# Lecture 10: Expectation-Maximization (EM) 方法

张伟平

Monday 16$^{\text{th}}$ November, 2009

# Contents

# Chapter 1
# EM optimization method

期望-最大化(EM)算法是一种在观测到数据后, 估计未知参数的迭代优化方法. 其 能非常简单地执行并且能够通过稳定,上升的步骤非常可靠地找到全局最优值. 对EM方法详细介绍请参考非常好的教材 McLachlan and Krishnan, 1997, EM algorithm and extensions, Wiley.

在频率概率框架下, 我们可以认为:除了观测到的样本$X$外, 还有一些与之伴随的未知的 缺失量(missing)或者没有观察到的量$Z$. 那么, 完整的样本应该是$Y = (X, Z)$. 问题的目的 是在得到观测的样本$x$后, 最大化似然函数$L(\theta|x) = \int f(x, z|\theta)dz$. 此似然函数 由于涉及到边际概率函数一般难以处理, 而采用$f(x, z|\theta)$和$f(z|x, \theta)$则可能容易处理, EM算法就是通过采用这些较容易的密度而避免直接考虑$L(\theta|x)$.

另外, 在Bayes理论框架下, 感兴趣的通常是对后验概率函数$f(\theta|x)$最大化, 以后验众数(似然思想)来估计参数. 此时, 最优化问题有时可以通过考虑引入一个未观测的参数$\psi$而得到简化.

有时候, 缺失数据$Z$并非真正的缺失了, 它们可能仅仅是为了简化我们的问题而引入的变量. 此时, $Z$常常被称为 隐变量(latent variable).

## 1.1 EM algorithm

记可观测的量为$X$, 缺失量为$Z$, 完全数据为$Y = (X, Z)$, 待估参数为$\theta$. 则我们有的信息是观测数据 的似然$L(\theta|x)$, 最大化此似然或者说是求$\theta$的极大似然估计是我们的目标. EM算法通过迭代方式来寻求最大化$L(\theta|x)$的解. 假设$\theta^{(t)}$表示在第$t$次迭代后的最大值点, $t = 0, 1, 2, \cdots$. 定义$Q(\theta|\theta^{(t)})$为在观测到$X = x$,以及在参数$\theta = \theta^{(t)}$的条件下 完全数据的联合对数似然函数的期望, 此期望是对$f_{Z|X}(z|x, \theta^{(t)})$计算. 即

$$Q(\theta|\theta^{(t)}) \quad = \quad E\{logL(\theta|Y)|x, \theta^{(t)}\}$$

$$= \quad E\{logf(x,Z|\theta)|x,\theta^{(t)}\}$$
$$= \quad \int logf(x,z|\theta)f(z|x,\theta^{(t)})dz$$

最后一式强调一旦我们给定了$X=x$, $Z$就是$Y$唯一的随机部分.

EM算法从$\theta^{(0)}$开始, 然后在两步之间交替: E表示期望, M表示最大化. 该算法概括如下:

(1) E步: 计算$Q(\theta|\theta^{(t)}) = E[L(\theta|Y)|x,\theta^{(t)}]$.

(2) M步: 关于$\theta$最大化$Q(\theta|\theta^{(t)})$, 并记$\theta^{(t+1)}$表示此时的最大值点.

(3) 返回到E步, 直至收敛准则达到.

**例1 椒花蛾(peppered moth)** 一个进化和工业污染的例子.

白桦尺蛾(peppered moth),鳞翅目(Lepidoptera)尺蛾科(Geometridae)昆虫, 学名Biston betularia. 翅黑或白色, 上有斑点, 分布欧洲. 其生命周期有四个阶段: 卵, 毛毛虫, 蛹, 成虫. 在由蛹变为成虫的时刻, 它们 会停留在树干上完成此过程, 此时就会被鸟类捕食. 1848年, 在英国曼彻斯特 首次注意到黑色型, 到1898年黑色型反而以99:1的比例超过了淡色型. 对这种现象的解释是: 工业区的树干被煤烟染黑, 黑色的蛾栖于树上, 目标不显著, 不易为鸟类捕食. 这是通过工业黑化现象进行自然选择的一个例子.

——百度百科

这种蛾子的色彩已经被确认是通过某单个基因决定的, 该基因有三个可能的等位基因: C, I和T. 其中 C对I是显性的, T对I是隐性的. 因此基因型CC, CI和CT导致(深色)黑色的表型. 基因型TT导致浅色 的表型. II和IT导致一种中间的表型, 外观上变化很广泛, 但通常以中间色彩杂色而成. 因此有6种 基因型, 3种表现型.

在英国和北美, 在燃煤的工业区, 浅色的蛾子几乎以及被深色的蛾子所替代. 这种等位基因频率在种群 内的变化被引为是在人类时间刻度下可以观察到微进化的一个例子. (被试验证实的)理论结果是”鸟类 在不同反射背景下对蛾子的捕食程度明显不同”. 在燃煤的工业区, 污染减弱了黑色表型蛾子栖息

在树皮表面的反射程度, 因而对其有利. 当环境改善后, 浅色表型增加, 黑色表型减少, 这并不令人奇怪.

因此, 监视这种蛾子的等位基因C,I,T的变化, 除了可以研究微进化过程外, 也可以用于环境污染监控. 假设Hardy-Weinberg平衡律成立, 记各个等位基因在种群中的频率为$p_C, p_I, p_T (p_C + p_I + p_T = 1)$, 则基因型CC,CI,CT,II,IT,TT的 频率分别为$p_C^2, 2p_Cp_I, 2p_Cp_T, p_I^2, 2p_Ip_T, p_T^2$.

假设我们随机捕了$n$只蛾子, 其中黑色表型的有$n_C$只, 中间表型的有$n_I$只, 浅色表型的有$n_T$只. 但是各个基因型的频数不可观测:

| | CC | CI | CT | II | IT | TT |
|---|---|---|---|---|---|---|
| 不可观测 | $n_{CC}$ | $n_{CI}$ | $n_{CT}$ | $n_{II}$ | $n_{IT}$ | $n_{TT}$ |
| 观测 | $n_C$ 黑色 | | | $n_I$ 中间 | | $n_T$ 浅色 |

因此, 此时观测数据为$x = (n_C, n_I, n_T)$, 而完全数据为$y = (n_{CC}, n_{CI}, n_{CT}, n_{II}, n_{IT}, n_{TT})$, 这里$n_{CC} + n_{CI} + n_{CT} = n_C$, $n_{II} + n_{IT} = n_I$, 以及$n_{TT} = n_T$. 我们希望由此数据 来估计各个等位基因的概率$p_C, p_I, p_T$.

从而完全数据的似然函数为

$$\begin{aligned} logf(y|p) &= n_{CC}logp_C^2 + n_{CI}log(2p_Cp_I) + n_{CT}log(2p_Cp_T) \\ &+ n_{II}log(p_I^2) + n_{IT}log(2p_Ip_T) + n_{TT}log(p_{TT}^2) \\ &+ log\frac{n!}{n_{CC}!n_{CI}!n_{CT}!n_{II}!n_{IT}!n_{TT}!} \end{aligned}$$

由于完全数据不可观测, 若记$N_{CC}, N_{CI}, N_{CT}, N_{II}, N_{IT}, N_{TT}$分别为各个基因 型的个数, $p = (p_{CC}, p_{CI}, p_{CT}, p_{II}, p_{IT}, p_{TT})$, 则有

1. $N_{CC}, N_{CI}, N_{CT}|n_C, n_I, n_T, p \sim$
$$MN\left(n_C, \frac{(p_C)^2}{1-(p_I+p_T)^2}, \frac{2p_Cp_I}{1-(p_I+p_T)^2}, \frac{2p_Cp_T}{1-(p_I+p_T)^2}\right)$$
2. $N_{II}, N_{IT}|n_C, n_I, n_T, p \sim$
$$MN\left(n_I, \frac{(p_I)^2}{2p_Cp_I+(p_T)^2}, \frac{2p_Cp_I}{2p_Cp_I+(p_T)^2}\right)$$
3. $N_{TT}|n_C, n_I, n_T, p \sim B(n_T, p_T^2)$

从而在EM算法的第$t$步中, 对完全似然函数取期望得到

$$\begin{aligned} Q(p|p^{(t)}) &= n_{CC}^{(t)} log(p_C^2) + n_{CI}^{(t)} log(log(2p_C p_I) + n_{CT}^{(t)} log(2p_C p_T) \\ &+ n_{II}^{(t)} log(p_I^2) + n_{IT}^{(t)} log(2p_I p_T) + n_{TT}^{(t)} log(p_{TT}^2) \\ &+ k(n_C, n_I, n_T, p^{(t)}), \end{aligned}$$

其中

$$\begin{aligned} n_{CC}^{(t)} &= E\{N_{CC}|n_C, n_I, n_T, p^{(t)}\} = n_C \frac{(p_C^{(t)})^2}{1-(p_I^{(t)}+p_T^{(t)})^2}, \\ n_{CI}^{(t)} &= E\{N_{CC}|n_C, n_I, n_T, p^{(t)}\} = n_C \frac{2p_C^{(t)} p_I^{(t)}}{1-(p_I^{(t)}+p_T^{(t)})^2}, \\ n_{CT}^{(t)} &= E\{N_{CC}|n_C, n_I, n_T, p^{(t)}\} = n_C \frac{2p_C^{(t)} p_T^{(t)}}{1-(p_I^{(t)}+p_T^{(t)})^2}, \\ n_{II}^{(t)} &= E\{N_{II}|n_C, n_I, n_T, p^{(t)}\} = n_I \frac{(p_I^{(t)})^2}{2p_C^{(t)} p_I^{(t)} + (p_T^{(t)})^2} \end{aligned}$$

$$n_{IT}^{(t)} \quad = \quad E\{N_{IT}|n_C, n_I, n_T, p^{(t)}\} = n_I \frac{2p_I^{(t)} p_T^{(t)}}{2p_C^{(t)} p_I^{(t)} + (p_T^{(t)})^2}.$$

最后一项

$$\begin{aligned} & k(n_C, n_I, n_T, p^{(t)}) \\ & = E\{log \frac{n!}{n_{CC}!n_{CI}!n_{CT}!n_{II}!n_{IT}!n_{TT}!} | n_C, n_I, n_T, p^{(t)}\} \end{aligned}$$

与$p$无关.

下面对$Q(p|p^{(t)})$进行最大化, 注意$p_C + p_I + p_T = 1$, 于是关于$p_C, p_I$求导

$$\begin{aligned} \frac{\partial Q(p|p^{(t)})}{\partial p_C} &= \frac{2n_{CC}^{(t)} + n_{CI}^{(t)} + n_{CT}^{(t)}}{p_C} - \frac{2n_{TT}^{(t)} + n_{CT}^{(t)} + n_{IT}^{(t)}}{1 - p_C - p_I}, \\ \frac{\partial Q(p|p^{(t)})}{\partial p_I} &= \frac{2n_{II}^{(t)} + n_{II}^{(t)} + n_{CI}^{(t)}}{p_I} - \frac{2n_{TT}^{(t)} + n_{CT}^{(t)} + n_{IT}^{(t)}}{1 - p_C - p_I} \end{aligned}$$

令导数为零,得到

$$p_C^{(t+1)} = \frac{2n_{CC}^{(t)} + n_{CT}^{(t)} + n_{CI}^{(t)}}{2n},$$

$$p_I^{(t+1)} = \frac{2n_{II}^{(t)} + n_{IT}^{(t)} + n_{CI}^{(t)}}{2n},$$
$$p_T^{(t+1)} = \frac{2n_{TT}^{(t)} + n_{CT}^{(t)} + n_{IT}^{(t)}}{2n}.$$

因此, 使用以上结论便可进行EM算法:

↑Code

```
moth<-function(p,n.obs){
   n<-sum(n.obs)
   nc<-n.obs[1]
   ni<-n.obs[2]
   nt<-n.obs[3]
   ntt<-nt

   cat(p,"\n")
   pct<-pit<-ptt<-rep(0,20)
   pct[1]<-p[1]
   pit[1]<-p[2]
   ptt[1]<-1-p[1]-p[2]
  for(i in 2:20){
```

```
 pc.old<-pct[i-1]
 pi.old<-pit[i-1]
 pt.old<-ptt[i-1]

 den<-pc.old^2+2*pc.old*pi.old+2*pc.old*pt.old
 ncc<-nc*pc.old^2/den
 nci<-2*nc*pc.old*pi.old/den
 nct<-2*nc*pc.old*pt.old/den
 nii<-ni*pi.old^2/(pi.old^2+2*pi.old*pt.old)
 nit<-2*ni*pi.old*pt.old/(pi.old^2+2*pi.old*pt.old)

 pct[i]<-(2*ncc+nci+nct)/(2*n)
 pit[i]<-(2*nii+nit+nci)/(2*n)
 ptt[i]<-(2*ntt+nct+nit)/(2*n)
 }
 return(list(pct=pct,pit=pit,ptt=ptt))
 }

n.obs<-c(85,196,341) # observed data,n_c,n_I,n_T
p<-c(1/3,1/3)
a<-moth(p,n.obs)
```

```
pct<-a$pct
pit<-a$pit
ptt<-a$ptt
#convergence diagnostics
# statistic R
rcc=sqrt( (diff(pct)^2+diff(pit)^2)/(pct[-20]^2+pit[-20]^2) )
rcc=c(0,rcc)  #adjusts the length to make the table below

d1=(pct[-1]-pct[20])/(pct[-20]-pct[20])
d1=c(d1,0)
d2=(pit[-1]-pit[20])/(pit[-20]-pit[20])
d2=c(d2,0)

#Table output
print(cbind(pct,pit,rcc,d1,d2)[1:9,],digits=5)
```

↓Code

其中,收敛的标准为$R^{(t)} = \frac{\|p^{(t)}-p^{(t-1)}\|}{\|p^{(t-1)}\|}$为由一次迭代到下一次迭代在$p^{(t-1)}$上相对改变的总量. (本例中我们以迭代10次为例,没有使用此标准控制收敛). 最后两列是验证EM算法的收敛速度为线性的.

**例2 Bayes 后验众数** 考虑一个具有似然$L(\theta|y)$, 先验$\pi(\theta)$以及缺失数据或者参数$Z$(即$y=(x,z)$) 的Bayes问题. 为找到后验众数, E步需要

$$\begin{aligned} Q(\theta|\theta^{(t)}) &= E\{log(L(\theta|y)\pi(\theta)k(y))|x,\theta^{(t)}\} \\ &= E\{logL(\theta|y)|x,\theta^{(t)}\}+log\pi(\theta)+E\{logk(y)|x,\theta^{(t)}\} \end{aligned}$$

其中最后一项可以在最大化$Q$中略去, 因其与$\theta$无关. 显然, 对此Bayes后验 众数问题, 和经典统计方法下的差异在于多了一项先验的对数. 因此此时EM算法为

1. E步: 计算$Q(\theta|\theta^{(t)})=E\{logL(\theta|y)|x,\theta^{(t)}\}$,
2. M步: 计算$\theta^{(t+1)}=argmaxQ(\theta|\theta^{(t)})$

考虑$X=(X_1,X_2,X_3)\sim MN(n,(2+\theta)/4,(1-\theta)/2,\theta/4)$, 为应用EM算法估计$\theta$, 我们视 完全数据为$Y=(Z_{11},Z_{12},X_2,X_3)\sim MN(n,1/2,\theta/4,(1-\theta)/2,\theta/4)$, 其中$Z_{11}+Z_{12}=X_1$.

因此, 有

$$l(\theta|Y) = (Z_{12} + X_3)log\theta + X_2log(1-\theta) + constants$$

$$E[l(\theta|Y)|X, \theta^{(t)}] = (E[Z_{12}|Z_{11} + Z_{12} = X_1, \theta^{(t)}] + X_3)log\theta$$

$$+X_2log(1-\theta) + constants$$

$$= (\frac{X_1\theta^{(t)}}{2+\theta^{(t)}} + X_3)log\theta + X_2log(1-\theta) + constants$$

考虑$\theta$的先验为$Beta(a, b)$,

$$\pi(\theta) = \frac{\Gamma(a+b)}{\Gamma(a)+\Gamma(b)}\theta^{a-1}(1-\theta)^{b-1}.$$

因此

$$Q(\theta|\theta^{(t)}) = (\frac{X_1\theta^{(t)}}{2+\theta^{(t)}} + X_3 + a - 1)log\theta + (X_2 + b - 1)log(1-\theta) + constants$$

所以得到

$$\theta^{(t+1)} = \left(\frac{X_1\theta^{(t)}}{2+\theta^{(t)}} + X_3 + a - 1\right) \Big/ \left(\frac{X_1\theta^{(t)}}{2+\theta^{(t)}} + X_3 + X_2 + a + b - 2\right)$$

R 代码如下

↑Code

```
theta=0.3
n<-50
x<-drop(rmultinom(1,n,c((2+theta)/4,(1-theta)/2,theta/4)))
# prior paramters
a<-.01
b<-.01

th<-rep(0,20)
th[1]<-0.2
for(t in 2:20){
  num<-x[1]*th[t-1]/(2+th[t-1])+x[3]+a-1
  den<-x[1]*th[t-1]/(2+th[t-1])+x[2]+x[3]+a+b-2
  th[t]<-num/den
  }
rcc<-sqrt( (diff(th)^2+diff(th)^2)/(th[-20]^2+th[-20]^2) )
print(cbind(th,c(0,rcc)),digits=5)
```

↓Code

## 1.2 Convergence

为了说明EM算法的收敛性, 我们通过说明在每步最大化过程增加了观测数据的对数似然量, $l(\theta|x)$. 以$X$表示可以观测项, $Z$表示缺失项, $Y = (X, Z)$, 则注意到观测数据密度的对数可以表示为

$$logf_X(x|\theta) = logf_Y(y|\theta) - logf_{Z|X}(z|x, \theta)$$

因此

$$E\{logf_X(x|\theta)|x, \theta^{(t)}\} = E\{logf_Y(y|\theta)|x, \theta^{(t)}\} - E\{logf_{Z|X}(z|x, \theta)|x, \theta^{(t)}\},$$

其中期望是关于$Z|x, \theta^{(t)}$计算的. 于是

$$logf_X(x|\theta) = Q(\theta|\theta^{(t)}) - H(\theta|\theta^{(t)}),$$

其中

$$H(\theta|\theta^{(t)}) = E\{logf_{Z|X}(Z|x, \theta)|x, \theta^{(t)}\}.$$

由于

$$\begin{aligned} H(\theta^{(t)}|\theta^{(t)}) - H(\theta|\theta^{(t)}) &= E\{logf_{Z|X}(Z|x,\theta^{(t)})|x,\theta^{(t)}\} \\ &- E\{logf_{Z|X}(Z|x,\theta)|x,\theta^{(t)}\} \\ &= \int -log\left[\frac{logf_{Z|X}(Z|x,\theta)|x,\theta^{(t)}}{logf_{Z|X}(Z|x,\theta^{(t)})|x,\theta^{(t)}}\right] f_{Z|X}(z|x,\theta^{(t)}dz \\ &\geq -log\int f_{Z|X}(z|x,\theta)dz = 0. \quad (Jensen's \ \ Inequality) \end{aligned}$$

即在每个$M$步中,

$$H(\theta^{(t)}|\theta^{(t)}) \geq H(\theta|\theta^{(t)}), \quad \forall\theta$$

等号当且仅当$\theta = \theta^{(t)}$时成立. 因此有

$$logf_X(x|\theta^{(t+1)}) - logf_X(x|\theta^{(t)}) \geq 0.$$

从而在每次迭代中选择$\theta^{(t+1)} = argmaxQ(\theta|\theta^{(t)})$就构成了标准的EM算法. 如果 我们仅仅是选择一个$\theta^{(t+1)}$使得$Q(\theta^{(t+1)}|\theta^{(t)}) > Q(\theta^{(t)}|\theta^{(t)})$, 则此

时的算法就称为是广义EM算法(GEM). 不管怎么样, 每一步增大$Q$, 都会增大对数似然. 使得该递增 收敛到某个极大似然估计的条件参看 Wu (1983)[1]和 Boyles (1983).[2]

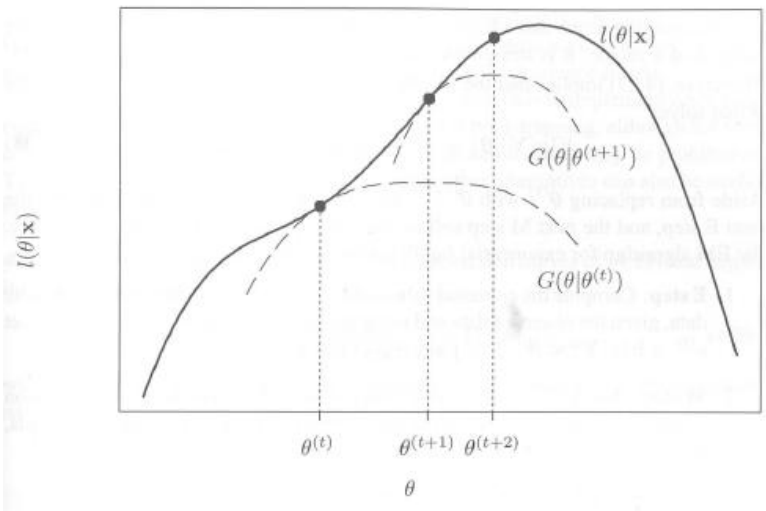
此外, 为进一步理解EM算法, 注意到观测数据的似然满足

$$l(\theta|x) \geq Q(\theta|\theta^{(t)}) + l(\theta^{(t)}|x) - Q(\theta^{(t)}|\theta^{(t)}) = G(\theta|\theta^{(t)}).$$

由于$G(\theta|\theta^{(t)})$的后两项和$\theta$无关, 因此$G(\theta|\theta^{(t)})$和$Q(\theta|\theta^{(t)})$ 在相同的点处达到最大值. 此外, $G$在$\theta^{(t)}$处和$l$相切, 且在任一处低于函数$l$. 在优化问题中, 函数$G$称为是$l$的一个劣化函数. 如下图所示

[1] C.F.J. Wu, On the convergence properties of the EM algorithm. Annals of Statistics, 11:95-103,1983.

[2] R.A. Boyles, On the convergence of the EM algorithm. Journal of the Royal Statistical Society, Series B, 45:47-50, 1983.




每个E步相当于构造函数$G$, 而每个M步等同于最大化该函数以给出一个上升的路径.

下面考虑EM方法的收敛速度. EM算法的全局收敛速度定义为

$$\rho = \lim_{t\to\infty} \frac{\|\theta^{(t+1)} - \hat{\theta}\|}{\|\theta^{(t)} - \hat{\theta}\|}$$

EM算法定义了一个映射$\theta^{(t+1)} = \Psi(\theta^{(t)})$, 其中 $\theta = (\theta_1, \ldots, \theta_p)$, $\Psi(\theta) = (\Psi(\theta_1), \ldots, \Psi(\theta_p))$, 当 EM算法收敛时, 如果收敛到该映射的一个不动点, 那

么$\hat{\theta} = \Psi(\hat{\theta})$. 设 $\Psi'(\theta)$表示其Jacobi矩阵, 其$(i,j)$元素为$\frac{\partial \Psi_i(\theta)}{\partial \theta_j}$. 则有

$$\theta^{(t+1)} - \hat{\theta} \approx \Psi'(\theta^{(t)})(\theta^{(t)} - \hat{\theta}).$$

因此当$p = 1$时, EM算法有线性收敛. 对$p > 1$, 若观测信息$-l''(\hat{\theta}|x)$是正定的, 则 收敛仍是线性的.

## 1.3 Usage in exponential families

当完全数据是被认为服从指数分布族里的某个分布时, 其密度可以写为$f(y|\theta) = c(\theta)exp\{\theta^T s(y)\}h(y)$, 其中$\theta$为自然参数向量, $s(y)$为充分统计量的向量. 此时, E步得到

$$Q(\theta|\theta^{(t)}) = k + logc(\theta) + \int \theta^T s(y) f_{Z|X}(z|x, \theta^{(t)})dz$$

其中$k$为与$\theta$无关的数. 在M步中, 令$Q$对$\theta$的导数为零, 整理得到

$$E_Y[s(Y)|\theta] = \frac{-c'(\theta)}{c(\theta)} = \int s(y) f_{Z|X}(z|x, \theta^{(t)})dz$$

其中左边等式是由指数族的性质得到的. 因此最大化$Q$相当于解上述等式, 且在每一个E步后, $Q$的形式不变, M步求解同样的最大化问题. 因此, 指数族的EM算法总结如下:

1. E步: 给定观测值和现有的参数值$\theta^{(t)}$, 计算$s^{(t)} = E_Y[s(Y)|\theta^{(t)}] = \int s(y) f_{Z|X}(z|x, \theta^{(t)}) dz$.
2. M步: 解方程$E_Y[s(Y)|\theta] = s^{(t)}$得到$\theta^{(t+1)}$.
3. 返回到E步, 直至收敛.

## 1.4 Usage in finite normal mixtures

EM算法经常被用于混合分布中参数的估计问题中. 考虑如下正态混合问题:

$$f(x|\theta) = \sum_{i=1}^{k} p_i f_i(x)$$

其中$p_i > 0, \sum_{i=1}^{k} p_i = 1$, $f_i(x)$为正态分布$N(\mu_i, \sigma^2)$的密度. $\theta = (p, \mu, \sigma^2)$为待估参数, 其中的$p = (p_1, \cdots, p_{k-1})$, $\mu = (\mu_1, \cdots, \mu_k)$. 除非$f(x|\theta)$ 是可识

别的, 否则使用$\mathbf{x} = (x_1, \cdots, x_n)$的信息估计$\theta$是没有意义的. 而对有限正态混合, 恰恰存在这种问题. 比如$k = 2$时, 我们不能区分$(p_1, \mu_1, \mu_2, \sigma^2)$和$(p_2, \mu_2, \mu_1, \sigma^2)$. 实际应用中很少考虑 参数的不可识别性问题. 但是这种不可识别性很容易解决, 比如规定$p_1 < p_2$(Titterington et al. 1985)对混合 分布下参数的识别性问题进行了更多的讨论. 此处我们不作讨论.

为将此问题转化为带有缺失信息的问题, 我们引入变量$\mathbf{z} = (z_1, \cdots, z_n)$, 其中$z_j = (z_{1j}, \cdots, z_{kj})$, 其分量$z_{ij} = (z_j)_i = 1$, 如果$x_j$来源于$f_i$; 否则取值0. 即将每个样本$x_j$所属的总体进行二进制编码. 则完全数据为

$$\mathbf{y} = (\mathbf{x}, \mathbf{z})$$

因此完全数据对数似然函数为

$$\begin{aligned} l(\theta|\mathbf{y}) &= log \prod_{j=1}^{n} f(x_j, z_j) = log \prod_{j=1}^{n} \prod_{i=1}^{k} [p_i f_i(x_j)]^{z_{ij}} \\ &= \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} log p_i - \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{n} z_{ij} [log\sigma^2 + (x_j - \mu_i)^2/\sigma^2] + const. \end{aligned}$$

**E Step**

注意由Bayes定理得到

$$E[Z_{ij}|x,\theta^{(t)}] = P(Z_{ij}=1|x,\theta^{(t)}) = \frac{p_i^{(t)} f_i(x_j)}{\sum_{i=1}^k p_i^{(t)} f_i(x_j)} := z_{ij}^{(k)}$$

因此

$$Q(\theta|\theta^{(t)} = \sum_{i=1}^k \sum_{j=1}^n z_{ij}^{(t)} log p_i - \frac{1}{2}\sum_{i=1}^k \sum_{j=1}^n z_{ij}^{(t)} [log\sigma^2 + (x_j-\mu_i)^2/\sigma^2] + const.$$

**M Step**

最大化$Q(\theta|\theta^{(t)})$得到

$$p_i^{(t+1)} = \frac{1}{n}\sum_{j=1}^n z_{ij}^{(t)}, \qquad \mu_i^{(t+1)} = \frac{\sum_{j=1}^n z_{ij}^{(t)} x_j}{\sum_{j=1}^n z_{ij}^{(t)}}$$

$$(\sigma^2)^{(t+1)} = \sum_{i=1}^k \sum_{j=1}^n z_{ij}^{(t)} (x_j - \mu_i^{(t+1)})^2/n$$

## 1.5 Variance estimation

在极大似然框架下, EM算法用以寻求一个极大似然估计, 但不能自动给出极大似然估计的协方差阵的一个估计. 我们已经知道对极大似然估计, 在正则化条件下, 其满足渐近正态性:

$$\sqrt{n}(\hat{\theta}-\theta) \to N(0, I^{-1}(\theta))$$

其中$I(\theta) = E[\frac{\partial log f(X|\theta)}{\partial \theta}][\frac{\partial log f(X|\theta)}{\partial \theta^T}]$为Fisher信息阵. 因此极大似然估计$\hat{\theta}$的协方差矩阵的一个自然估计就是$\frac{1}{n} I(\hat{\theta})^{-1}$.

注意到$I(\theta) = -E\frac{\partial^2 log f(X|\theta)}{\partial \theta \partial \theta^T}$, 因此若记样本$\mathbf{x} = (x_1, \ldots, x_n)$, 则对数似然函数为

$$l(\theta|\mathbf{x}) = logL(\theta|\mathbf{x}) = logf(\mathbf{x}|\theta) = \sum_{i=1}^{n} logf(x_i|\theta).$$

从而Fisher信息阵$I(\theta)$可以通过计算观测信息(对数似然函数的Hessian阵)

$$-l''(\hat{\theta}|\mathbf{x}) = -\sum_{i=1}^{n} \frac{\partial^2 log f(x_i|\theta)}{\partial \theta \partial \theta^T}|_{\theta=\hat{\theta}}$$

来估计.

在Bayes分析中, 后验众数的协方差矩阵估计可同理来估计, 这是因为在后验分布下, 也有渐近正态性成立. 区别 在于要计算对数后验密度的Hessian阵.

### 1.5.1 Louis method

注意到观测对数似然可以表示为

$$l(x|\theta) = log f_X(x|\theta) = Q(\theta|\theta^{(t)}) - H(\theta|\theta^{(t)})$$

两边对$\theta$求二阶偏导再取负号得到

$$-l''(\theta|x) = -Q''(\theta|\omega)|_{\omega=\theta} + H''(\theta|\omega)_{\omega=\theta}$$

即

$$\hat{i}_X(\theta) = \hat{i}_Y(\theta) - \hat{i}_{Z|X}(\theta) \tag{1.1}$$

其中$\hat{i}_X(\theta) = -l''(\theta|x)$为观测信息, 而$\hat{i}_Y(\theta)$和$\hat{i}_{Z|X}(\theta)$ 分别称为完全信息和缺失信息. 交换积分和求导顺序(可能时), 有

$$\hat{i}_Y(\theta) = -Q''(\theta|\omega)|_{\omega=\theta} = -E[l''(\theta|Y)|x,\theta]$$

即为完全信息的Fisher信息阵. 类似的结果对$-H''$也成立. 上面的结果说明观测信息等于完全信息减去 缺失信息.

可以证明

$$\hat{i}_{Z|X}(\theta) = var\{\frac{\partial log f_{Z|X}(Z|x,\theta)}{\partial\theta}\}$$

其中方差是对$Z|x,\theta$的条件分布计算的. 进一步若$\hat{\theta}$为极大似然估计, 则可以证明

$$\hat{i}_{Z|X}(\hat{\theta}) = E_{Z|x,\hat{\theta}}\left\{\frac{\partial log f_{Z|X}(Z|x,\theta)}{\partial\theta}\frac{\partial log f_{Z|X}(Z|x,\theta)}{\partial\theta^T}|_{\theta=\hat{\theta}}\right\}$$

公式(1.1)使得我们可以通过完全数据似然和缺失数据的条件似然来表达$\hat{i}_X(\theta)$, 从而可以避免计算可能很复杂的边际似然. 但是并不比直接计算$-l''(\theta|x)$明显容易.

如果$\hat{i}_Y(\theta)$和$\hat{i}_{Z|X}(\theta)$难以解析计算, 可以通过Monte Carlo方法来估计, 比如$\hat{i}_Y(\theta)$的一个自然估计为

$$\hat{i}_Y(\theta) = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial^2 log f(y_i|\theta)}{\partial\theta\partial\theta^T}$$

其中$y = (x, z)$是模拟的完全数据集, 它有观察信息$x$和从$f_{Z|X}(z|x, \theta)$中抽取的样本$z$构成. 类似也可以得到$\hat{i}_{Z|X}(\theta)$的估计.

**例 删失的指数数据** 假定完全数据$Y_1, \cdots, Y_n \sim exp(\lambda)$, 而观测数据 是右删失的, 即只能观测到$x = (x_1, \cdots, x_n)$, 其中$x_i = (\min(y_i, c_i), \delta_i)$, $c_i$为 删失水平, 如果$y_i \le c_i$, 则$\delta_i = 1$; 否则$\delta_i = 0$.

显然完全数据的似然函数为$l(\lambda|y) = nlog\lambda - \lambda\sum_{i=1}^{n} y_i$. 因此

$$\begin{aligned} Q(\lambda|\lambda^{(t)}) &= E[l(\lambda|Y)|x, \lambda^{(t)}] \\ &= nlog\lambda - \lambda\sum_{i=1}^{n} E[Y_i|x_i, \lambda^{(t)}] \end{aligned}$$

$$= \quad nlog\lambda - \lambda \sum_{i=1}^{n} [y_i\delta_i + (c_i + 1/\lambda^{(t)})(1-\delta_i)]$$

$$= \quad nlog\lambda - \lambda \sum_{i=1}^{n} [y_i\delta_i + c_i(1-\delta_i)] - C\lambda/\lambda^{(t)}.$$

其中$C = \sum_{i=1}^{n}(1-\delta_i)$. 因此$-Q''(\lambda|\lambda^{(t)}) = n/\lambda^2$.

一个删失事件$Z_i$的未观测到的结果有密度$f_{Z_i|X}(z_i|x,\lambda) = \lambda exp\{-\lambda(z_i - c_i)\}I_{z_i>c_i}$. 因此容易计算得到

$$\frac{dlogf_{Z|X}(Z|x,\lambda)}{\partial\lambda} = C/\lambda - \sum_{i:\delta_i=0}(Z_i - c_i).$$

由于$Z_i - c_i$服从指数分布, 因此上式的方差为

$$\hat{i}_{Z|X}(\lambda) = \sum_{i:\delta_i=0} var(Z_i - c_i) = C/\lambda^2.$$

这样, 应用Louis方法, 即有

$$\hat{i}_X(\lambda) = n/\lambda^2 - C/\lambda^2$$

对这个简单例子, 可以直接计算得到$-l''(\lambda|x) = n/\lambda^2 - C/\lambda^2$.

### 1.5.2 SEM algorithm

在前面我们曾引入映射$\theta^{(t+1)} = \Psi(\theta^{(t)})$, 以及不动点$\hat{\theta}$(极大似然估计)和$(i,j)$元为$\frac{\partial \Psi_i(\theta)}{\partial \theta_j}$的 矩阵$\Psi'(\theta)$. Dempster et al. (1977) 证明了

$$\Psi'(\hat{\theta})^T = \hat{i}_{Z|X}(\hat{\theta})\hat{i}_Y(\hat{\theta})^{-1}$$

注意由前面的信息表示(1.1)有

$$\hat{i}_X(\theta) = [I - \hat{i}_{Z|X}(\theta)\hat{i}_Y(\theta)^{-1}]\hat{i}_Y(\theta)$$

其中$I$为单位阵, 因此得到$\hat{\theta}$协方差矩阵的估计

$$\hat{var}(\hat{\theta}) = [\hat{i}_X(\hat{\theta})]^{-1} = \hat{i}_Y(\hat{\theta})^{-1}[I - \Psi'(\hat{\theta})^T]^{-1}$$

此式的意义在于将想要得到的协方差矩阵表示成完全数据的协方差阵加上一个考虑到了缺失信息的矩阵.

因此上式在得到$\Psi'(\hat{\theta})$的后, 即可得到极大似然估计$\hat{\theta}$的协方差矩阵的估计. Dempster et al. 使用数值微分的方法估计$\Psi'$, 并称他们的方法为扩展的EM算法(Supplemented EM algorithm). 该想法如下

1. 运行EM算法直至收敛, 找到最大值点$\hat{\theta}$.

2. 从非常靠近$\hat{\theta}$的$\theta^{(0)}$重新开始EM算法.

3. 对$t = 0, 1, 2, \cdots$开始SEM迭代. 在第$t$步SEM迭代中, 其是首先通过E步和M步生成了$\theta^{(t)}$, 然后定义

$$\theta^{(t)}(j) = (\hat{\theta}_1, \cdots, \hat{\theta}_{j-1}, \theta_j^{(t)}, \hat{\theta}_{j+1}, \cdots, \hat{\theta}_p), j = 1, \ldots, p$$

对$i, j = 1, \cdots, p$, 计算

$$r_{ij}^{(t)} = \frac{\Psi_i(\theta^{(t)}(j)) - \hat{\theta}_i}{\theta_j^{(t)} - \hat{\theta}_j}$$

注意$\Psi$的定义, 因此$\Psi_i(\theta^{(t)}(j))$表示在有了$\theta^{(t)}(j)$后, 再进行一次E步, 得到$\theta_*^{(t+1)} = \Psi_i(\theta^{(t)}(j))$. 另外, 由于$\Psi(\hat{\theta}) = \hat{\theta}$, 因此此即为计算$\Psi'$的数值微分步.

4. 当$r_{ij}^{(t)}$所有的值对$t \geq t^*$稳定时, 就可作为$\Psi'$的估计.

**例 椒花蛾续** 对前面的椒花蛾例子, 可以使用SEM方法对极大似然估计的协方差矩阵进行估计. 由$p_C^{(0)} = 0.07$和$p_I^{(0)} = 0.19$开始, 程序如下

↑Code

```
#First, run EM using the code in the basic EM algorithm given above.
#Then we run SEM as follows.
#Starting values for SEM
#Choosing the parameter values at the 100th iteration of EM.
pcthat=pct[20]
pithat=pit[20]
ptthat=ptt[20]

#Initialize the parameter estimate vectors, one vector for each phenotype
sempct=rep(0,20)
sempit=rep(0,20)
semptt=rep(0,20)

sempct[1]=0.07      #Initial SEM estimates of the parameters
sempit[1]=0.19
```

```
semptt[1]=0.74
rij=array(0,c(3,3,20))

#This for loop implements the SEM algorithm for the peppered moth
#example.
#Below t is the number of SEM iterations

for (t in 2:20) {
  #Take standard EM step (see code above for detailed description)
  denom1=(sempct[t-1]^2+2*sempct[t-1]*sempit[t-1]+2*sempct[t-1]*semptt[t-1])
  ncct=nc*sempct[t-1]^2/denom1
  ncit=2*nc*sempct[t-1]*sempit[t-1]/denom1
  nctt=nc-ncct-ncit
  niit=ni*sempit[t-1]^2/(sempit[t-1]^2+2*sempit[t-1]*semptt[t-1])
  nitt=ni-niit
  nttt=nt
  sempct[t]=(2*ncct+ncit+nctt)/(2*n)
  sempit[t]=(2*niit+ncit+nitt)/(2*n)
  semptt[t]=(2*nttt+nctt+nitt)/(2*n)

  #SEM loop over each parameter
```

```
for (j in 1:3) {
  #start at estimates from the the 20th iteration of EM
  sempj=c(pcthat,pithat,ptthat)

  #replace the jth element of sempj with the most recent EM estimate
  sempj[j]=c(sempct[t],sempit[t],semptt[t])[j]

  #Take one EM step for sempj
  denom1=(sempj[1]^2+2*sempj[1]*sempj[2]+2*sempj[1]*sempj[3])
  ncct=nc*sempj[1]^2/denom1
  ncit=2*nc*sempj[1]*sempj[2]/denom1
  nctt=nc-ncct-ncit
  niit=ni*sempj[2]^2/(sempj[2]^2+2*sempj[2]*sempj[3])
  nitt=ni-niit
  nttt=nt
  nextstep=c((2*ncct+ncit+nctt)/(2*n),(2*niit+ncit+nitt)/(2*n),
     (2*nttt+nctt+nitt)/(2*n))

  # Calculate rij.
  rij[,j,t]=(nextstep-c(pcthat,pithat,ptthat))/
                (sempj[j]-c(pcthat,pithat,ptthat)[j]) }
```

```
    }

  for(t in 1:t){
  cat(t,sempct[t],sempit[t],semptt[t])
  cat("\n")
  print(rij[,,t])
  cat("\n")
  }

#Note that the algorithm becomes unstable on 8th iteration
#Below is the output for iteration 8
#EM after 20 iterations
#> cbind(pct,pit,ptt)[20,]
#       pct        pit        ptt
#0.07083691 0.18873652 0.74042657
#SEM after 7 iterations
#> cbind(sempct,sempit,semptt)[6,]
#    sempct     sempit     semptt
#0.07083691 0.18873670 0.74042639
rij[,,6]
#             [,1]         [,2]        [,3]
```

```
#[1,]  0.034117920 -0.002601059 -0.00260106
#[2,] -0.006910223  0.140676982 -0.03519589
#[3,] -0.027207449 -0.138075923  0.03779695
#Now need iyhat (inverse of the information matrix for Y)
denom1=(pcthat^2+2*pcthat*pithat+2*pcthat*ptthat)
ncct=nc*pcthat^2/denom1
ncit=2*nc*pcthat*pithat/denom1
nctt=nc-ncct-ncit
niit=ni*pithat^2/(pithat^2+2*pithat*ptthat)
nitt=ni-niit
nttt=nt
#Required derivatives for iyhat
d20q=-(2*ncct+ncit+nctt)/pcthat^2 -(2*nttt+nitt+nctt)/(ptthat^2)
d02q=-(2*niit+ncit+nitt)/pithat^2 -(2*nttt+nitt+nctt)/(ptthat^2)
d12q=-(2*nttt+nitt+nctt)/(ptthat^2)

iyhat=-cbind(c(d20q,d12q),c(d12q,d02q))

solve(iyhat)
#[1,]  5.290920e-05 -1.074720e-05
#[2,] -1.074720e-05  1.230828e-04
```

```
#Since the percentages of the 3 moth phenotypes add up to 1, we only
#presented results for carbonaria and insularia phenotypes.

psiprime=rij[,,6] #(7th iteration)
psiprime22=psiprime[-3,-3]

#variance matrix of MLE
varhat=solve(iyhat)%*%(diag(2)+t(psiprime22)%*%solve(diag(2)-t(psiprime22)))
varhat=(varhat+t(varhat))/2
varhat
#                [,1]           [,2]
#[1,]  5.481298e-05 -1.223007e-05
#[2,] -1.223007e-05  1.433249e-04
#sd(pc), sd(pi)
sqrt(diag(varhat))
#[1] 0.007403579 0.011971838
#cor(pc,pi)
varhat[1,2]/prod(sqrt(diag(varhat)))
#[1] -0.1379833
#var(pt)
```

```
varhat[1,1]+varhat[2,2]+2*varhat[1,2]
#[1] 0.0001736777
#sd(pt)
sqrt(sum(varhat))
#[1] 0.01317868
#cor(pc,pt)
(-varhat[1,1]-varhat[1,2])/(sqrt(varhat[1,1])*sqrt(sum(varhat)))
#[1] -0.4364370
#cor(pi,pt)
(-varhat[2,2]-varhat[1,2])/(sqrt(varhat[2,2])*sqrt(sum(varhat)))
#[1] -0.8309075
```

↓Code

### 1.5.3 Bootstrap method

Bootstrap方法是应用起来最方便的方法. 其算法如下

对$b = 1, \cdots, B$,

1. 使用观测数据$x_1, \ldots, x_n$和EM算法得到一个估计$\hat{\theta}$, 并记为$\hat{\theta}_1$.

2. 有放回的从$x_1, \ldots, x_n$中选出$x_1^*, \ldots, x_n^*$, 应用EM算法得到估计$\hat{\theta}_j$, $j = 2, \cdots, B$.

3. 使用$\hat{\theta}_1, \cdots, \hat{\theta}_B$得到$\hat{\theta}$的样本协方差矩阵, 作为其协方差矩阵的估计.

### 1.5.4 Empirical Information

当数据是$i.i.d$时, 注意到Fisher信息阵为

$$I(\theta) = cov\left[\frac{\partial l(\theta|\mathbf{x})}{\partial \theta}\right] = cov\left[\sum_{i=1}^{n} l'(\theta|x_i)\right]$$

因此可以使用样本协方差

$$\frac{1}{n}\sum_{i=1}^{n}[l'(\theta|x_i) - \bar{l}'(\theta|\mathbf{x})][l'(\theta|x_i) - \bar{l}'(\theta|\mathbf{x})]^T \tag{1.2}$$

来估计上式, 其中$\bar{l}'(\theta|\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} l'(\theta|x_i)$.

该方法吸引人的地方在于估计(1.2)的每一项都可以在EM算法执行过程中得到, 不需要额外的分析. 事实上, 由于$\hat{\theta}^{(t)}$最大化$Q(\theta|\theta^{(t)} - l(\theta|x)$, 因此关

于$\theta$求导得到

$$Q'(\theta|\theta^{(t)})|_{\theta=\theta^{(t)}} = l'(\theta|x)|_{\theta=\theta^{(t)}}$$

由于$Q'$通常在每个M步中计算, 因此(1.2)的每一项都是可以得到的.

## 1.6 EM Variants

### 1.6.1 Improving the E step

E步需要计算条件期望, 其值我们用$Q(\theta|\theta^{(t)})$来表示. 如果此期望难以解析计算时, 则可以 通过Monte Carlo方法来近似.

#### Monte Carlo EM

Wei and Tanner (1990) 提出第$t$次迭代的E步可以通过如下两步来替代

1. 从条件分布$f_{Z|X}(z|x,\theta^{(t)}$中抽取$i.i.d$样本$Z_1^{(t)},\ldots,Z_{m_t}^{(t)}$, 每个 $Z_j^{(t)}$都是缺失数据$Z$的一个观测. 这样$Y_j = (x, Z_j)$表示一个补齐的完整数据.

2. 计算$\hat{Q}^{(t+1)}(\theta|\theta^{(t)}) = \frac{1}{m_t}\sum_{j=1}^{m_t} log f_Y(Y_j^{(t)}|\theta)$.

那么$\hat{Q}^{(t+1)}(\theta|\theta^{(t)})$即作为$Q(\theta|\theta^{(t)})$的一个估计. M步改为最大化 $\hat{Q}^{(t+1)}(\theta|\theta^{(t)})$. 推荐的策略是在初期的EM迭代中使用较小的$m_t$, 并随着迭代的进行逐渐增大$m_t$, 以减少在$\hat{Q}$中 引入的Monte Carlo波动性. 不过这种Monte Carlo EM算法和普通的EM算法收敛方式不一样, 随着迭代的进行, $\theta^{(t)}$的值最终在真实的最大值附件波动, 其精度依赖于$m_t$.

### 1.6.2 Improving the M step

EM算法的特点就在于对利用完全似然的$Q$求导和最大化通常要比计算观测数据的极大似然简单. 然而, 在某些情况下, 即使$Q$的导出是容易的, M步也不容易实施. 为此提出了一些策略来克服此困难.

#### EM gradient algorithm

如果最大化不能用解析的方式得到, 那么可以考虑使用优化方法. 这将导致嵌套迭代. Lange 提出 使用单步的Newton法替代M步, 从而可以近似取得最大

值而不用真正的精确求解. M步用下面替代

$$\begin{aligned}\theta^{(t+1)} &= \theta^{(t)} - Q''(\theta|\theta^{(t)})^{-1}|_{\theta=\theta^{(t)}} Q'(\theta|\theta^{(t)})|_{\theta=\theta^{(t)}} \\ &= \theta^{(t)} - Q''(\theta|\theta^{(t)})^{-1}|_{\theta=\theta^{(t)}} l'(\theta^{(t)}|x)\end{aligned}$$

EM梯度算法和完全的EM算法对$\hat{\theta}$有相同的收敛速度.

## 1.7 Pros and Cons

**EM算法的优点**

1. EM算法是数值稳定的, EM算法的每一次迭代会增加对数似然.

2. 在非常一般的条件下, EM算法有可靠的全局收敛性, 即从参数空间的任何初始值出发, EM算法 一般总能收敛到对数似然函数的一个局部最大值点.

3. EM算法很容易实施,每一次迭代中的E步是对完全似然的期望, 而M步是完全数据的ML估计. 其经常 有closed form.

4. EM算法是容易进行程序设计的, 因其只涉及到似然函数, 而不需要其导数.

5. EM算法在计算机上实施时需要的存储空间很小, 其在每步迭代中不需要存储信息阵或者其导数等.

6. 由于完全数据问题一般是一个标准问题, 因此在完全数据的MLE不存在显式解时, M步经常可以使用标准的统计方法来解决此问题. 一些扩展的EM算法就是基于此.

7. 相比其他优化方法而言, EM算法需要的分析计算工作一般不太多, 其只要求完全对数似然的条件期望和最大化.

8. 每一个迭代的成本是比较低的, 其抵消了EM算法需要很大迭代次数以达到收敛的不足之处.

9. 通过监视每次迭代对数似然的单调性, 可以方便的监视收敛性和程序错误等.

10. EM算法可以用于对"缺失"值进行估计.

**对EM算法的一些批评**

1. 不能自动给出参数估计的协方差矩阵的估计.

2. 在一些看起来很简单的问题或者包含太多缺失信息的问题里, EM算法可能收敛很慢.

3. 当存在多个极值点时, EM算法不能保证收敛到全局最大值点, 此时, 收敛到的极值点依赖于初始值.

4. 在一些问题里, EM算法需要的E步或者M步可能不能给出分析解.