sklearn实现与传统实现的对比解读

1. 传统

（1）随机初始化相关参数：均值$\mu_k^{(0)}$、方差$\Sigma_k^{(0)}$、各个高斯分布的权重$\pi_k^{(0)}$

（2）利用EM迭代计算

·E step

a. 根据第k个高斯分布的相关参数（均值μ_k/方差Σ_k）生成高斯模型$N(\mu_k, \Sigma_k)$

b. 计算样本在各个高斯分布函数下的值与相应高斯分布的概率的乘积：$\pi_k^{i-1} N(\mathrm{x}_n | \mu_k, \Sigma_k)$

c. 求和$\sum_{k=1}^{K} \pi_k^{i-1} N(\mathrm{x}_n | \mu_k, \Sigma_k)$，用于归一化

d. 归一化：计算各个高斯分布对于数据样本n的贡献

$$\gamma(z_{nk}) = \frac{\pi_k^{i-1} N(\mathrm{x}_n | \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k^{i-1} N(\mathrm{x}_n | \mu_k, \Sigma_k)}$$

·M step

a. 计算第k个高斯分布对各个样本的贡献和N_k

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

b. 更新高斯分布的相关参数（均值μ_k^{new}\协方差Σ_k^{new}\权重系数π_k^{new}）

该过程和传统高斯分布的均值和方差计算类似：

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathrm{x_n}$$

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

$$\pi_k^{new} = \frac{N_k}{N}$$

c. 按照得到的值，计算对数似然函数，判断是否符合要求

$$lnp(X | \pi, \mu, \Sigma) = \sum_{n=1}^{N} ln \left\{ \sum_{k=1}^{K} \pi_k N(X_n | \mu_k, \Sigma_k) \right\}$$

2. sklearn.mixture.GaussianMixture

sklearn中实现了计算混合高斯模型的包装类，使用方法简单，可作为联邦算法的验证。

初始化参数：

- n_components 高斯混合模型的个数
- tol 阈值
- max_iter 最大迭代次数
- n_init 初始化次数，用于产生最佳初始参数
- init_params 初始化高斯混合模型的权重参数，均值和协方差的方式
- random_state 随机数种子
- means_init 初始化均值
- weights_init 初始化权重

属性：

- weights_ 混合高斯模型中各个高斯模型的概率
- means_ 混合高斯模型中各个高斯模型的均值
- covariances_ 混合高斯模型中各个高斯模型的协方差
- n_iter_ EM的最佳拟合达到收敛所使用的迭代次数

源码解读与分析

解读原因：由于最终测试的时候发现和sklearn实现存在部分误差，打算还是深入看一下sklearn是如何实现的。

准备

初始化先验概率、均值、协方差等。

此外，sklearn将传统的协方差Σ进行了拆分，使用Cholesky进行LU分解，分解后得到协方差的上三角矩阵L（对角线下方全为0），然后进行了求逆，得到L^{-1}，方便之后EM迭代的计算。

E-step：计算后验概率

· **step1**: 将传统的高斯模型进行修改，为了便于计算，对指数部分处理:

$$e^{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)}$$

记$y = XL^{-1} - \mu L^{-1}$，之后对y取平方，得到:

$$(X-\mu)^T\Sigma^{-1}(X-\mu) = sqaure(XL^{-1} - \mu L^{-1})$$

· **step2**: 对于原来的计算公式，为了方便运算，都加了对数运算，此时高斯分布模型的公式变为:

$$log(\frac{1}{(2\pi)^{D/2}|\Sigma_j|^{1/2}}e^{-\frac{1}{2}(X_i-\mu_j)^T\Sigma_j^{-1}(X_i-\mu_j)})$$
$$= -\frac{1}{2}(Dlog(2\pi) + sqaure(XL^{-1} - \mu L^{-1})) + |L|$$

· **step3**: 计算分母，再利用指数运算，恢复数据，计算全概率，之后又加上对数，得到$log(norm)$。(可能是便于计算)

· **step4**: 根据贝叶斯公式，计算样本i在各个高斯分布下的后验概率，如下:
$log(\gamma_{ij}) = log(\pi_i N(X_i)|\mu_j, \Sigma_j) = log(\pi) + log(norm)$

M-step：根据样本的后验概率更新高斯函数的参数

计算方法和传统的一样:

此时，由于上述得到的后验概率取了对数，在这里利用指数恢复为γ_{ij}。之后，再进行各个参数的更新。