期望最大化（EM）

1. 摘要

期望最大化算法(Expectation-maximization algorithm)是机器学习中一个非常重要的算法，又称作 EM 算法。EM算法是由Dempster等人1977年提出的统计模型参数估计的一种算法。它采用的迭代交替搜索方式可以简单有效的求解最大似然函数估计问题。已知的概率模型内部存在隐含的变量，导致了不能直接用极大似然法来估计参数，EM算法就是通过迭代逼近的方式用实际的值带入求解模型内部参数的算法。它在当代的工业、商业和科学研究领域发挥了重要的作用。

EM算法分成2个部分，E步骤和M步骤。其中，E步骤叫做期望化步骤，M步骤为最大化步骤。 本文将会详细推导和解释以上两个步骤。

二、 案例描述

要解决的问题：期望最大化算法是解决对于不可观察变量进行似然估计的一种方法。最大期望算法是一种启发式的迭代算法，是一种从“不完全数据”中求极大似然的方法。在人工智能、机器学习、数理统计、模式识别等许多应用都需要进行模型的参数估计，极大似然估计和极大后验似然估计是必要进行的。然而在理想的可观察变量模型中，即变量分布式均匀的时候，做出以上两个估计是显然可以的。但是实际的情况往往不是这样，某些变量并不是可以观察的，对这类模型进行极大似然估计就比较复杂了。

举例：在一个样本集｛,,……｝,将它建模成P（x），此时有一个样本点,并且P（）<ε,(ε是阈值)。但是在一整个样本集中我们无法确定哪个样本点是，即它是不可观察的。但是我们可以假设样本点为某一类，计算出分布律，再通过分布律反向推导出概率参数，最后将得出的参数代入假设的公式中进行调整。  
 已知：此时并不知道 属于哪一类，我们假设为某一类，得到相应的分布概率后进行调整。样本集为一维高斯分布估计，令为,=j～multinomal(),则有：

P(,)= P(|)\*P();

P(=j)=;

而P(|=j)～N(,)，则：

P(|=j)= exp(- );

L(, μ,Σ)=\*P(,);  
 未知：样本点的分布律

参数,,

三、 解决思路

解决方法:期望最大化。

期望最大化主要是用来计算后验分布的众数或极大似然估计。该方法广泛的应用于缺损数据、截尾数据、成群数据、带有复杂参数的数据等不完整数据。期望最大化算法流行的原因，一是在于它的理论简单化和一般性，二是许多应用都能够纳入到期望最大化算法的范畴，期望最大化算法已经成为统计学上的一个标准工具。

理由：(1)EM算法中，由于似然函数是有界的，并且算法的每一步迭代都使似然函数增加，根据单调有界定理可以证明EM算法具有收敛性；

(2) EM算法是一种初始值敏感的算法，选取不同初始参数会有不同的最终结果；

(3) EM算法得到的不会是全局最优，每次迭代逼近的都是当前的局部最优。

四、 解决原理

EM算法，分成2个部分，E步骤和M步骤。其中，E步骤叫做期望化步骤，M步骤为最大化步骤。

整体算法的步骤如下所示：

1. 初始化分布参数。

2. E步骤：(1) 将当前轮次得到的参数https://gss3.bdstatic.com/7Po3dSag_xI4khGkpoWK1HF6hhy/baike/s%3D24/sign=9af1d3cdad86c9170c03553dc93d6f72/060828381f30e924b7a2ab0446086e061d95f784.jpg作为固定的观测点；

(2) 利用参数https://gss3.bdstatic.com/7Po3dSag_xI4khGkpoWK1HF6hhy/baike/s%3D24/sign=9af1d3cdad86c9170c03553dc93d6f72/060828381f30e924b7a2ab0446086e061d95f784.jpg计算出隐含变量Z的分布，计算期望E；

(3) 将隐含变量Z的分布带入构造似然函数在当前观测点下的下界函数，计算其最大似然估计值，以此实现期望化的过程。

3. M步骤：最大化在E-步骤上的最大似然估计值来计算参数的值；

4. 重复2、3步骤直到收敛。

如此循环既可以求出似然函数的一个局部最优。图1更直观得展现这第i次迭代的过程。

（一）对于一般的EM算法的公式表达形式：

给定训练样本是｛,,……｝, 样例间独立，我们想找到每个样例隐含的类别z，能使得p(x,z)最大。p(x,z)的最大似然估计如下：

l(θ)=

l(θ)=,(是为了方便下文计算引入的函数)

l(θ)=>=

(Jensen’s Inequality: f(x)为凸函数，x为变量，则E[f(x)]=f[E(x)])

取等号时：

l(θ)=

对于每一个样例i，让https://gss1.bdstatic.com/9vo3dSag_xI4khGkpoWK1HF6hhy/baike/s%3D16/sign=a9d14d2603f79052eb1f43380cf3ed2d/5d6034a85edf8db1426642510323dd54564e74be.jpg 表示该样例隐含变量z的某种分布， https://gss1.bdstatic.com/9vo3dSag_xI4khGkpoWK1HF6hhy/baike/s%3D16/sign=a9d14d2603f79052eb1f43380cf3ed2d/5d6034a85edf8db1426642510323dd54564e74be.jpg 满足的条件是 =1 ，>=0。由此，可以确定式子的下界，然后不断的提高此下界达到逼近最后真实值的目的值，这个不等式变成等式为止，然后再依据Jensen不等式，当不等式变为等式的时候，当且仅当，也就是说X是常量，推出就是下面的公式：

=c(常数)

由于Q是随机变量z的概率密度函数，因此，可以得到：分子的和等于c（分子分母都对所有z求和：多个等式分子分母相加不变，这个认为每个样例的两个概率比值都是c）。

=

由此可得出EM算法的一般过程：循环重复E步骤和M步骤直到收敛。

E-step: =

M-step: θ:=argmax \*

（二）对于一维高斯分布估计的EM，也分为两个步骤，E步骤和M步骤。过程如下：

E-step:=P(=j|;, μ,Σ)

=

=

M-step: =\*

=

=

五、 实验程序

用MATLAB进行仿真，源程序如下（一维高斯分布估计）：

%EM

M=3; % M个高斯分布混合

N=600; % 样本数

th=0.000001; % 收敛阈值

K=2; % 样本维数

% 待生成数据的参数

a\_real =[2/3;1/6;1/6];% 混合模型中基模型高斯密度函数的权重

mu\_real=[3 4 6;5 3 7];% 均值

cov\_real(:,:,1)=[5 0;0 0.2];% 协方差

cov\_real(:,:,2)=[0.1 0;0 0.1];

cov\_real(:,:,3)=[0.1 0;0 0.1];

% 生成符合标准的样本数据（每一列为一个样本）

x=[ mvnrnd( mu\_real(:,1) , cov\_real(:,:,1) , round(N\*a\_real(1)) )' ,...

mvnrnd( mu\_real(:,2) , cov\_real(:,:,2) , round(N\*a\_real(2)) )' ,...

mvnrnd( mu\_real(:,3) , cov\_real(:,:,3) , round(N\*a\_real(3)) )' ];

% mvnrnd 用来生成多维正态数据

% 初始化参数

a=[1/3;1/3;1/3];

mu=[1 2 3;2 1 4];

cov(:,:,1)=[1 0;0 1];

cov(:,:,2)=[1 0;0 1];

cov(:,:,3)=[1 0;0 1];

t=inf;% inf 代表无穷大

w=0;

while t>=th

a\_old = a;

mu\_old = mu;

cov\_old= cov;

rznk\_temp=zeros(M,N);

for k=1:M

for n=1:N

% 计算P(x|mu\_cm,cov\_cm)

rznk\_temp(k,n)=exp(-1/2\*(x(:,n)-mu(:,k))'\*inv(cov(:,:,k))\*(x(:,n)-mu(:,k))); %inv 求逆矩阵

end

rznk\_temp(k,:)=rznk\_temp(k,:)/sqrt(det(cov(:,:,k)));% 求矩阵的行列式

end

rznk\_temp=rznk\_temp\*(2\*pi)^(-K/2);

%E step

% 求rznk

rznk=zeros(M,N);

for n=1:N

for k=1:M

rznk(k,n)=a(k)\*rznk\_temp(k,n);

end

rznk(:,n)=rznk(:,n)/sum(rznk(:,n));

end

% M step（以下求各个参数）

% 求Nk

nk=zeros(1,M);

nk=sum(rznk');

% 求a

a=nk/N

% 求MU

for k=1:M

mu\_k\_sum=0;

for n=1:N

mu\_k\_sum=mu\_k\_sum+rznk(k,n)\*x(:,n);% 对rznk(k,n)\*x(:,n)进行求和

end

mu(:,k)=mu\_k\_sum/nk(k);

end

% 求COV

for k=1:M

cov\_k\_sum=0;

for n=1:N

cov\_k\_sum=cov\_k\_sum+rznk(k,n)\*(x(:,n)-mu(:,k))\*(x(:,n)-mu(:,k))';

% 对rznk(k,n)\*(x(:,n)-mu(:,k))\*(x(:,n)-mu(:,k))'进行求和

end

cov(:,:,k)=cov\_k\_sum/nk(k);

end

t=max([norm(a\_old(:)-a(:))/norm(a\_old(:));norm(mu\_old(:)-mu(:))/norm(mu\_old(:));norm(cov\_old(:)-cov(:))/norm(cov\_old(:))]);

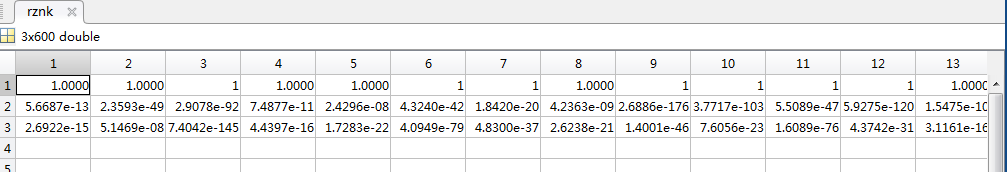
% 如果A为矩阵，n=norm(A)，返回A的最大奇异值

w=w+1 ;% 迭代次数

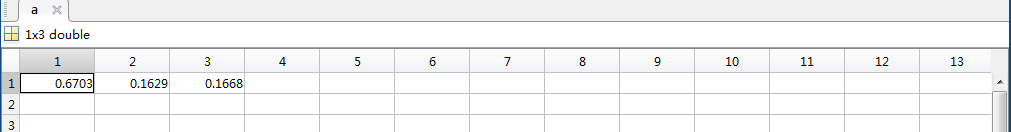
end

仿真结果如下：

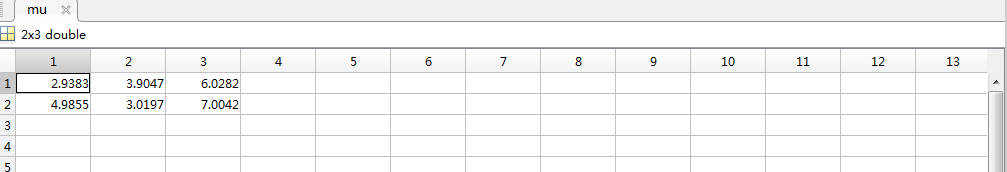
rznk（由于rnzk结果是3\*600的矩阵，这里只显示一部分）：



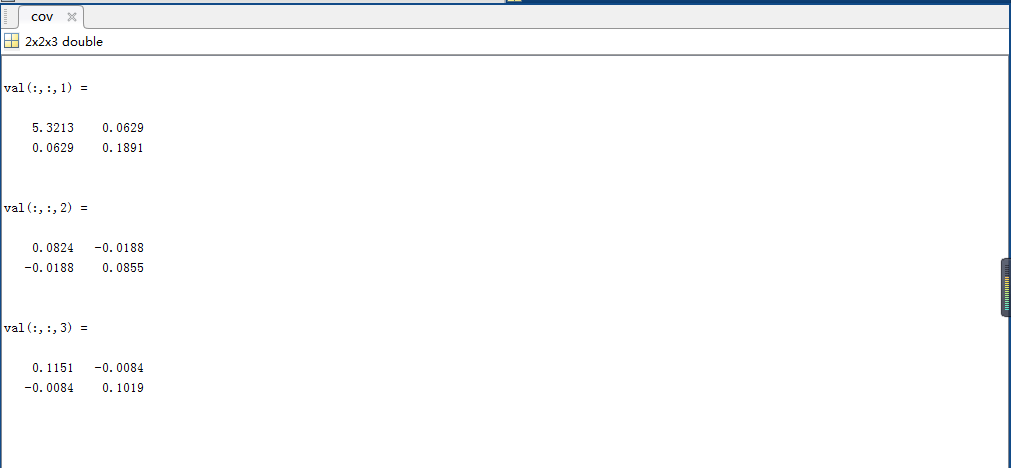
a:



mu:



cov:



六、 实验结果的解释说明

在这个程序中，输出的rznk即为我们上文中未知参数，通过假设为某一类别求出分布律=，程序实现为：

for n=1:N

for k=1:M

rznk(k,n)=a(k)\*rznk\_temp(k,n);

end

rznk(:,n)=rznk(:,n)/sum(rznk(:,n));

end

a为上文中，=\*，程序实现为：

nk=zeros(1,M);

nk=sum(rznk');

a=nk/N;

mu为上文中，=，程序实现为：

for k=1:M

mu\_k\_sum=0;

for n=1:N

mu\_k\_sum=mu\_k\_sum+rznk(k,n)\*x(:,n);

end

mu(:,k)=mu\_k\_sum/nk(k);

end

cov为上文中，=，程序实现为：

for k=1:M

cov\_k\_sum=0;

for n=1:N

cov\_k\_sum=cov\_k\_sum+rznk(k,n)\*(x(:,n)-mu(:,k))\*(x(:,n)-mu(:,k))';

end

cov(:,:,k)=cov\_k\_sum/nk(k);

end

即，a,mu,cov都是求得的参数。

七、 结论

EM算法包括两个步骤：由E步和M步组成，它是通过迭代地最大化完整数据的对数似然函数的期望来最大化不完整数据的对数似然函数。

通过交替使用这两个步骤，EM算法逐步改进模型的参数，使参数和训练样本的似然概率逐渐增大，最后终止于一个极大点。直观地理解EM算法，它也可被看作为一个逐次逼近算法：事先并不知道模型的参数，可以随机的选择一套参数或者事先粗略地给定某个初始参数，确定出对应于这组参数的最可能的状态，计算每个训练样本的可能结果的概率，在当前的状态下再由样本对参数修正，重新估计参数，并在新的参数下重新确定模型的状态，这样，通过多次的迭代，循环直至某个收敛条件满足为止，就可以使得模型的参数逐渐逼近真实参数。

在本文的程序中，我们实现了一维高斯分布估计EM的整个过程，首先我们不知道未知变量属于哪一类别，假设为某一类，计算其分布律rznk()，并通过分布律求得各个未知参数a（）,mu（）,cov（），再做一个循环，用求得的参数调整分布律。在这个程序中，经过多次试验发现，迭代次数在30左右时达到最优解，阈值也在不断的迭代中达到最合适的值。

EM算法的主要目的是提供一个简单的迭代算法计算后验密度函数，它的最大优点是简单和稳定，但容易陷入局部最优。

八、 参考文献

[1] 张林兰;电子市场中的双边同步自动协商研究[D].武汉：华中科技大学，2010年

[2] 乌力吉;互补问题的乘子法[D].内蒙古：内蒙古大学，2004年

[3] 杨凯;基于期望理论的我国巨灾债券定价模型研究[D].哈尔滨：哈尔滨工业大学，2008年

[4] 彭飞;基于行为金融的资产选择模型研究[D].成都：西南交通大学，2005年

[5] 孔令臣;对称锥互补问题的互补函数和价值函数研究[D].北京：北京交通大学，2007年

[6] 赵文玲;约束优化问题的一类罚函数方法与误差界理论及其应用[D].大连：大连理工大学，2008年

[7] 张继军;信用风险度量及其与宏观经济关系研究[D].天津：天津大学，2009年

[8] 刘伟;保险风险模型的最优控制问题研究[D].武汉：武汉大学，2010年

[9] 葛焰明;考虑转移因素的航空收益管理[D].上海：复旦大学，2011年