混合高斯分布 EM 算法估计

杨洪易 ZY2203119
yhongyi@buaa.edu.cn

摘要:

本文以给定的两种高斯分布均值与标准差生成混合高斯分布样本，使用 EM 算法完成混合高斯分布模型参数的估计，并结合高斯分布样本参数真值完成估计参数的模型性能评估。

简介:

一.混合高斯分布

一维高斯分布的概率密度函数如下:

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

一维高斯分布曲线取决于两个参数：**均值和标准差**。分布的均值决定了图形中心的位置，标准差决定了图像的高度和宽度。标准差大时，曲线呈现出“矮胖”，标准差小时，曲线呈现出“高瘦”。因此通过改变均值和标准差，根据其概率密度函数得到不同的高斯分布。

混合高斯分布概率密度函数是各个组合起来分布的出现概率加权平均:

$$p(x)=\sum_{k=1}^{K}p(k)p(x|k)=\sum_{k=1}^{K}\pi_k N(x|u_k,\Sigma_k)$$

其中，$p(x|k)=N(x|u_k,\Sigma_k)$ 是第 k 个高斯模型的概率密度函数，可以看成选定第 k 个模型后，该模型产生 x 的概率。

混合高斯模型的本质就是融合几个单高斯模型，来使得模型更加复杂，从而产生更复杂的样本。理论上，如果某个混合高斯模型融合的高斯模型个数足够多，它们之间的权重设定得足够合理，这个混合模型可以拟合任意分布的样本。

二.EM 算法

EM（Expectation-Maximum）算法也称期望最大化算法。

EM 算法是一种迭代优化策略，由于它的计算方法中每一次迭代都分两步，其中一个为期望步（E 步），另一个为极大步（M 步），所以算法被称为 EM 算法（Expectation-Maximization Algorithm）。EM 算法受到缺失思想影响，最初是为了解决数据缺失情况下的参数估计问题，其基本思想是：首先根据己经给出的观测数据，估计出模型参数的值；然后再依据上一步估计出的参数值估计缺失数据的值，再根据估计出的缺失数据加上之前己经观测到的数据重新再对参数值进行估计，然后反复迭代，直至最后收敛，迭代结束。

1. 极大似然估计

极大似然估计，只是一种概率论在统计学的应用，它是参数估计的方法之一。极大似然估计是建立在这样的思想上：已知某个参数能使这个样本出现的概率最大，我们当然不会再去选

择其他小概率的样本，所以干脆就把这个参数作为估计的真实值。极大似然函数表示为：

$$L(\theta) = L(x_1, x_2, \ldots, x_n; \theta) = \prod_{i=1}^{n} p(x_i; \theta), \theta \in \Theta$$

满足极大似然估计的参数表示为：

$$\hat{\theta} = argmaxL(\theta)$$

2. Jensen 不等式

设 f 是定义域为实数的函数，如果对所有的实数 x，f(x)的二阶导数都大于 0，那么 f 是凸函数。Jensen 不等式定义如下：

如果 f 是凸函数，X 是随机变量，那么：

$$E[f(X)] \geq f(E[X])$$

模型：

使用 EM 算法，通过 E-step 与 M-step 收敛得到模型参数估计。分析输入 $x = (x_1, x_2, ..., x_n)$

可以得到：联合分布 $p(x, z; \theta)$，条件分布 $p(z \mid x, \theta)$ 。对于分布 Q 与隐含数据 z 有：

E-step：

$$Q_i(z_i) = p(z_i|x_i, \theta_j)$$

$$l(\theta, \theta_j) = \sum_{i=1}^{n} \sum_{z_i} Q_i(z_i) log \frac{p(x_i, z_i; \theta)}{Q_i(z_i)}$$

M-step：

$$\theta_{j+1} = argmaxl(\theta, \theta_j)$$

程序实验：

1.环境

Anaconda3 python3.9.16

2.实验过程

使用 csv_reader 按行读取全部数据，并对估计的参数给定初值

```
# 数据初始化
pi = 0.7
u1 = 175
u2 = 160
Sig1 = 20
Sig2 = 20
height=[]
```

```
# 读取文件中数据
with open('./height_data.csv', encoding="utf8") as f:
    csv_reader = csv.reader(f)
    # skip the header
    next(csv_reader)
    for line in csv_reader:
      #np.array(height)=int(line[2])
      height.append(line[0])
      i=i+1
```

E-step：得到数组 P

```
for cnt in range(0,N):

    PD1 = stats.norm.pdf(H[cnt],u1,Sig1)
    PD2 = stats.norm.pdf(H[cnt],u2,Sig2)

    P[cnt] = pi*PD1/(pi*PD1+(1-pi)*PD2)
```

M-step：更新均值与标准差

```
Psum = np.sum(P)
Pxsum1 = np.dot(P,H)
Pxsum2 = np.dot((1-P),H)

pi = Psum/N

u1 = Pxsum1/Psum
u2 = Pxsum2/(N-Psum)
#两个分子
numerator1 = np.sum(np.dot(P,np.square(H-u1)))
numerator2 = np.sum(np.dot((1-P),np.square(H-u2)))

Sig1 = np.sqrt(numerator1/Psum)
Sig2 = np.sqrt(numerator2/(N-Psum))
```

使用估计参数预测样本性别，按照估计的参数进行正态分布样本预测，比较样本属于男性和女性的概率，选择概率较大者为预测结果：

```
correct=0
for n in range(0,N):
    pre1=stats.norm.pdf(H[n],u1,Sig1)
    pre2=stats.norm.pdf(H[n],u2,Sig2)
    if pre1>=pre2:
        gender_pre='M'
    else:
        gender_pre='F'
    if n<=500-1:
        gender_real='F'
    else:
        gender_real='M'
    if gender_pre == gender_real:
        correct=correct+1
print("accuracy:",correct/2000)
```

3.实验结果与评估

```
number: 2000 mean2: 176.24334102229437 mean1: 164.2889194716009 std2: 5.095290774789345 std1: 3.2183520381756723 type choosing: 0.74
93561851231751
accuracy: 0.922
```

	mean1	std1	mean2	std2	pi
真值	164	3	176	5	0.75
EM 估计	164.289	3.218	176.243	5.095	0.7494
相对误差	0.17%	7.3%	0.14%	1.9%	-0.08%

在参数估计方面，各参数相对误差均较小，在估计均值与选择概率方面相对误差较小，达到0.2%以内，在标准差估计方面相对误差相对较大，但基本准确。说明 EM 算法能够较为准确的实现 GMM 参数估计。

在性别预测方面，预测准确率达到了 92.2%，较为准确的预测了给定样本的性别所属。

结论:

经过多次迭代收敛，EM 算法能够的完成混合模型的参数估计，且数据与真值具有较小的相对误差，同时基于实验估计的参数预测给定样本性别，并达到了较高的预测成功率，证明了算法的准确性。

参考资料:

[1] 混合高斯分布与其参数估计 - 知乎 (zhihu.com)
[2] 人工智能课程讲义