**关于HMM的模式识别学习报告**

报告人：何峙

学习：212155122

专业：大数据与人工智能

**前言**

模式识别这门课，从最开始的决策论，再到极大似然估计、贝叶斯估计、非参数估计等各种估计技术，归根到底，无不源于贝叶斯的公式：



而隐马尔可夫模型（以下简称HMM）是学习这么课程目前为止，对贝叶斯理论诠释得最综合的一个模型实例，是一种动态的贝叶斯网络，它也是我感受最深，觉得不可思议的技术。特地深入学习，以此作为该课程学习的阶段性总结。

**HMM的一些研究现状**

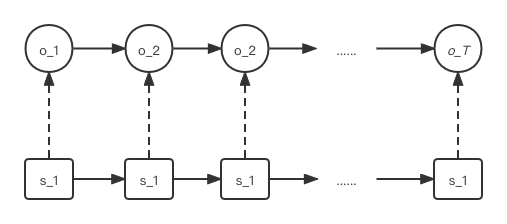
HMM的现实应用研究相当广泛，尤其适合需要处理序列问题的场景，从语音识别、自然语言处理等基础学科，再到上层的金融、生物、机械工程、网络工程的应用领域均有涉及。

华南理工大学的萧超武[1]等提出的一种基于双层HMM的驾驶模式识别方法，认为驾驶行为是一系列驾驶意图的先后序列，利用传感器数据作为最初的观测数据输入到下层HMM，训练出表征隐藏状态的驾驶操作序列（如猛加油、滑行、急刹车，等），再将这些驾驶操作序列最为上层HMM观察序列的输入，训练出表征上层隐藏序列的驾驶意图（准备、操作、恢复等状态），最后得出什么样的驾驶模式能更有效节能的结论，十分具有现实意义。  
 大连理工大学的瞿晓娟[2]等研究使用HMM构建驾驶疲劳识别模型。首先提取人体生物电信号——EGG，然后提取EGG的各个能量比特征指标作为观测序列，将隐藏状态分为“清醒”和“疲劳”两种类别，利用Baum-Welch算法对HMM进行训练，识别出驾驶人的疲劳状态水平，进一步完善了疲劳驾驶识别的工具集。

**HMM理论回顾**

**关于HMM的定义**

HMM是在马尔可夫链的基础上发展起来的，是一种序列生成模型。定义观测序列集合为O = {o1, o2, o3, …, oM}，隐藏状态集为S = {s1, s2, …, sN}，那么HMM的序列结构类似于概率图，如Fig.1所示。



**Fig.1** HMM序列基本结构

Fig.1的第一行横向箭头表示观测序列的生成，第二行横向箭头表示状态之间的转移，虚线箭头表示当前状态产生某个观测。

接着定义HMM的参数θ = {Α, Β, π}，其中：

* A为概率转移概率矩阵：

A = [aij]NxN，aij = P(it+1 = sj | it = si)

表示由状态i转移到状态j的概率，其中i, j = 1, 2, …, N

* B为观察生成概率矩阵：

B = [bj(k)]NxM，bj(k) = P(vt = ok | it = sj )

其中k = 1, 2, …, M，而j = 1, 2, …, N

* π为初始状态向量：

πi = P(i1 = si)， 其中i = 1, 2, … , N

重要的是，HMM作了两个基本假设：

1. 任意时刻t的状态仅取决于其前一时刻的状态，即：

P(it | it-1, vt-1, it-2, vt-2, …, i1, v1) = P(it | it-1)

1. 任意时刻的观测只取决于该时刻的状态，即：

P(vt | it-1, vt-1, it-2, vt-2, …, i1, v1) = P(vt | it)

**研究HMM的三个基本问题：**

1. 估值问题

已知模型参数，给定观测序列，计算产生这个序列的概率。如果使用穷举法，其时间复杂度将达到O(T \* NT)，其中T为序列长度，N为状态数，计算量太大一般不采用。使用前向算法或后向算法，可使时间复杂度降为O(T \* N2)。

例如前向算法，先定义前向概率αt(i)：在已知模型参数条件下，在t时刻观察序列为v1,v2,v3,…,vt且此时状态为si的概率：αt(i) = P(v1,v2,v3,…,vt,it = si | θ)

那么，前向算法过程可描述如下[3]：

|  |
| --- |
| 输入：HMM参数θ，观察序列O  输出：观测序列概率P(O|θ)   1. 初始化：t=1时刻的前向概率：α1(i) 2. 迭代，对t=1,2,…,T-1,求：      1. 终止： |

后向算法类似，可参考本课程课件，这里不再赘述。

1. 解码问题

已知模型参数，给定观测序列，求这个观测序列对应的隐藏状态序列。解决此问题多使用Viterbi算法，它是一个动态规划的算法，从t=1开始递推计算状态为si的各条分路径的最大概率，每个时刻选择出概率最大的路径，然后回溯，将路径上各个结点连接起来即得到最优路径，这条路径即隐藏状态序列。具体算法描述可参考李航的《统计学习方法》第二版[3]。

1. 学习问题

此问题要求解HMM的参数。分成两种情况：

（1）给定观测序列及对应的隐藏状态序列。此时只需要进行监督学习即可，使用极大似然估计即可求得参数，如求状态转移概率:



其中cij为出现状态i转移到状态j这种情况的频次。

（2）只给定观测序列，对应的隐藏状态序列未知。这时可采用Baum-Welch算法。该算法本质上是EM算法。指定参数的初始值θ0，将其和观测序列代入Q函数，然后最大化这个函数，求得一组参数θ\*k，使得P(O|θ\*k) > P(O|θ\*k-1)，不断迭代（其中k是迭代的轮次），使得模型参数收敛。具体算法实现过程可参考李航的《统计学习方法》第二版[3]。

**HMM上机实验**

现以一个中文分词应用验证HMM的三个基本问题。

假如现有一句话（字序列）：

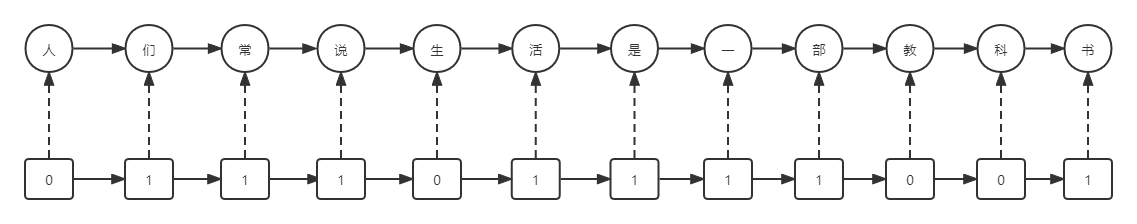
“人们常说生活是一部教科书”，

已分词如下：

['人们', '常', '说', '生活', '是', '一', '部', '教科书']，

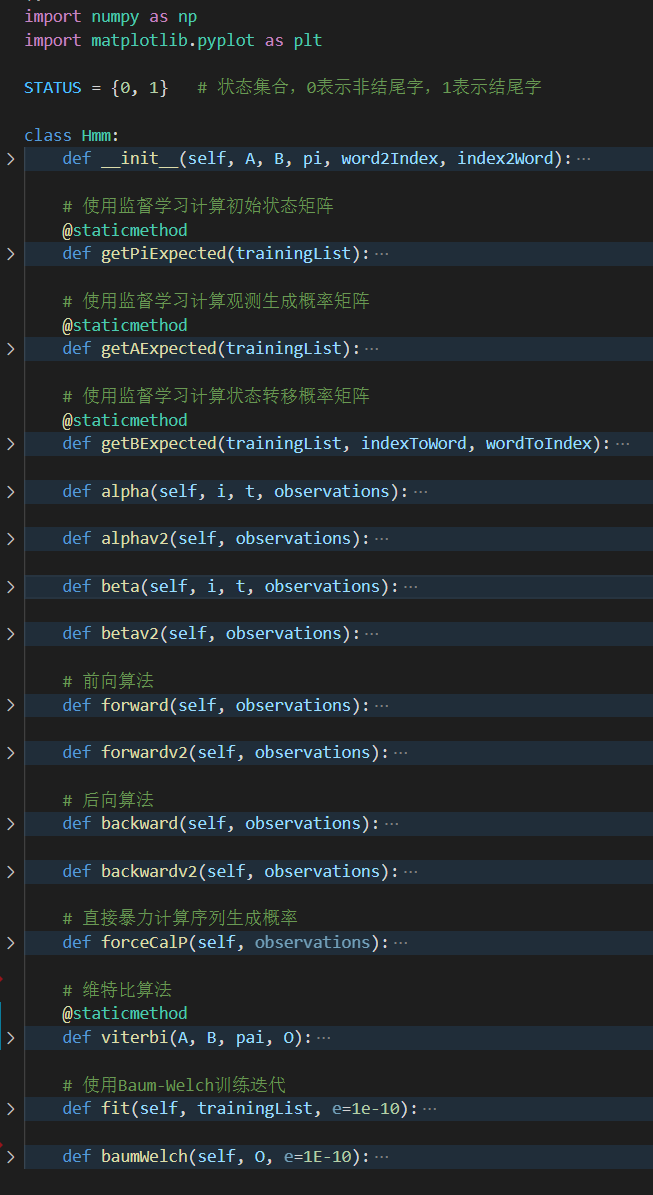
定义状态集S = {0, 1}，0表示非终结字，1表示终结字，那么对句子分词实际上就是找到句子中标记为1的那个字，然后切分即可：

“人/0们/1常/1说/1生/0活/1是/1一/1部/1教/0科/0书/1”，



**Fig.2** 用HMM模型表示一句中文的分词结构

现通过编写代码方式验证这三个问题[[1]](#footnote-0)。



**Fig.3** HMM代码图示

1. 学习问题

|  |  |  |
| --- | --- | --- |
| π | A | B |
|  |  |  |

**Table 1** 由监督学习得出HMM的参数

**Table 2** 前向算法和后向算法计算结果

|  |  |  |
| --- | --- | --- |
|  | 前向算法 | 后向算法 |
| P(O | θ ) | 3.7272e-13 | 3.7272e-13 |

**Table 3** 估值问题中使用前向算法，计算方式分别使用递归运算和矩阵计算的对比

|  |  |  |
| --- | --- | --- |
|  | 递归运算 | 矩阵运算 |
| P(O | θ) | 3.7272e-13 | 3.7272e-13 |
| 耗时（单位：秒） | 0.010911 | 0.000167 |

Table 2为分别使用前向算法和后向算法的计算结果，计算结果是一致的。

代码中使用了两种方式计算前向概率/后向概率：递归和矩阵运算。使用递归运算时，发现观测序列越长，计算速度越慢，譬如计算一个观测序列长度为25的前向概率，用递归方法，计算耗时竟然达到48秒之久。而改用numpy库的矩阵运算，计算速度有十分明显的提升。从Table 3可以看到，计算本实验的句子序列，使用矩阵运算比递归运算快几乎66倍，而两种方法的计算结果也是一致的。

1. 解码问题

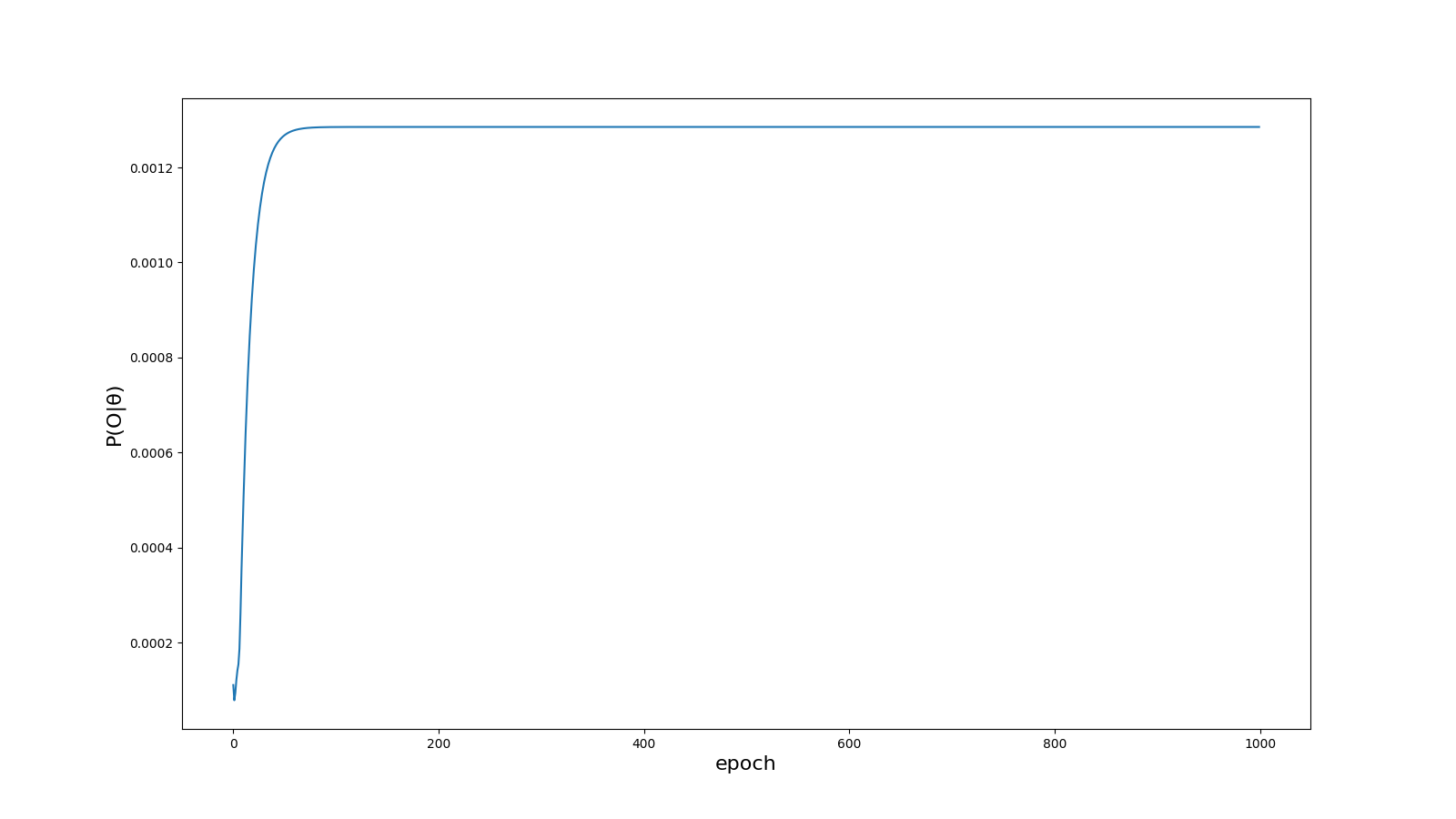
1637339840(1)

**Fig.4** 使用Viterbi算法计算状态序列结果

使用Table 1的参数进行解码问题，得出结果如图Fig.4所示。对比Fig.2，分词结果是一致的。

1. 学习问题

将句子“人们常说生活是一部教科书”直接输入到HMM模型中用Baum-Welch算法进行学习。



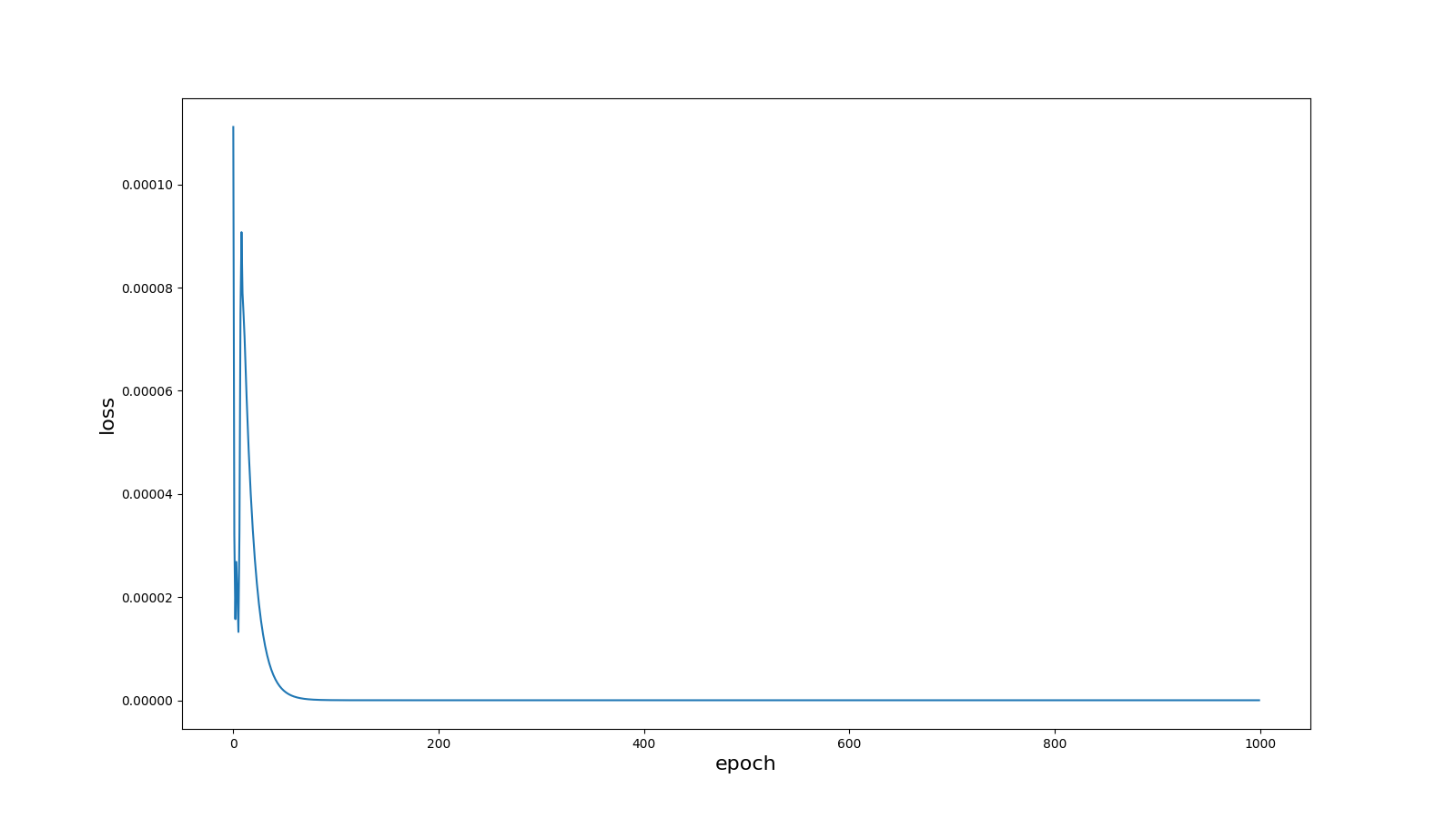
**Fig.5** P(O | θ)不断迭代的值

Fig.5 表示观测序列的概率随着模型参数不断迭代而不断增大，最终逼近一个最大值，符合EM算法的表现形式，形象说明Baum-Welch算法本质上就是EM算法。

Fig.6 是参数迭代的损失趋势，loss的计算方式为：

loss=|P(O|θ\*k) - P(O|θ\*k-1)|， 其中k为迭代轮数

跟Fig.1的结果是一致的，算法大概迭代到100轮时已经收敛。



**Fig.6** 训练损失趋势

**总结与展望**

**参考文献**

[1] 萧超武. 基于HMM的驾驶模式识别方法研究及应用[D]. 华南理工大学, 2015.

[2] 翟晓娟. 基于HMM的随机驾驶人疲劳状态识别研究[D]. 大连理工大学, 2019.

[3] 李航. 统计学习方法(第2版)[M]: 清华大学出版社, 2019.

1. 相关代码可查验：https://github.com/kevinva/hmmmmmmmmmm [↑](#footnote-ref-0)