基于 EM 算法的高斯混合模型参数估计

ZY2207304 范想明 xmfan@buaa.edu.cn

摘要

本文针对一组一维身高数据组成的高斯混合模型，分析了其模型特征，并基于 EM 算法对该高斯混合模型进行参数估计。通过实验得到了高斯混合模型的参数的估计值，参数估计值与生成模型的原始参数较为接近，同时基于算法得到的参数生成的拟合曲线与原始数据具有良好的贴合性。

1. 背景介绍

高斯模型可简单分为单高斯模型（SGM）和高斯混合模型（GMM）两种，其中单高斯模型也就是我们平时说所的高斯分布(正态分布)，概率密度函数服从高斯分布的模型叫做单高斯模型。

高斯混合模型（GMM）是单高斯模型（SGM）的延伸，就是用多个高斯概率密度函数（正态分布曲线）精确地量化变量分布，是将变量分布分解为若干基于高斯概率密度函数（正态分布曲线）分布的统计模型。它可以平滑的近似任何形状的分布。反过来解释就是，K 个单高斯模型混合在一起，生成的模型，就是高斯混合模型，它是一个参数化的分布族，由多个高斯分布的线性组合构成。每个高斯分布表示一个“成分”，因此，GMM 通常被用来解决聚类问题。

EM 算法是一种求解包含隐变量（潜在变量）的概率模型参数的算法。EM 算法分为两步：E 步和 M 步。在 E 步中，通过当前模型参数和数据，计算隐变量的后验分布；在 M 步中，根据隐变量的后验分布，更新模型参数。在 GMM 中，E 步计算每个样本属于每个高斯分布的概率（即后验分布），M 步则根据后验分布更新高斯分布的均值、方差和权重。EM 算法的迭代过程是逐步逼近真实参数的过程，直到满足收敛条件。

2. 研究方法

对于单高斯模型，当样本数据 x 为一维数据时，高斯模型的概率密度函数为：

$$p(x;\mu,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

其中：μ 为数据的均值，σ 为数据的标准差。

对于混合高斯分布，当用 z 表示样本数据来自于哪个子分布，则 GMM 的概率密度函数可被表示为：

$$p(x;\theta)=\Sigma_z p(x,z;\theta)=\Sigma_k p(z=k;\theta)p(x;z=k,\theta)=\sum_{k=1}^{K}\alpha_k\phi(x;\theta_k) \tag{2}$$

其中：α_k 为样本数据属于第 k 个子模型的概率，且 $\sum_{k=1}^{K}\alpha_k=1$；

$\phi(x|\theta_k)$ 为第 k 个单高斯子模型的概率密度函数，其中 $\theta_k=(\mu_k,\sigma_k)$。

对于 EM 算法，当样本数据 $\{x_1,x_2,...,x_n\}$ 是由 n 个独立的样本构成，则当对概率密度函数 $p(x,z)$进行参数估计时，它的极大似然函数为：

$$L(\theta)=\sum_{i=1}^{n}\log p(x;\theta)=\sum_{i=1}^{n}\log(\sum_{k=1}^{K}\alpha_k\phi(x;\theta_k)) \tag{3}$$

该函数无法通过常规的求偏导方法进行求解，EM 算法通过逐步迭代的方法进行求解。在 EM 算法的每一次迭代中，包含以下两个步骤：

1) E 步：基于已有的 θ 值（初次为人为给定，后续由 M 步更新），猜测估计隐藏变量；
2) M 步：根据 E 步求得的隐藏变量，计算新一轮迭代的模型参数 θ。

重复计算 E 步和 M 步直至收敛（$|\theta_{i+1}-\theta_i|<\varepsilon$，$\varepsilon$ 是一个很小的正数，表示经过一次迭代之后参数变化非常小）。至此，我们就找到了高斯混合模型的参数。需要注意的是，EM 算法具备收敛性，但并不保证找到全局最大值，有可能找到局部最大值。解决方法是初始化几次不同的参数进行迭代，取结果最好的那次。

3. 实验过程与结果

首先分析实验数据：

```
# 读取csv文件
df = pd.read_csv('height_data.csv', header=0)
data = df['height'].values.ravel()

plt.figure()
plt.hist(data, bins=35, rwidth=0.9, label='Sample data')
plt.xlabel('Height (cm)')
plt.ylabel('Count')
plt.title('Distribution of Heights')
plt.legend()
plt.show()
```

将其绘制成频率直方图如图 1 所示。

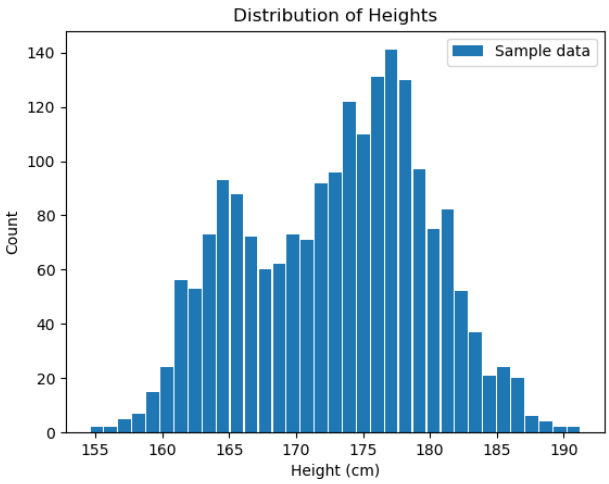

图 1 原始数据的频率直方图

易看出其具有两个峰值，即可认为其是一个二维高斯混合模型。由此，可给定算法的初值为：

- 模型 1：权重 0.5，均值 165，标准差 5；
- 模型 2：权重 0.5，均值 178，标准差 5。

```
# 初始化模型参数
weights = np.array([0.5, 0.5])
means = np.array([[165.0], [178.0]])
standard_deviation = np.array([[[5.0]], [[5.0]]])
n_components = 2
n_features = 1
n_samples = len(data)
max_iter = 1000
tolerance = 1e-6
```

而后，根据 EM 算法，基于初始模型参数进行迭代：

```
# EM算法的E步和M步
for i in range(max_iter):
    # E步
    likelihood = np.zeros((n_samples, n_components))
    for k in range(n_components):
        likelihood[:, k] = weights[k] * gaussian_pdf(data, means[k],
standard_deviation[k])
    posterior = likelihood / np.sum(likelihood, axis=1,
keepdims=True)

    # M步
```

```
    weights = np.mean(posterior, axis=0)
    means = np.sum(data.reshape(-1, 1) * posterior, axis=0) / \
        np.sum(posterior, axis=0)
    standard_deviation = np.sqrt(np.sum((data.reshape(-1, 1) - \
means)**2 * \   posterior, axis=0) / np.sum(posterior, axis=0))

    # 计算对数似然函数值，检查收敛性
    log_likelihood = np.sum(np.log(np.sum(likelihood, axis=1)))
    if i > 0 and np.abs(log_likelihood - prev_log_likelihood) <
tolerance:
        break
    prev_log_likelihood = log_likelihood
```

最后输出实验结果如图 2 所示。

```
weights: [0.27130672 0.72869328]
means: [164.44808771 176.25192443]
standard deviation: [2.93153856 4.7362253 ]
```

图 2 参数估计结果

即：两个子模型的参数为：

- 模型 1：权重 0.2713，均值 164.45，标准差 2.93；
- 模型 2：权重 0.7287，均值 176.25，标准差 4.74。

将计算得到的模型与原始模型进行对比如图 3 所示，可以看出，通过估计得到的参数拟合的曲线与原始数据具有很好的贴合性。

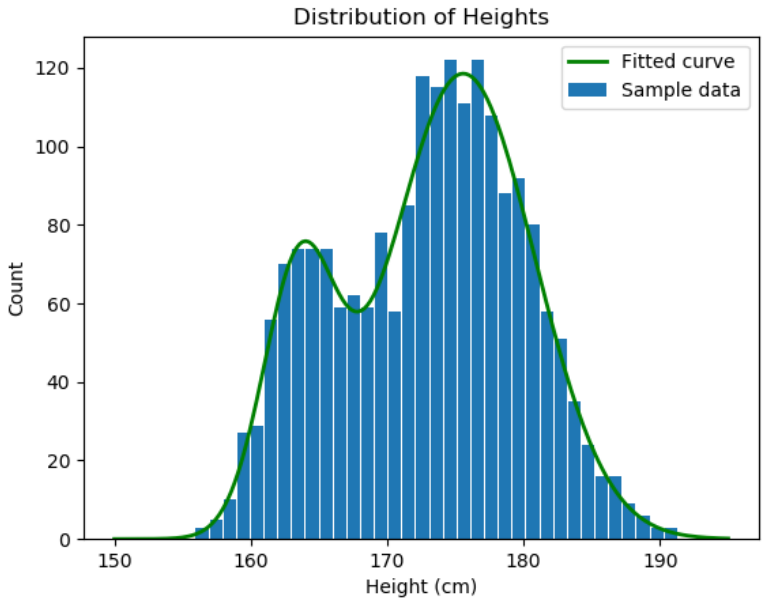

图 3 身高的分布中样本数据与拟合曲线的对比