EM算法学习(Expectation Maximization Algorithm)

一、前言

这是本人写的第一篇博客，是学习李航老师的《统计学习方法》书以及斯坦福机器学习课Andrew Ng的EM算法课后，对EM算法学习的介绍性笔记，如有写得不恰当或错误的地方，请指出，并多多包涵，谢谢。另外本人数学功底不是很好，有些数学公式我会说明的仔细点的，如果数学基础好，可直接略过。

二、基础数学知识

在正式介绍EM算法之前，先介绍推导EM算法用到的数学基础知识，包括凸函数，Jensen不等式。

1.凸函数

对于凸函数，凹函数，如果大家学过高等数学，都应该知道，需要注意的是国内教材如同济大学的《高等数学》的这两个概念跟国外刚好相反，为了能更好的区别，本文章把凹凸函数称之为上凸函数，下凸函数，具体定义如下：

上凸函数：函数f(x)满足对定义域上任意两个数a,b都有f[(a+b)/2] ≥ [f(a)+f(b)]/2

下凸函数：函数f(x)满足对定义域上任意两个数a,b都有f[(a+b)/2] ≤ [f(a)+f(b)]/2

更直观的可以看图2.1和2.2：

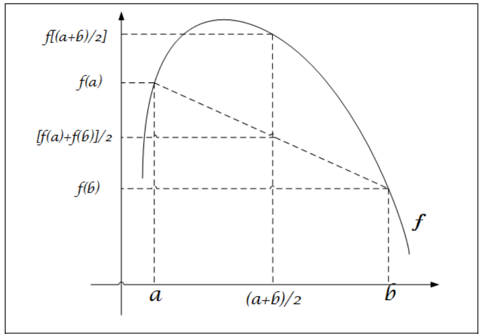

图2.1 上凸函数

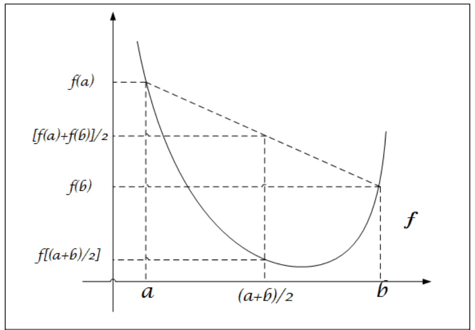

图2.2 下凸函数

可以清楚地看到图2.1上凸函数中，f[(a+b)/2] ≥ [f(a)+f(b)]/2，而且不难发现，如果f(x)是上凸函数，那么-f(x)是下凸函数。

当a≠b时，f[(a+b)/2] > [f(a)+f(b)]/2成立，那么称f(x)为严格的上凸函数，等号成立的条件当且仅当a=b,下凸函数与其类似。

2.Jensen不等式

有了上述凸函数的定义后，我们就能很清楚的Jensen不等式的含义，它的定义如下：

如果f是上凸函数，X是随机变量，那么f(E[X]) ≥ E[f(X)]

特别地，如果f是严格上凸函数，那么E[f(X)] = f(E[X])当且仅当p(X=E[X])，也就是说X是常量。

那么很明显 log x 函数是上凸函数，可以利用这个性质。

有了上述的数学基础知识后，我们就可以看具体的EM算法了。

三、EM算法所解决问题的例子

在推导EM算法之前，先引用《统计学习方法》中EM算法的例子：

例1. (三硬币模型) 假设有3枚硬币,分别记作A,B,C。这些硬币正面出现的概率分别为π，p和q。投币实验如下，先投A，如果A是正面，即A=1，那么选择投B；A=0，投C。最后，如果B或者C是正面，那么y=1；是反面，那么y=0；独立重复n次试验（n=10），观测结果如下： 1,1,0,1,0,0,1,0,1,1假设只能观测到投掷硬币的结果，不能观测投掷硬币的过程。问如何估计三硬币正面出现的概率，即π，p和q的值。

解：设随机变量y是观测变量，则投掷一次的概率模型为

$$P(y|\theta) = \pi p^y (1-p)^{1-y} + (1-\pi) q^y (1-q)^{1-y}$$

有n次观测数据Y，那么观测数据Y的似然函数为

$$P(Y \mid \theta) = \prod_{n}^{j=1} \left[\pi p^{y_j} (1-p)^{1-y_j} + (1-\pi) q^{y_j} (1-q)^{1-y_j} \right]$$

那么利用最大似然估计求解模型解，即

$$\begin{aligned} \hat{\theta} &= arg arg\, max_\theta\, log\, P(Y \mid \theta) && (1) \\ . \quad &= arg\, max_\theta \sum_{j=1}^{10} log\, P(y^j \mid \theta) && (2) \\ . \quad &= arg\, max_\theta \sum_{j=1}^{10} log \left[\pi p^{y_j} (1-p)^{1-y_j} + (1-\pi) q^{y_j} (1-q)^{1-y_j} \right] && (3) \end{aligned}$$

这里将概率模型公式和似然函数代入（1）式中，可以很轻松地推出 (1)=> (2) => (3)，然后选取θ(π,p,q)，使得（3）式值最大，即最大似然。然后，我们会发现因为（3）中右边多项式+符号的存在，使得（3）直接求偏导等于0或者用梯度下降法都很难求得θ值。

这部分的难点是因为（3）多项式中+符号的存在，而这是因为这个三硬币模型中，我们无法得知最后得结果是硬币B还是硬币C抛出的这个隐藏参数。那么我们把这个latent 随机变量加入到 log-likelihood 函数中，得

$$
\begin{aligned}
l(\theta) &= \sum_{j=1}^{10} log \sum_{i=1}^{2} P(y_j, z_i \mid \theta) && (4) \\
&= \sum_{j=1}^{10} log \sum_{i=1}^{2} Q_j(z_i) \frac{P(y_j, z_i \mid \theta)}{Q_j(z_i)} && (5) \\
&\geq \sum_{j=1}^{10} \sum_{i=1}^{2} Q_j(z_i) log \frac{P(y_j, z_i \mid \theta)}{Q_j(z_i)} && (6)
\end{aligned}
$$

略看一下，好像很复杂，其实很简单，请容我慢慢道来。首先是公式（4），这里将z_i做为隐藏变量，当z_1为结果由硬币B抛出，z_2为结果由硬币C抛出，不难发现

$$
\begin{aligned}
\sum_{i=1}^{2} P(y_j, z_i \mid \theta) &= P(y_j \mid \theta) \\
&= \pi p^{y_j}(1-p)^{1-y_j} + (1-\pi) q^{y_j}(1-q)^{1-y_j}
\end{aligned}
$$

注：一下Q中有些许漏了下标j,但不影响理解

接下来公式说明（4）=>（5）（其中（5）中Q(z)表示的是关于z的某种分布，$\sum_i Q(z_i \mid \theta) = 1$），很直接，在P的分子分母同乘以$Q(z_i)$。最后是（5）=>（6），到了这里终于用到了第二节介绍的Jensen不等式，数学好的人可以很快发现，

$\sum_{i=1}^{2} Q(z_i) \frac{P(y_j, z_i \mid \theta)}{Q(z_i)}$就是$[\frac{P(y_j, z_i \mid \theta)}{Q(z_i)}]$的期望值（如果不懂，可google期望公式并理解），

且log是上凸函数，所以就可以利用Jensen不等式得出这个结论。因为我们要让log似然函数l(θ)最大，那么这里就要使等号成立。根据Jensen不等式可得，要使等号成立，则要使$P(y_j, z_i \mid \theta)/Q(z_i) = c$成立。

再因为$\sum_i Q(z_i \mid \theta) = 1$，所以得$\sum_i P(y_j, z_i \mid \theta) = c$,c为常数，那么

$$
\begin{aligned}
Q_j(z_i \mid \theta) &= P(y_j, z_i \mid \theta) / \sum_i P(y_j, z_i \mid \theta) \\
&= P(y_j, z_i \mid \theta) / P(y_j \mid \theta) \\
&= P(z_i \mid y_j, \theta)
\end{aligned}
$$

这里可以发现

$$
Q_j(z_1 \mid \theta) = \frac{\pi p^{y_j}(1-p)^{1-y_j}}{\pi p^{y_j}(1-p)^{1-y_j} + (1-\pi) q^{y_j}(1-q)^{1-y_j}}
$$

$$
Q_j(z_2 \mid \theta) = \frac{(1-\pi) q^{y_j}(1-q)^{1-y_j}}{\pi p^{y_j}(1-p)^{1-y_j} + (1-\pi) q^{y_j}(1-q)^{1-y_j}}
$$

OK,到这里，可以发现公式（6）中右边多项式已经不含有“+”符号了，如果知道Q(z)的所有值，那么可以容易地进行最大似然估计计算，但是Q的计算需要知道θ的值。这样的话，我们是不是可以先对θ进行人为的初始化θ_0，然后计算出Q的所有值Q_1(在θ_0固定的情况下，可在Q_1取到公式（6）的极大值)，然后在对公式（6）最大似然估计，得出新的θ_1值（在固定Q_1的情况下，取到公式（6）的极大值），这样又可以计算新的Q值Q_1,然后依次迭代下去。答案当然是可以。因为Q_1是在θ_0的情况下产生的，可以调节公式（6）中θ值，使公式（6）的值再次变大，而θ值变了之后有需要调节Q使（6）等号成立，结果又变大，直到收敛（单调有界必收敛），如果到现在还不是很清楚，具体清晰更广义的证明可以见下部分EM算法说明。

ps:看到上述的橙黄色内容，如果大家懂得F函数的极大-极大算法的话，就可以知道其实它们是一码事。

另外对公式（6）进行求偏导等于0，求最大值，大家可以自己练习试试，应该很简单的，这里不做过多陈述。

在《统计学习方法》书中，进行两组具体值的计算

（1）π_0=0.5, p_0=0.5, q_0=0.5，迭代结果为π=0.5, p=0.6, q=0.5

（2）π_0=0.4, p_0=0.6, q_0=0.7，迭代结果为π=0.4064, p=0.5368, q=0.6432

两组值的最后结果不相同，这说明EM算法对初始值敏感，选择不同的初值可能会有不同的结果，只能保证参数估计收敛到稳定点。因此实际应用中常用的办法就是选取多组初始值进行迭代计算，然后取结果最好的值。

在进行下部分内容之前，还需说明下一个东西。在上面的举例说明后，其实可以发现上述的解决方法跟一个简单的聚类方法很像，没错，它就是K-means聚类(不懂的见百度百科关于K-means算法的说明）。K-means算法先假定k个中心，然后进行最短距离聚类，之后根据聚类结果重新计算各个聚类的中心点，一次迭代，是不是很像，而且K-means也是初始值敏感，因此其实K-means算法也包含了EM算法思想，只是这边EM算法中用P概率计算，而K-means直接用最短距离计算。所以EM算法可以用于无监督学习。在下一篇文章，我准备写下典型的用EM算法的例子，高斯混合模型（GMM,Gaussian Mixture Model）。

四、EM算法

1.模型说明

考虑一个参数估计问题，现有$\{y_1, y_2, \dots, y_n\}$共n个训练样本，需有多个参数θ去拟合数据，那么这个log似然函数是：

$$l(\theta) = \sum_{j=1}^{n} log\, P(y_j \mid \theta)$$

可能因为θ中多个参数的某种关系（如上述例子中以及高斯混合模型中的3类参数），导致上面的log似然函数无法直接或者用梯度下降法求出最大值时的θ值，那么这时我们需要加入一个隐藏变量z，以达到简化l(θ),迭代求解l(θ)极大似然估计的目的。

2.EM算法推导

这小节会对EM算法进行具体推导，许多跟上面例子的解法推导是相同的，如果已经懂了，可以加速阅读。首先跟“三硬币模型”一样，加入隐变量z后，假设Q(z)是关于隐变量z的某种分布，那么有如下公式：

$$
\begin{aligned}
l(\theta) &= \sum_{j=1}^{n} log \sum_{i} P(y_j, z_i \mid \theta) && (7) \\
&= \sum_{j=1}^{n} log \sum_{i} Q_j(z_i) \frac{P(y_j, z_i \mid \theta)}{Q_j(z_i)} && (8) \\
&\geq \sum_{j=1}^{n} \sum_{i} Q_j(z_i) log \frac{P(y_j, z_i \mid \theta)}{Q_j(z_i)} && (9)
\end{aligned}
$$

公式（7）是加入隐变量，（7）=>（8）是在$P(y_j, z_i \mid \theta)$基础上分子分母同乘以$Q_j(z_i)$，（8）=>（9）用到Jensen不等式（跟“三硬币模型”一样），等号成立的条件是$P(y_j, z_i \mid \theta)/Q_j(z_i) = c$，c是常数。再因为$\sum_i Q_j(z_i) = 1$，则有如下Q的推导：

$$
\begin{aligned}
&\sum_i P(y_j, z_i \mid \theta)/c = 1 \\
&\Rightarrow \sum_i P(y_j, z_i \mid \theta) = c \\
&\Rightarrow Q_j(z_i) = P(y_j, z_i \mid \theta) / \sum_i P(y_j, z_i \mid \theta) \\
&\qquad\qquad = P(y_j, z_i \mid \theta)/P(y_j \mid \theta) \\
&\qquad\qquad = P(z_i \mid y_j, \theta)
\end{aligned}
$$

再一次重复说明，要使（9）等式成立，则$Q_j(z_i)$为y_j,z的后验概率。算出$Q_j(z_i)$后（9）就可以进行求偏导，以剃度下降法求得θ值，那么又可以计算新的$Q_j(z_i)$值，依次迭代，EM算法就实现了。

EM算法（1）：

选取初始值θ^0初始化θ，t=0

Repeat {

E步：

$$
\begin{aligned}
Q_j^t(z_i) &= P(y_j, z_i \mid \theta^t) / \sum_i P(y_j, z_i \mid \theta^t) \\
. \qquad &= P(y_j, z_i \mid \theta^t) / P(y_j \mid \theta^t) \\
. \qquad &= P(z_i \mid y_j, \theta^t)
\end{aligned}
$$

M步：

$$
\theta^{t+1} = arg\, max_\theta \sum_{j=1}^{n} \sum_i Q_j^t(z_i) log \frac{P(y_j, z_i \mid \theta)}{Q_j^t(z_i)}
$$

$$
. \qquad t = t + 1
$$

}直到收敛

3.EM算法收敛性证明

当θ取到θ ^{t}值时，求得

$$
\theta^{t+1} = arg\, max_\theta \sum_{j=1}^{n} \sum_i Q_j^t(z_i) log \frac{P(y_j, z_i \mid \theta)}{Q_j^t(z_i)}
$$

那么可得如下不等式：

$$
\begin{aligned}
l(\theta^{t+1}) &= \sum_{j=1}^{n} log \sum_i Q_j^t(z_i) \frac{P(y_j, z_i \mid \theta^{t+1})}{Q_j^t((z_i)} \qquad (10) \\
. \qquad &\geq \sum_{j=1}^{n} \sum_i Q_j^t(z_i) log \frac{P(y_j, z_i \mid \theta^{t+1})}{Q_j^t((z_i)} \qquad (11) \\
. \qquad &\geq \sum_{j=1}^{n} \sum_i Q_j^t(z_i) log \frac{P(y_j, z_i \mid \theta^{t})}{Q_j^t((z_i)} \qquad (12)
\end{aligned}
$$

（10）=>（11）是因为Jensen不等式，因为等号成立的条件是θ为θ ^{t}的时候得到的$Q_j(z_i)$，而现在$P(y_j, z_i \mid \theta)$中的θ值为θ $^{t+1}$，所以等号不一定成立，除非θ $^{t+1}$=θ ^{t}，

（11）=>（12）是因为θ $^{t+1}$已经使得$\sum_{j=1}^{n}\sum_{i} Q_j^t(z_i) log \frac{P(y_j, z_i \mid \theta)}{Q_j^t(z_i)}$取得最大值，那必然不会小于（12）式。

所以l(θ)在迭代下是单调递增的，且很容易看出l(θ)是有上界的（单调有界收敛），则EM算法收敛性得证。

4.EM算法E步说明

上述EM算法描述，主要是参考Andrew NG教授的讲义，如果看过李航老师的《统计方法学》，会发现里面的证明以及描述表明上有些许不同，Andrew NG教授的讲义的说明（如上述）将隐藏变量的作用更好的体现出来，更直观，证明也更简单，而《统计方法学》中则将迭代之间θ的变化罗列的更为明确，也更加准确的描述了EM算法字面上的意思：每次迭代包含两步：E步，求期望；M步，求极大化。下面列出《统计方法学》书中的EM算法，与上述略有不同：

EM算法（2）：

选取初始值θ ^{0}初始化θ，t=0

Repeat {

E步：

$$\begin{aligned} H(\theta, \theta^t) &= E_z[logP(Y, Z \mid \theta) \mid Y, \theta^t] \\ &= \sum_z P(Z \mid Y, \theta^t) logP(Y, Z \mid \theta) \end{aligned} \qquad (13)$$

.

M步：

$$\theta^{t+1} = arg\, max_\theta\ H(\theta, \theta^t)$$

}直到收敛

(13)式中，Y={y_1,y_2,...,y_m},Z={z_1,z_2,...,z_m},不难看出将（9）式中两个Σ对换，就可以得出（13）式，而（13）式即是关于分布z的一个期望值，而需要求这个期望公式，那么要求出所有的EM算法（1）中E步的值，所以两个表明看起来不同的EM算法描述其实是一样的。

五、小结

EM算法的基本思路就已经理清，它计算是含有隐含变量的概率模型参数估计，能使用在一些无监督的聚类方法上。在EM算法总结提出以前就有该算法思想的方法提出，例如HMM中用的Baum-Welch算法就是。

在EM算法的推导过程中，最精妙的一点就是（10）式中的分子分母同乘以隐变量的一个分布，而套上了Jensen不等式，是EM算法顺利的形成。

六、主要参考文献

[1]Rabiner L, Juang B. An introduction to hidden markov Models. IEEE ASSP Magazine, January 1986，EM算法原文

[2]http://v.163.com/special/opencourse/machinelearning.html，Andrew NG教授的公开课中的EM视频

[3]http://cs229.stanford.edu/materials.html, Andrew NG教授的讲义，非常强大，每一篇都写的非常精炼，易懂

[4]http://www.cnblogs.com/jerrylead/archive/2011/04/06/2006936.html, 一个将Andrew NG教授的公开课以及讲义理解非常好的博客，并且我许多都是参考他的

[5]http://blog.csdn.net/abcjennifer/article/details/8170378, 一个浙大研一的女生写的，里面的博客内容非常强大，csdn排名前300，ps:本科就开博客，唉，我的大学四年本科就给了游戏，玩，惭愧哈，导致现在啥都不懂。

[6]李航.统计学习方法.北京：清华大学出版社，2012

下节预告：高斯混合模型（GMM,Gaussian Mixture Model）及例子，大家可以看看JerryLead的 http://www.cnblogs.com/jerrylead/archive/2011/04/06/2006924.html,写得挺好，大部分翻译自 Andrew NG教授的讲义

分类: 统计学习方法

标签: EM

Mind Puzzle
关注 - 0
粉丝 - 39
+加关注

2 0

» 下一篇：高斯混合模型（GMM）

posted @ 2013-04-05 22:02 Mind Puzzle 阅读(6624) 评论(7) 编辑 收藏

#1楼 2013-09-23 20:02 Ja °

总结的非常好，找了N多篇看到的最好的一篇。本科同样玩过来的，哎

支持(0) 反对(0)

#2楼 2013-11-19 08:26 Julia H

good!

支持(0) 反对(0)

#3楼 2014-04-10 21:44 宋明强

有没有程序实现和简单实例呢？

支持(0) 反对(0)

#4楼[楼主] 2014-04-10 22:55 Mind Puzzle

@ 宋明强
没有特别为这个写个程序，不过网上有很多HMM的，GMM，PLSA的算法（即混合模型）都是用了EM算法的，我自己本身写过一个Python版HM

M算法，你可以去Github上找找看。

支持(1) 反对(0)

#5楼 2014-04-24 14:57 紫梦lan

@ Mind Puzzle
github地址是？

支持(0) 反对(0)

#6楼 2014-09-25 21:08 yimmon

“因为我们要让log似然函数l(θ)最大，那么这里就要使等号成立。”
这里说得有点容易让人误解。使等号成立是为了确定一个l(θ)的下界，然后通过不断最大化下界从而实现最大化l(θ)。

支持(0) 反对(0)

#7楼 2014-11-22 00:28 c++_thinker

(13)式中，Y={y1,y2,...,ym},Z={z1,z2,...,zm},不难看出将（9）式中两个Σ对换，就可以得出（13）式
就上一句我有个问题，就是说对换过log项的分母为什么不见了？不是太理解

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

最新IT新闻:

· Dnsmasq发现三个远程代码执行漏洞

· 专访诺奖得主巴里什：新世纪三大物理突破都得靠大装置

· 法官宣布Waymo和Uber诉讼官司推迟至12月4日

· 上线新功能，Instagram看准的是“即看即买”的移动购物体验

· 假期荐书 | 这两本书改变了扎克伯格对创新的思考方式

» 更多新闻...

最新知识库文章:

· 实用VPC虚拟私有云设计原则

· 如何阅读计算机科学类的书

· Google 及其云智慧

· 做到这一点，你也可以成为优秀的程序员

· 写给立志做码农的大学生

» 更多知识库文章...

本博客用于记录本人学习技术的过程，在每篇文章的后面我会列出参考文献，以供参考，内容如有雷同，请勿见怪。

昵称：Mind Puzzle
园龄：4年6个月
粉丝：39
关注：0
+加关注

< 2013年4月 >

日	一	二	三	四	五	六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

搜索

找找看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

我的标签

EM(4)

GMM(2)

HMM(1)

Map-Matching(1)

随笔分类

LBS(1)

统计学习方法(4)

随笔档案

2014年5月 (1)

2014年4月 (2)

2013年4月 (2)

最新评论

1. Re:基于隐马尔科夫模型(HMM)的地图匹配(Map-Matching)算法

@李屠户请问下，你有没有搞过类似的。...

--cn_world

2. Re:高斯混合模型（GMM）

非常感谢，学习啦

--jiwy

3. Re:EM算法(Expectation Maximization Algorithm)

@PawnZhP(yj,zi|θ) = c Qj(zi), 所以∑iP(yj,zi|θ)=c∑iQj(zi)=c...

--Mind Puzzle

4. Re:EM算法(Expectation Maximization Algorithm)

请问“再因为∑iQj(zi)=1，所以得∑iP(yj,zi|θ)=c”这块怎么理解？怎么得到的∑iP(yj,zi|θ)=c？

--PawnZh

5. Re:基于隐马尔科夫模型(HMM)的地图匹配(Map-Matching)算法

非常好的文章，而且已经有开源的 Java 实现了:

--oldrev

阅读排行榜

1. 高斯混合模型（GMM）(14685)

2. EM算法学习(Expectation Maximization Algorithm)(6624)
3. 基于隐马尔科夫模型(HMM)的地图匹配(Map-Matching)算法(6139)
4. 高斯混合模型(GMM)(3308)
5. EM算法(Expectation Maximization Algorithm)(1967)

评论排行榜

1. EM算法(Expectation Maximization Algorithm)(8)
2. EM算法学习(Expectation Maximization Algorithm)(7)
3. 高斯混合模型（GMM）(5)
4. 基于隐马尔科夫模型(HMM)的地图匹配(Map-Matching)算法(2)

推荐排行榜

1. 基于隐马尔科夫模型(HMM)的地图匹配(Map-Matching)算法(4)
2. EM算法(Expectation Maximization Algorithm)(3)
3. EM算法学习(Expectation Maximization Algorithm)(2)
4. 高斯混合模型（GMM）(1)
5. 高斯混合模型(GMM)(1)