Homework 03 MoG

王晓捷 (11521053)

May 18, 2016

1 引言

混合高斯分布（MoG）是一种无监督学习算法，常用于聚类。当聚类问题中各个类别的尺寸不同、聚类间有相关关系的时候，往往使用MoG更为合适。对一个样本来说，MoG得到的是其属于各个类的概率，而不是完全的属于某个类，这种聚类方法被称为软聚类。

在MoG问题中，数据属于哪个分布可以看成是一个隐含变量z。MoG模型中存在两个假设[1]：

- 假设1: z服从多项式分布，即：
$$z^{(i)} \sim Multinomial(\phi) \tag{1.1}$$
其中，$\sum_j \phi_j = 1$。
- 假设2：已知z时，x服从正态分布，即条件概率$p(x|z)$服从正态分布，即：
$$p(x^{(i)}|z^{(i)} = j) \sim N(\mu_j, \Sigma_j) \tag{1.2}$$

由以上两个假设可以得到联合概率分布
$$p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)}) \tag{1.3}$$
由上面公式可知，MoG模型的参数即为ϕ，μ和Σ。为了求解出这3个参数，写出其对应的似然函数，如下：
$$\begin{aligned} l(\phi, \mu, \Sigma) &= \sum_{i=1}^{m} logp(x^{(i)}; \phi, \mu, \Sigma) \\ &= \sum_{i=1}^{m} logp(x^{(i)}|z^{(i)}\mu, \Sigma)p(z^{(i)}; \phi) \end{aligned} \tag{1.4}$$
然而，无法使用普通的求偏导的方式来求解$\max l(\phi, \mu, \Sigma)$来获得$\phi$,$\mu$,$\Sigma$。如果我们知道$z^{(i)}$的值的话，就可以求出其偏导。但是，我们现在并不知道$z^{(i)}$的值，可以使用EM算法进行迭代估计出$z^{(i)}$从而得到参数。

2 方法概述

EM算法只包含两步，其基本思想如下[1]：

(a) 设置初始参数：ϕ,μ,Σ

(b) E-step：根据当前参数与观测数据x，估计隐含变量z的分布
$$\begin{aligned} w_j^{(i)} &:= p(z^{(i)} = j|x^{(i)}, \phi, \mu, \Sigma) \\ &:= \frac{p(x^{(i)}|z^{(i)} = j; \mu, \Sigma)p(z^{(i)} = j; \phi)}{\sum_k p(x^{(i)}|z^{(i)} = k; \mu, \Sigma)p(z^{(i)} = k; \phi)} \end{aligned} \tag{2.1}$$

(c) M-step: 根据z的分布，对μ,Σ进行重新估计
$$\phi_j := \frac{1}{m} \textstyle\sum_{i=1}^{m} w_j^(i) \tag{2.2}$$
$$\mu_j := \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}} \tag{2.3}$$

$$\Sigma_j := \frac{\sum_{i=1}^{m} w_j^{(i)}(x^{(i)} - \mu)(x^{(i)} - \mu)^T}{\sum_{i=1}^{m} w_j^{(i)}} \tag{2.4}$$

(d) 第b步和第c步反复进行，直到参数变化小于阈值或者目标函数的变化小于阈值为止。

3 实验结果

实验假设该混合高斯模型由两个分量组成，每个分量的维度为2维。实验结果如图1 和图2 所示。图1 是针对两个高斯分量的真实参数，图2是由EM算法求解出的参数。

EM算法迭代次数为26次。

```
Real phi: [ 0.6  0.4]
 Real mean: [[ 1  5]
             [-3  4]]
 Real covarance: [[[ 3.   0. ]
                   [ 0.   0.5]]

                  [[ 1.   1. ]
                   [ 1.   2. ]]]
```

图 1: 混合高斯真实参数

```
phi by em: [ 0.61103149  0.38896851]
mean by em: [[ 0.92955879  4.98749395]
             [-3.00668717  4.02840147]]
covarance by em: [[[ 3.03089748 -0.01085535]
                   [-0.01085535  0.4995882 ]]

                  [[ 1.06544534  1.148074  ]
                   [ 1.148074    2.14998166]]]
```

图 2: 由EM求解得到的参数

4 小结与讨论

从第3部分可以看出，EM算法求解出的参数十分接近真实值。

参考文献

[1] Andrew Ng. Cs229 lecture notes. CS229 Lecture notes, 1(1):1–3, 2000.