期望最大化算法

binary_seach

December 3, 2012

Contents

1 简介

期望最大化 (Expectation Maximization) 算法最初是由 Ceppellini[2] 等人 1950 年在讨论基因频率的估计的时候提出的。后来又被 Hartley[3] 和 Baum[4] 等人发展的更加广泛。目前引用的较多的是 1977 年 Dempster[5] 等人的工作。它主要用于从不完整的数据中计算最大似然估计。后来经过其他学者的发展，这个算法也被用于聚类等应用。

2 一个例子 [6]

本章节希望通过一个例子来解释期望最大化算法的来由以及合理性。

考虑一个投掷硬币的实验：现在我们有两枚硬币 A 和 B，这两枚硬币和普通的硬币不一样，他们投掷出正面的概率和投掷出反面的概率不一定相同。我们将 A 和 B 投掷出正面的概率分别记为 θ_A 和 θ_B。我们现在独立地做 5 次试验：随机的从这两枚硬币中抽取 1 枚，投掷 10 次，统计出现正面的次数。那么我们就得到了如表格1的实验结果。

试验代号	投掷的硬币	出现正面的次数
1	B	5
2	A	9
3	A	8
4	B	4
5	A	7

Table 1: 硬币投掷实验的结果

在这个实验中，我们记录两组随机变量 $X = (X_1, X_2, X_3, X_4, X_5), Z = (Z_1, Z_2, Z_3, Z_4, Z_5)$，其中 $X_i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ 代表试验 i 中出现正面的次数，$Z_i \in \{A, B\}$ 代表这次试验投掷的是硬币 A 还是硬币 B。

我们的目标是通过这个实验来估计 $\theta = (\theta_A, \theta_B)$ 的数值。这个实验中的参数估计就是**有完整数据的参数估计**，这个是因为我们不仅仅知道每次试验中投掷出正面的次数，我们还知道每次试验中投掷的是硬币 A 还是 B。

一个很简单也很直接的估计 θ 的方法如公式1所示。

$$\begin{aligned} \hat{\theta_A} &= \frac{\text{用硬币 A 投掷出正面的次数}}{\text{用硬币 A 投掷的次数}} \\ \hat{\theta_B} &= \frac{\text{用硬币 B 投掷出正面的次数}}{\text{用硬币 B 投掷的次数}} \end{aligned} \tag{1}$$

实际上这样的估计就是统计上的极大似然估计 (maximum likelihood estimation) 的结果。用 $P(X, Z|\theta)$ 来表示 X,Z 的联合概率分布（其中带有参

数 θ)，那么对于上面的实验，我们可以计算出他们出现我们观察到的结果即 $x^0 = (5, 9, 8, 4, 7), z^0 = (B, A, A, B, A)$ 的概率2。函数 $P(X = x^{(0)}, Z = z^{(0)}|\theta)$ 就叫做 θ 的似然函数。我们将它对 θ 求偏导并令偏导数为 0，就可以得到如1 的结果。

$$\begin{aligned} P(X = x^{(0)}, Z = z^{(0)}|\theta) = C_5^3 P(Z = A)^3 (1 - P(Z = A))^2 \\ \times C_{10}^9 \theta_A^9 (1 - \theta_A) \\ \times C_{10}^8 \theta_A^8 (1 - \theta_A)^2 \\ \times C_{10}^7 \theta_A^7 (1 - \theta_A)^3 \\ \times C_{10}^5 \theta_B^5 (1 - \theta_B)^5 \\ \times C_{10}^4 \theta_B^4 (1 - \theta_B)^6 \end{aligned} \tag{2}$$

我们将这个问题稍微改变一下，我们将我们所观察到的结果修改一下：我们现在只知道每次试验有几次投掷出正面，但是不知道每次试验投掷的是哪个硬币，也就是说我们只知道表1中第一列和第三列。这个时候我们就称 Z 为**隐藏变量 (Hidden Variable)**，X 称为**观察变量 (Observed Variable)**。这个时候再来估计参数 θ_A 和 θ_B，就没有那么多数据可供使用了，这个时候的估计叫做**不完整数据的参数估计**。

如果我们这个时候有某种方法（比如，正确的猜到每次投掷硬币是 A 还是 B)，这样的话我们就可以将这个不完整的数据估计变为完整数据估计。

当然我们如果没有方法来获得更多的数据的话，那么下面提供了一种在这种不完整数据的情况下来估计参数 θ 的方法。我们用迭代的方式来进行：

(1) 我们先赋给 θ 一个初始值，这个值不管是经验也好猜的也好，反正我们给它一个初始值。在实际使用中往往这个初始值是有其他算法的结果给出的，当然随机给他分配一个符合定义域的值也可以。这里我们就给定 $\theta_A = 0.7, \theta_B = 0.4$

(2) 然后我们根据这个来判断或者猜测每次投掷更像是哪枚硬币投掷的结果。比如对于试验 1，如果投掷的是 A，那么出现 5 个正面的概率为 $C_1 0^5 \times 0.7^5 \times (1-0.7)^5 \approx 0.1029$；如果投掷的是 B，出现 5 个正面的概率为 $C_1 0^5 \times 0.4^5 \times (1-0.4)^5 \approx 0.2007$；基于试验 1 的试验结果，可以判断这个试验投掷的是硬币 A 的概率为 0.1029/(0.1029+0.2007)=0.3389，是 B 的概率为 0.2007/(0.1029+0.2007)= 0.6611。因此这个结果更可能是投掷 B 出现的结果。

(3) 假设上一步猜测的结果为 B,A,A,B,A，那么根据这个猜测，可以像完整数据的参数估计一样 (公式2) 重新计算 θ 的值。

这样一次一次的迭代 2-3 步骤直到收敛，我们就得到了 θ 的估计。现在你可能有疑问，这个方法靠谱么？事实证明，它确实是靠谱的。

期望最大化算法就是在这个想法上改进的。它在估计每次投掷的硬币的时候，并不要确定住这次就是硬币 A 或者 B，它计算出来这次投掷的硬币是 A 的概率和是 B 的概率；然后在用这个概率（或者叫做 Z 的分布）来计算似然函数。期望最大化算法步骤总结如下：

E 步骤 先利用旧的参数值 θ' 计算隐藏变量 Z 的（条件）分布 $P_{\theta'}(Z = z_j|X_i = x_i)$，然后计算 $logP_\theta(Z, X = x)$[1]的期望

$$E(logP_\theta(Z, X = x)) = \sum_{i=1}^{n}\sum_{j} P_{\theta'}(Z = z_j|X_i = x_i)logP_\theta(Z = z_j, X_i = x_i) \tag{3}$$

其中 θ 是当前的 θ 值，而 θ' 是上一次迭代得到的 θ 值。公式3中已经只剩下 θ 一个变量了，θ' 是一个确定的值，这个公式或者函数常常叫做 Q 函数，用 $Q(\theta, \theta')$ 来表示。

M 步骤 极大化 Q，往往这一步是求导，得到由旧的 θ 值 θ' 来计算新的 θ 值的公式

$$\frac{\partial Q}{\partial \theta} = 0 \tag{4}$$

总结一下，期望最大化算法就是先根据参数初值估计隐藏变量的分布，然后根据隐藏变量的分布来计算观察变量的似然函数，估计参数的值。前者通常称为 E 步骤，后者称为 M 步骤。

3 数学基础 [7]

首先来明确一下我们的目标：我们的目标是在观察变量 X 和给定观察样本 $x_1, x_2, ..., x_n$ 的情况下，极大化对数似然函数

$$l(\theta) = \sum_{i=1}^{n} \ln P(X_i = x_i) \tag{5}$$

其中只包含观察变量的概率密度函数

$$P(X_i = x_i) = \sum_{j} P(X_i = x_i, Z = z_j) \tag{6}$$

[1]这里因为参数 θ 的写法与条件概率的写法相同，因此将参数 θ 写到下标以更明确的表述

其中 Z 为隐藏随机变量,$\{z_j\}$ 是 Z 的所有可能的取值。那么

$$\begin{aligned} l(\theta) &= \sum_{i=1}^{n} \ln \sum_{j} P(X_i = x_i, Z = z_j) \\ &= \sum_{i=1}^{n} \ln \sum_{j} \alpha_j \frac{P(X_i = x_i, Z = z_j)}{\alpha_j} \end{aligned} \tag{7}$$

这里我们引入了一组参数（不要怕多，我们后面会处理掉它的）α，它满足 $\forall$可能的$j, \alpha_j \in (0, 1]$ 和 $\sum_j \alpha_j = 1$ 到这里，先介绍一个凸函数的性质，或者叫做凸函数的定义。$f(x)$ 为凸函数，$\forall i = 1, 2, ..., n, \lambda_i \in [0, 1], \sum_{i=1}^{n} \lambda_i = 1$，对 $f(x)$ 定义域中的任意 n 个 $x_1, x_2, ..., x_n$ 有

$$f(\sum_{i=1}^{n} \lambda_i x_i) \leq \sum_{i=1} n \lambda_i f(x_i))^2 \tag{8}$$

对于严格凸函数，上面的等号只有在 $x_1 = x_2 = ... = x_n$ 的时候成立。关于凸函数的其他性质不再赘述。对数函数是一个严格凸函数。因而我们可以有下面这个结果：

$$\begin{aligned} l(\theta) &= \sum_{i=1}^{n} \ln \sum_{j} \alpha_j \frac{P(X_i = x_i, Z = z_j)}{\alpha_j} \\ &\geq \sum_{i=1}^{n} \sum_{j} \alpha_j \ln \frac{P(X_i = x_i, Z = z_j)}{\alpha_j} \end{aligned} \tag{9}$$

现在我们根据等号成立的条件来确定 α_j 即

$$\frac{P(X_i = x_i, Z = z_j)}{\alpha_j} = c \tag{10}$$

其中 c 是一个与 j 无关的常数。因为 $\sum_j \alpha_j = 1$，稍作变换就可以得到

$$\alpha_j = \frac{P(X_i = x_i, Z = z_j)}{P(X_i = x_i)} \tag{11}$$

现在来解释一下我们得到了什么。α_j 就是 $Z = z_j$ 在 $X = x_i$ 下的条件概率或者后验概率。求 α 就是求隐藏随机变量 Z 的条件分布。总结一下目前得到的公式就是

$$l(\theta) = \sum_{i=1}^{n} \sum_{j} \alpha_j \ln \frac{P(X_i = x_i, Z = z_j)}{\alpha_j} \tag{12}$$

直接就极大值比较难求，EM 算法就是按照下面这个过程来的。

[2]它就是大名鼎鼎的琴生（Jensen）不等式

(1) 根据上一步的 θ 来计算 α，即隐藏变量的条件分布

(2) 极大化似然函数来得到当前的 θ 的估计

3.1 极大似然估计 [1]

好吧，我觉得还是再说说极大似然估计吧。给定一个概率分布 D，假设其概率密度函数为 f，其中 f 带有一组参数 θ。为了估计这组参数 θ，我们可以从这个分布中抽出一个具有 n 个采样值的 $X_1, X_2, ..., X_n$，那么这个就是 n 个（假设独立）同分布随机变量，他们分别有取值 $x_1, x_2, ..., x_n$，那么我们就可以计算出出现这样一组观察值的概率密度为

$$l(\theta) = \prod_{i=1}^{n} f(x_i) \tag{13}$$

对于 f 是离散的情况，就计算出现这一组观察值的概率。

$$l(\theta) = \prod_{i=1}^{n} P(X_i = x_i) \tag{14}$$

注意，这个函数中是含有参数 θ 的。θ 的极大似然估计就是求让上面似然函数取极大值的时候的参数 θ 值。

一般来说，会将上面那个似然函数取自然对数，这样往往可以简化计算。记住，这样仅仅是为了简化计算[3]。取了自然对数之后的函数叫做对数似然函数。

$$\ln l(\theta) = \sum_{i=1}^{n} \ln f(x_i) \tag{15}$$

因为对数是一个严格单调递增的凹函数[4]，所以对似然函数取极大值与对对数似然函数取极大值是等价的。

[3]取了对数之后还可以跟信息熵等概念联系起来

[4]关于凸函数有很多种说法，上凸函数和下凸函数，凸函数和凹函数等等，这里指的是二阶导数大于（等于）0 的一类函数，而凹函数是其相反数为凸函数的一类函数

3.2 期望最大化算法收敛性

如何保证算法收敛呢？我们只用证明 $l(\theta^{(t+1)}) \geq l(\theta^{(t)})$ 就可以了。

$$\begin{aligned} l(\theta^{(t+1)}) &= \sum_{i=1}^{n} \sum_{j} \alpha_j^{(t+1)} \ln \frac{P(X = x_i, Z = z_j)^{(t+1)}}{\alpha_j^{(t+1)}} \\ &\geq \sum_{i=1}^{n} \sum_{j} \alpha_j^{(t)} \ln \frac{P(X = x_i, Z = z_j)^{(t+1)}}{\alpha_j^{(t)}} \\ &\geq \sum_{i=1}^{n} \sum_{j} \alpha_j^{(t)} \ln \frac{P(X = x_i, Z = z_j)^{(t)}}{\alpha_j^{(t)}} \\ &= l(\theta^{(t)}) \end{aligned} \tag{16}$$

其中第一个大于等于号是因为只有当 α 取值合适（琴生不等式等号成立条件）的时候才有等号成立，第二个大于等于号正是 M 步骤的操作所致。

这样我们就知道 $l(\theta)$ 是随着迭代次数的增加越来越大的，收敛条件是值不再变化或者变化幅度很小。

4 应用举例

4.1 参数估计

很直接的应用就是参数估计，上面举的例子就是参数估计。

4.2 聚类

但是如果估计的参数可以表明类别的话，比如某个参数表示某个样本是否属于某个集合。这样的话其实聚类问题也就可以归结为参数估计问题。

References

[1] 最大似然估计 [Online]. Available:http://zh.wikipedia.org/wiki/%E6%9C%80%E5%A4%A7%E4%BC%BC%E7%84%B6%E4%BC%B0%E8%AE%A1

[2] Ceppellini, r., Siniscalco, M. & Smith, C.A. Ann. Hum. Genet. 20, 97–115 (1955).

[3] Hartley, H. Biometrics 14, 174–194 (1958).

[4] Baum, L.E., Petrie, T., Soules, G. & Weiss, N. Ann. Math. Stat. 41, 164–171 (1970).

[5] Dempster, A.P.; Laird, N.M.; Rubin, D.B. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society. Series B (Methodological) 39 (1): 1–38. JSTOR 2984875. MR 0501537.

[6] What is the expectation maximization algorithm? [Online]. Avaiable:http://ai.stanford.edu/~chuongdo/papers/em_tutorial.pdf

[7] The EM Algorithm [Online]. Available:http://www.cnblogs.com/jerrylead/archive/2011/04/06/2006936.html