma_k,k=1,\cdots,K$$

, 令其偏导数为零

$$\Sigma_k=\frac{\sum_{i=1}^{n}q_{ik}\left(x_i-\vec{\mu}_k\right)\left(x_i-\vec{\mu}_k\right)^{\mathrm{T}}}{\sum_{i=1}^{n}q_{ik}},k=1,\cdots,K\tag{24}$$

对于隐变量所服从的分布 $Z\sim\mathcal{Q}$ 的参数

$$\alpha_k,k=1,\cdots,K$$

, 因为需要满足 $\sum_{k=1}^{K}\alpha_k=1$, 由(22)使用拉格朗日乘子法并去掉与所求变量无关的项, 得到拉格朗日函数

$$\mathcal{L}\left(\alpha_1,\cdots,\alpha_k,\lambda\right)=\sum_{i=1}^{n}\sum_{k=1}^{K}q_{ik}\log\alpha_k+\lambda\left(\sum_{k=1}^{K}\alpha_k-1\right)\tag{25}$$

对(25)进行偏导数求解并令其为0, 可得

$$\because \frac{\partial \mathcal{L}}{\partial \alpha_k} = \frac{1}{\alpha_k}\sum_{i=1}^{n} q_{ik} + \lambda = 0 \Rightarrow \alpha_k = -\frac{\sum_{i=1}^{n} q_{ik}}{\lambda}, k = 1, \cdots, K$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{k=1}^{K} \alpha_k - 1 = 0 \Rightarrow \sum_{k=1}^{K} \alpha_k = 1 \quad (26)$$

$$\therefore \alpha_k = \frac{\sum_{i=1}^{n} q_{ik}}{n}, k = 1, \cdots, K$$

联立两个等式可得alphak

微信扫一扫
关注公众号

由(23)(24)(26)我们可得下界函数 $\mathcal{B}(\Theta, \Theta_{t-1})$ 的最优解为

$$\underset{\Theta}{\operatorname{argmax}} \mathcal{B}(\Theta, \Theta_{t-1}) =$$

$$\begin{cases} \vec{\mu}_k = \frac{\sum_{i=1}^{n} q_{ik} x_i}{\sum_{i=1}^{n} q_{ik}}, & k = 1, \cdots, K \\ \Sigma_k = \frac{\sum_{i=1}^{n} q_{ik} (x_i - \vec{\mu}_k)(x_i - \vec{\mu}_k)^{\mathrm{T}}}{\sum_{i=1}^{n} q_{ik}}, & k = 1, \cdots, K \\ \alpha_k = \frac{\sum_{i=1}^{n} q_{ik}}{n}, & k = 1, \cdots, K \end{cases} \quad (27)$$

由(27)我们就可以给出EM算法对GMM模型进行求解的具体过程:

对于第 t 次迭代,

1.借助第 t-1 次迭代的参数估计值

$$\Theta_{t-1} = \left\{\alpha_k^{t-1}, \vec{u}_k^{t-1}, \Sigma_k^{t-1}\right\}_{k=1}^{K}$$ 最后想要的结果

, 构造GMM模型对数似然函数的下界函数

$$\mathcal{B}(\Theta, \Theta_{t-1}) = \sum_{i=1}^{n}\sum_{k=1}^{K} q_{ik}\left[-\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}\cdot\left[(x_i - \vec{\mu}_k)^{\top}\Sigma_k^{-1}(x_i - \vec{\mu}_k)\right] + \log\alpha_k\right]$$ (E-step)

2.对(E-step)进行最优化, 得到当前的参数估计值

$$\Theta_t$$

$$\Theta_t = \underset{\Theta}{\operatorname{argmax}}\, \mathcal{B}(\Theta, \Theta_{t-1}) = \begin{cases} q_{ik} = \frac{\alpha_k^{t-1}\cdot\mathcal{N}(x_i;\vec{u}_k^{t-1},\Sigma_k^{t-1})}{\sum_{k=1}^{K}\alpha_k^{t-1}\cdot\mathcal{N}(x_i;\vec{u}_k^{t-1},\Sigma_k^{t-1})}, & k = 1, \cdots, K \\ \vec{\mu}_k = \frac{\sum_{i=1}^{n} q_{ik} x_i}{\sum_{i=1}^{n} q_{ik}}, & k = 1, \cdots, K \\ \Sigma_k = \frac{\sum_{i=1}^{n} q_{ik}(x_i - \vec{\mu}_k)(x_i - \vec{\mu}_k)^{\mathrm{T}}}{\sum_{i=1}^{n} q_{ik}}, & k = 1, \cdots, K \\ \alpha_k = \frac{\sum_{i=1}^{n} q_{ik}}{n}, & k = 1, \cdots, K \end{cases}$$ (M-step)

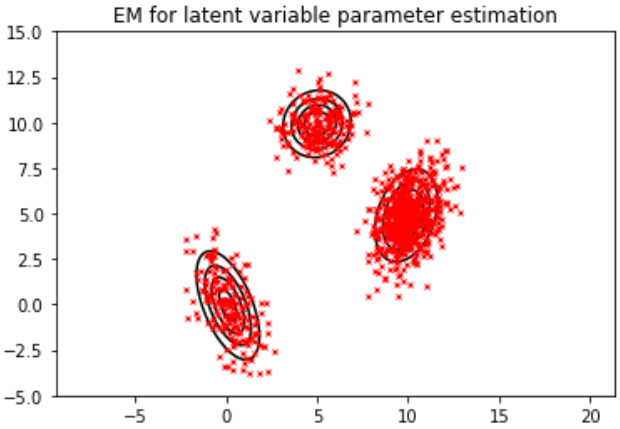
程序结束的条件是估计的参数变化量足够小
最后想要的结果是估计的参数

求解不完全数据问题实例的概率分布

由前述章节我们已经得到了使用EM算法对GMM模型进行求解的具体过程, 现在我们就可以来解决本文开头所阐述的不完全数据问题实例了. 对其中的1000个样本点, 我们使用GMM模型来对其分布进行建模, 在每次迭代中, 我们先利用第 t-1 次迭代的参数估计值

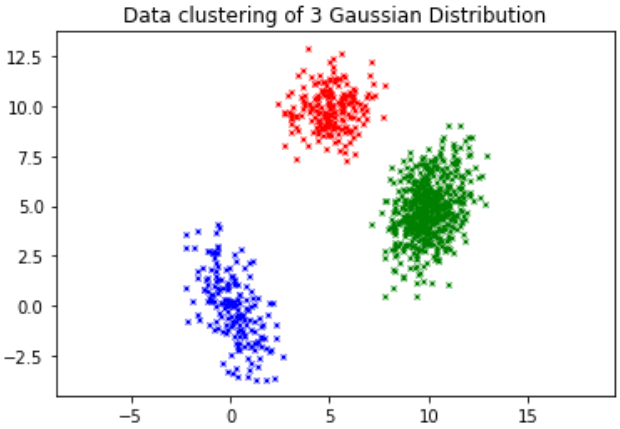
$$\Theta_{t-1} = \left\{\alpha_k^{t-1}, \vec{u}_k^{t-1}, \Sigma_k^{t-1}\right\}_{k=1}^{K}$$

求出 z_i 关于 x_i 的后验分布 q_{ik} , 再根据M-step求出第 t 次的参数估计值, 后反复迭代直至收敛.



使用EM算法求得高斯分布概率密度函数等高线示意图

上图为使用EM算法对GMM模型进行参数估计后得到的各个高斯分布概率密度函数的等高线示意图, 可以看到, 各个高斯分布概率密度函数的等高线形状与数据的分布情况有非常高的吻合程度.



样本点所属高斯分布的预测着色示意图

上图为1000个样本点所属高斯分布的预测着色示意图, 在使用EM算法得到GMM模型的参数估计值后, 只需计算

$$\underset{z_i}{\operatorname{argmax}}\, p(z_i|x_i;\Theta_t), i=1,\cdots,n$$

即可得到样本点所属高斯分布的预测值.

训练代码的主体如下:

```
for t in range(1, 1000):
    # 求出 z_i 关于 x_i 的后验分布 q_{ik}
    q = compute_q(alpha, mu, sigma, samples, total, n_gauss)
    # 计算均值向量的估计值
    for k in range(n_gauss):
        mu[k] = np.sum(
            np.stack((q[:, k], q[:, k]), axis=1) * samples,
            axis=0) / np.sum(q[:, k])
    # 计算协方差矩阵的估计值
    for k in range(n_gauss):
        res_mat = np.mat("0.0 0.0; 0.0 0.0")
        for i in range(total):
            vec = np.mat(samples[i]) - np.mat(mu[k])
            res_mat += (q[i, k] * np.matmul(vec.T, vec))
        res_mat /= np.sum(q[:, k])
        sigma[k] = res_mat
    # 计算隐变量分布的参数估计值
    for k in range(n_gauss):
        alpha[k] = (np.sum(q[:, k]) / total)
```

k-means算法与GMM模型的关系

在前面我们曾提到, k-means [10]算法是GMM模型的一个变种, 那具体这二者之间是一个怎么样的关系呢? 为了回答这个问题, 我们需要先对GMM模型做几个限制:
不完全数据

$$\{(x_i)\}_{i=1}^n$$

中的每个样本将不再依概率 $Z \sim \mathcal{Q}$ 归属于每个高斯分布, 后验概率 $q_{ik} = p\left(z_i|x_i;\Theta\right), k = 1, \cdots, K$ 值为1, 其余皆取值为0, 即每次迭代中每个样本将以概率为1只归属于一个确定的高斯分布, 而不是以后验 q_{ik} 归属于各个

各个高斯分布的协方差矩阵为单位矩阵 $\mathbf{I}$;

隐变量 $Z \sim \mathcal{Q}$ 为均匀分布, 即样本是等概率从各个高斯分布中采样得到.
有了以上的限制, 我们可以来重新审视M-step所表达的意义了

$$\underset{\Theta}{\operatorname{argmax}}\, \mathcal{B}(\Theta, \Theta_{t-1}) = \begin{cases} q_{ik} = \frac{\alpha_k^{t-1} \cdot \mathcal{N}(x_i; \vec{u}_k^{t-1}, \Sigma_k^{t-1})}{\sum_{k=1}^K \alpha_k^{t-1} \cdot \mathcal{N}(x_i; \vec{u}_k^{t-1}, \Sigma_k^{t-1})} \\ \vec{\mu}_k = \frac{\sum_{i=1}^n q_{ik} x_i}{\sum_{i=1}^n q_{ik}} \\ \Sigma_k = \frac{\sum_{i=1}^n q_{ik}(x_i - \vec{\mu}_k)(x_i - \vec{\mu}_k)^{\mathrm{T}}}{\sum_{i=1}^n q_{ik}} \\ \alpha_k = \frac{\sum_{i=1}^n q_{ik}}{n} \end{cases} \Rightarrow \begin{cases} \underbrace{q_{ik} = \frac{\mathcal{N}(x_i; \vec{u}_k^{t-1}, \mathbf{I})}{\sum_{k=1}^K \mathcal{N}(x_i; \vec{u}_k^{t-1}, \mathbf{I})}}_{\text{取值最大赋值1,反之为0}} \\ \underbrace{\vec{\mu}_k = \frac{\sum_{i=1}^n q_{ik} x_i}{\sum_{i=1}^n q_{ik}}}_{\text{属同个高斯分布的样本均值}} \\ \underbrace{\Sigma_k = \mathbf{I}}_{\text{协方差矩阵为单位矩阵}\mathbf{I}} \\ \underbrace{\alpha_k = \frac{1}{K}}_{\text{等概率从各高斯分布采样}} \end{cases} \quad \text{(M-step-constraint)}$$

我们来分析一下(M-step-constraint)的操作. 首先, 由于协方差矩阵 $\Sigma_k = \mathbf{I}, k = 1, \cdots, K$, 所以后验概率 q_{ik} 中的高斯分布概率密度函数值实际上反比于样 x_i 与对应高斯分布均值向量 $\vec{\mu}_k^{t-1}$ 的欧氏距离, 即

$$\mathcal{N}\left(x_i; \vec{u}_k^{t-1}, \mathbf{I}\right) \propto \left[\left(x_i - \vec{u}_k^{t-1}\right)^\top \left(x_i - \vec{u}_k^{t-1}\right)\right]^{-1}$$

所以样本 $x_i, i = 1, \cdots, n$ 将以概率为1被归于与之欧氏距离最小的均值向量所属的高斯分布; 然后, 使用归属于同个高斯分布的样本的均值更新对应高斯分布的均值向量. 算法流程具体为

对于第 t 次迭代,

1.根据样本 $x_i, i = 1, \cdots, n$ 与第 t-1 次迭代各个高斯分布的均值向量 $\vec{\mu}_k^{t-1}, k = 1, \cdots, K$ 的欧氏距离 $\left(x < em > i - \vec{u} < /em > k^{t-1}\right)^\top \left(x < em > i - \vec{u} < /em > k^{t-1}\right)$ 将样本标记为属于与之距离最小的高斯分布;

2.使用标记为属于同一个高斯分布的样本的均值向量更新对应高斯分布的均值向量.

这恰恰是k-means算法的完整描述, 只是在聚类操作中我们习惯使用”簇”的概念来表达此处的高斯分布. 由此可见, 我们能从GMM模型的EM算法求解过程中, 通过加以限制得到k-means算法, 亦既k-means算法是GMM模型的一个特例.

总结

GMM模型是典型的概率图模型, 其优异的数学性质使之在拟合数据分布时有很强的建模能力. 求解GMM模型的EM算法给带隐变量的模型的参数估计提供了强有力的武器, 其在工业界中亦得到广泛应用.

引用

[1] Wikipedia contributors. “中心极限定理.”维基百科, 自由的百科全书. 维基百科, 自由的百科全书, 9 May 2018. Web. 9 May 2018.‹https://zh.wikipedia.org/w/index.php?title=%E4%B8%AD%E5%BF%83%E6%9E%81%E9%99%90%E5%AE%9A%E7%90%86&oldid=49494817› (https://zh.wikipedia.org/w/index.php?title=%E4%B8%AD%E5%BF%83%E6%9E%81%E9%99%90%E5%AE%9A%E7%90%86&oldid=49494817›).

[2] Wikipedia contributors. (2019, March 14). Jensen’ s inequality. InWikipedia, The Free Encyclopedia. Retrieved 03:35, May 27, 2019, fromhttps://en.wikipedia.org/w/index.php?title=Jensen%27s_inequality&oldid=887772941 (https://en.wikipedia.org/w/index.php?title=Jensen%27s_inequality&oldid=887772941)

[3] Bishop, C. (2013).Pattern recognition and machine learning. 2nd ed. New York: Springer, pp.423-455.

[4] Li, H. (2012).统计学习方法. 1st ed. Beijing: 清华大学出版社, pp.155~165.

[5] Mp.weixin.qq.com. (2019). 理解EM算法. [online] Available at: https://mp.weixin.qq.com/s/5V4LgKDNID4DhBE0ky6fRQ (https://mp.weixin.qq.com/s/5V4LgKDNID4DhBE0ky6fRQ) [Accessed 28 May 2019].

[6] August, 一. (2018). 人人都懂EM算法. [online] Zhuanlan.zhihu.com. Available at: https://zhuanlan.zhihu.com/p/36331115 (https://zhuanlan.zhihu.com/p/36331115) [Accessed 29 May 2019].

[7] Zhang, L. (2019). 多元高斯分布完全解析. Retrieved from https://zhuanlan.zhihu.co (https://zhuanlan.zhihu.com/p/58987388)

[8] Xie, P. (2015). 概率图模型（PGM）有必要系统地学习一下吗？. Re https://www.zhihu.com/question/23255632/answer/56330768 (https://www.zhihu.com/question/23255632/answer/56330768)

微信扫一扫
关注公众号

[9] Wikipedia contributors. (2019, May 2). Graphical model. InWikipedia, The Free Encyclopedia. Retrieved 06:58, June 5, 2019, fromhttps://en.wikipedia.org/w/index.php?title=Graphical_model&oldid=895103850 (https://en.wikipedia.org/w/index.php?title=Graphical_model&oldid=895103850)

[10] Wikipedia contributors. (2019, June 3). K-means clustering. InWikipedia, The Free Encyclopedia. Retrieved 07:00, June 5, 2019, fromhttps://en.wikipedia.org/w/index.php?title=K-means_clustering&oldid=900149325 (https://en.wikipedia.org/w/index.php?title=K-means_clustering&oldid=900149325)

[11] Wikipedia contributors. (2019, May 9). Hidden Markov model. InWikipedia, The Free Encyclopedia. Retrieved 07:10, June 5, 2019, fromhttps://en.wikipedia.org/w/index.php?title=Hidden_Markov_model&oldid=896345228 (https://en.wikipedia.org/w/index.php?title=Hidden_Markov_model&oldid=896345228)

[12] Wikipedia contributors. (2019, June 4). Likelihood function. InWikipedia, The Free Encyclopedia. Retrieved 10:57, June 5, 2019, fromhttps://en.wikipedia.org/w/index.php?title=Likelihood_function&oldid=900332716 (https://en.wikipedia.org/w/index.php?title=Likelihood_function&oldid=900332716)

高斯混合模型与EM算法的数学原理及应用实例.pdf

Follow us

CSDN博客 (https://blog.csdn.net/SIGAI_CSDN)

知乎 (https://www.zhihu.com/org/bei-jing-zhang-liang-wu-xian-ke-ji-you-xian-gong-si/posts)

新浪微博 (https://weibo.com/p/1006066555787294)

DEEPACTION (about.html)
北京张量无限科技有限公司
北京市海淀区中关村智造大街G座1层
info@sigai.cn
联系我们 (about.html)