

**《机器学习基础》课程报告**

****

2022-2023学年第1学期（CST30106）

姓名: 陈鹏宇

学号: 20204227

成绩:

重庆大学计算机学院

**EM算法浅析**

**摘 要：**首先从双硬币问题引入EM算法，分析EM算法能解决什么问题和EM算法的直观理解，随后将推导EM算法并证明收敛性来解释其原理，总结出EM算法的优势和限制，最后介绍EM算法应用的例子。

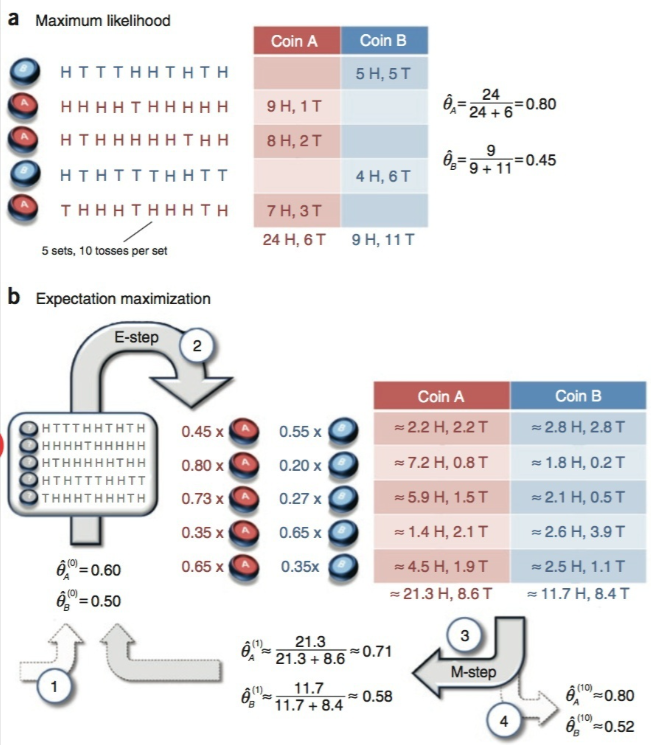
**关键词：**EM算法；极大似然函数；算法推导；

1 模型引入

1.1双硬币问题

 这是Nature Biotech的EM tutorial文章中比较经典的一个例子[[[1]](#endnote-0)]。我们假设有两枚硬币A、B，他们由于质量分布不均匀，所以投掷出来正反面的概率各不相同。随机投掷硬币10次为一组，共5组，在知道每组投掷的是哪枚硬币的情况下很容易就能计算出AB硬币投掷出正面的概率。但如果我们并不知道每组投掷的是哪枚硬币，就无法直接计算AB硬币正面朝上的概率了，这时就需要引入EM算法。

如果想计算出正面朝上的概率，需要先知道每个硬币的种类，但要估计种类又必须知道概率。而EM的解决方法就是先初始化正面朝上的概率，然后用去估计每个硬币的种类，接着根据种类按照最大似然估计出新的概率，循环直到收敛，如下图所展示。通过双硬币问题的求解我们能引入到EM算法的具体概念。



**图 1 双硬币模型图**

1.2EM算法

1.2.1 概念

EM算法[[[2]](#endnote-1)]是1977年由Dempster、Laird和Rubin总结提出的用于含有隐含变量的概率模型参数的极大似然估计方法,主要用来计算在不完全数据或混合模型情况下的参数估计问题。通俗的来讲，EM算法就是在有缺失数据时极大似然估计的一种迭代方法。

1.2.2 直观理解

通过解决双硬币问题，我们能发现EM算法主要分为两步。

* E步：基于给定观测数据和当前模型参数估计隐变量分布，再通过这个估计的分布来计算似然函数的期望值

(1)

* M步：基于观测数据和E步求得的隐变量分布，求得最大化似然期望时对应的参数

(2)

循环迭代E步和M步直到收敛即可得到局部最优解。但仅基于E步和M步的计算步骤，我们无法解决两个问题，一是为什么一定会收敛，二是为什么是局部最优解。所以我们需要剖析EM算法的原理。

2 EM算法的原理

2.1公式推导

根据给定数据集，假设样本间相互独立，根据极大似然原理估计模型参数的公式我们能得到以下似然函数

(3)

但由于的未知性，所以我们需要引入隐变量z利用EM算法解决。加入隐变量后可以将替换为，即求和每个样本所有可能分布z的联合概率密度。

(4)

其中z在求解问题中的具体含义是某个样本处于z分布，以双硬币问题为例，z代表选中硬币为硬币A或者硬币B。其中，每个样本所属隐变量分布的概率和为1。

(5)

代表i样本属于z隐变量分布的概率。根据(5)改写(4)可得

(6)

Jensen不等式给出，如果f是凹函数，X是随机变量，则，当X = E[X]时等号成立，当f严格是凹函数时，则，若为凸函数则相反。根据log函数的性质，由于log(x)的二阶导数恒小于0，为凹函数。我们可以将log函数看为一个整体，看为概率，看为关于z变量的函数，再根据期望公式：

(7)

结合Jensen不等式和式(7)、式(6)：

(8)

(9)

至此，我们发现EM算法的原理。式(9)这个过程可以看作求出了的下界函数，而EM算法的核心就是拔高下界，从而达到逐步极大化似然函数的目的。根据下界函数可知，主要有两个参数影响，一是，二是，我们可以通过调整这两个参数使下界不断上升，EM算法首先会固定其中的第一个参数，然后使用 MLE 计算第二个变量值；接着通过固定第二个变量，再使用 MLE 估测第一个变量值，依次迭代，直至收敛到局部最优解。按照这个思路，我们要找到等式成立的条件。根据Jensen不等式，要想让等式成立，只需要让随机变量变成常数值：

(10)

(11)

因此得到：

(12)

再结合联合分布到边缘分布计算公式和条件概率公式可得：

(13)

至此，我们发现在推出，在固定参属下，使下界提高的的计算公式就是后验概率，解决了其如何选择的问题，这其实就是上部分提到的E步，目的是构建的下界。

* E步：计算每一个样本在某隐变量分布下的概率

(14)

M步则在给定的情况下，调整，从而极大化似然函数的下界。

* M步：计算极大化似然函数时的取值

(15)

2.2证明过程

我们通过公式推导得到了EM算法的原理，接下将证明EM算法的收敛性。要证明收敛性，即可证明迭代结果是逐步增加的，即证明

(16)

恒成立。也就是说极大似然估计单调递增，最终会达到极值点。选定后，根据E步(14)，我们可以得到

(17)

且根据式(17)的推导，此时，Jensen不等式等式成立：

(18)

随后进行M步，即下界的提高。我们将固定，将当作未知数，对求导可得：

(19)

易得的导数恒大于0，即单调递增恒成立，至此证毕。或者从定义出发，由于M步是固定的情况下极大化，即

(20)

而根据M步的定义和E步保证等式成立的条件[[[3]](#endnote-2)]

(21)

即可证毕

(22)

2.3优势和限制

通过之前的推导和证明不难发现，EM算法主要是通过迭代方法解决带隐变量的参数估计问题，但通过之前的学习我们知道梯度下降法同样能解决。只是如果我们需要求解的参数越多，那么计算量会成指数级上升，对计算带来麻烦。相对来说，EM算法是一种非梯度优化算法。

2.3.1 优势

* 算法步骤简单，效果稳定，相比于梯度下降法，能更有效解决含隐变量的参数估计问题或者混合模型问题。
* 自收敛的分类算法，既不需要实现设定类别也不需要数据间的两两比较合并等操作。

2.3.2 限制

* 在极大似然函数不为凸函数的情况下，EM算法不能保证收敛到全局最优解。
* 对初始参数过于敏感，在初始参数选择不当的情况下，很容易影响收敛效率以及能否得到全局最优解。
* EM算法收敛速度慢，不适合大规模或者高维数据集。尽管随着多位学者的深入研究，对EM算法在收敛速度上进行改进，但收敛速度仍然不理想[[[4]](#endnote-3)]。
* 对于某些特殊模型，M步的计算是比较困难的，而在某些情况下，要获得E步中的期望显示也是非常困难甚至不可能的[[[5]](#endnote-4)]。

3 EM算法的应用

通过上述EM的引入和推导，我们能认识到EM算法在含缺失数据问题中的应用。随着当前研究的数据量成指数级增长，EM算法的缺陷也越来越明显，神经网络拟合、卡尔曼滤波法逐渐代替EM进行数据添加。但与其说EM是一种算法，不如说是一种解决问题的思想，与SVM等一些具体解决问题的算法不同，EM算法更像是其他算法的逻辑基础，起到简化问题的作用。复杂的问题通过引入合适的隐变量，能更高效的解决问题。

除数据添加外，EM算法主要应用在聚类算法计算[[[6]](#endnote-5)]中，如K-means（k均值算法）和GMM（高斯混合模型）。下面将简单介绍EM在k-means中的思想。

1. means主要也是两个步骤交替进行，分别对应EM中的E步和M步：

* E步：（更新）簇划分，将每个点选择最近的类优化目标函数，分给中心距离它最近的类。
* M步：计算每个簇的均值向量，更新每个类的中心点，可以认为是在各类分布均为单位方差的高斯分布的假设下，最大化似然值。

更直观解释的话，K-means使用的是hard EM算法，即二分抉择，而传统EM为soft EM，即一种概率分布。

4 小结

本文首先从双硬币模型引入EM算法，介绍了EM算法概念和最直观的理解。然后根据根据凸凹函数的性质和Jensen不等式对EM算法进行推导和收敛性证明，分析了EM的原理，即利用创造下界函数并不断优化下界的方法逐步逼近极值点的核心思想，然后得出它的优势和限制。由于篇幅有限，最后仅简单介绍了EM算法在聚类中的应用。总之，综合本文对EM算法的浅析，能初步理解EM算法的原理，掌握使用EM算法处理含隐变量参数估计的问题，同时认识到EM算法还有部分局限性。

**参考文献：**

1. [] Do, C. B., & Batzoglou, S. (2008). What is the expectation maximization algorithm? Nature Biotechnology, 26(8), 897–899. [↑](#endnote-ref-0)
2. [] Dempster A P, Laird N M, Rubin D B. Maximum likelihood from incomplete data via the EM algorithm[J]. Journal of the royal statistical society. Series B (methodological), 1977: 1-38. [↑](#endnote-ref-1)
3. [] 李顺静. 基于EM算法的缺失数据的统计分析及应用[D].重庆工商大学,2015. [↑](#endnote-ref-2)
4. [] 李柏椿. EM算法及其改进算法在参数估计中的应用研究[D].重庆大学,2017. [↑](#endnote-ref-3)
5. [] 张宏东. EM算法及其应用[D].山东大学,2014. [↑](#endnote-ref-4)
6. [] 岳佳. 基于EM算法的模型聚类的研究及应用[D].江南大学,2007. [↑](#endnote-ref-5)