隐马尔科夫模型HMM（三）鲍姆-韦尔奇算法求解HMM参数

隐马尔科夫模型HMM（一）HMM模型

隐马尔科夫模型HMM（二）前向后向算法评估观察序列概率

隐马尔科夫模型HMM（三）鲍姆-韦尔奇算法求解HMM参数

隐马尔科夫模型HMM（四）维特比算法解码隐藏状态序列

在本篇我们会讨论HMM模型参数求解的问题，这个问题在HMM三个问题里算是最复杂的。在研究这个问题之前，建议先阅读这个系列的前两篇以熟悉HMM模型和HMM的前向后向算法，以及EM算法原理总结，这些在本篇里会用到。在李航的《统计学习方法》中，这个算法的讲解只考虑了单个观测序列的求解，因此无法用于实际多样本观测序列的模型求解，本文关注于如何使用多个观测序列来求解HMM模型参数。

1. HMM模型参数求解概述

HMM模型参数求解根据已知的条件可以分为两种情况。

第一种情况较为简单，就是我们已知D个长度为T的观测序列和对应的隐藏状态序列，即$\{(O_1, I_1), (O_2, I_2), \ldots (O_D, I_D)\}$是已知的，此时我们可以很容易的用最大似然来求解模型参数。

假设样本从隐藏状态q_i转移到q_j的频率计数是A_{ij},那么状态转移矩阵求得为：

$$A = \Big[a_{ij}\Big], \text{其中} a_{ij} = \frac{A_{ij}}{\sum\limits_{s=1}^{N} A_{is}}$$

假设样本隐藏状态为q_j且观测状态为v_k的频率计数是B_{jk},那么观测状态概率矩阵为：

$$B = \Big[b_j(k)\Big], \text{其中} b_j(k) = \frac{B_{jk}}{\sum\limits_{s=1}^{M} B_{js}}$$

假设所有样本中初始隐藏状态为q_i的频率计数为$C(i)$,那么初始概率分布为：

$$\Pi = \pi(i) = \frac{C(i)}{\sum\limits_{s=1}^{N} C(s)}$$

可见第一种情况下求解模型还是很简单的。但是在很多时候，我们无法得到HMM样本观察序列对应的隐藏序列，只有D个长度为T的观测序列，即$\{(O_1), (O_2), \ldots (O_D)\}$是已知的，此时我们能不能求出合适的HMM模型参数呢？这就是我们的第二种情况，也是我们本文要讨论的重点。它的解法最常用的是鲍姆-韦尔奇算法，其实就是基于EM算法的求解，只不过鲍姆-韦尔奇算法出现的时代，EM算法还没有被抽象出来，所以我们本文还是说鲍姆-韦尔奇算法。

2. 鲍姆-韦尔奇算法原理

鲍姆-韦尔奇算法原理既然使用的就是EM算法的原理，那么我们需要在E步求出联合分布$P(O,I|\lambda)$基于条件概率$P(I|O,\overline{\lambda})$的期望，其中$\overline{\lambda}$为当前的模型参数，然后再M步最大化这个期望，得到更新的模型参数λ。接着不停的进行EM迭代，直到模型参数的值收敛为止。

首先来看看E步，当前模型参数为$\overline{\lambda}$, 联合分布$P(O,I|\lambda)$基于条件概率$P(I|O,\overline{\lambda})$的期望表达式为：

$$L(\lambda, \overline{\lambda}) = \sum\limits_{I}P(I|O,\overline{\lambda})logP(O,I|\lambda)$$

在M步，我们极大化上式，然后得到更新后的模型参数如下：

$$\overline{\lambda} = arg\;\max_{\lambda}\sum\limits_{I}P(I|O,\overline{\lambda})logP(O,I|\lambda)$$

通过不断的E步和M步的迭代，直到$\overline{\lambda}$收敛。下面我们来看看鲍姆-韦尔奇算法的推导过程。

3. 鲍姆-韦尔奇算法的推导

公告

★珠江追梦，饮岭南茶，恋鄂北家★
昵称：刘建平Pinard
园龄：1年5个月
粉丝：1057
关注：13
+加关注

< 2018年3月 >

日	一	二	三	四	五	六
25	26	27	28	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	31
1	2	3	4	5	6	7

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

随笔分类(101)

0040. 数学统计学(4)
0081. 机器学习(62)
0082. 深度学习(10)
0083. 自然语言处理(23)
0121. 大数据挖掘(1)
0122. 大数据平台(1)
0123. 大数据可视化

随笔档案(101)

2017年8月 (1)
2017年7月 (3)
2017年6月 (8)
2017年5月 (7)
2017年4月 (5)
2017年3月 (10)
2017年2月 (7)
2017年1月 (13)
2016年12月 (17)
2016年11月 (22)
2016年10月 (8)

常去的机器学习网站

52 NLP
Analytics Vidhya

我们的训练数据为$\{(O_1, I_1), (O_2, I_2), \ldots (O_D, I_D)\}$其中任意一个观测序列$O_d = \{o_1^{(d)}, o_2^{(d)}, \ldots o_T^{(d)}\}$,其对应的未知的隐藏状态序列表示为：$I_d = \{i_1^{(d)}, i_2^{(d)}, \ldots i_T^{(d)}\}$

首先看鲍姆-韦尔奇算法的E步，我们需要先计算联合分布$P(O, I|\lambda)$的表达式如下：

$$P(O,I|\lambda) = \prod_{d=1}^D \pi_{i_1^{(d)}} b_{i_1^{(d)}}(o_1^{(d)}) a_{i_1^{(d)} i_2^{(d)}} b_{i_2^{(d)}}(o_2^{(d)}) \ldots a_{i_{T-1}^{(d)} i_T^{(d)}} b_{i_T^{(d)}}(o_T^{(d)})$$

我们的E步得到的期望表达式为：

$$L(\lambda, \overline{\lambda}) = \sum_I P(I|O,\overline{\lambda}) logP(O,I|\lambda)$$

在M步我们要极大化上式。由于$P(I|O,\overline{\lambda}) = P(I,O|\overline{\lambda})/P(O|\overline{\lambda})$,而$P(O|\overline{\lambda})$是常数，因此我们要极大化的式子等价于：

$$\overline{\lambda} = arg\;\max_{\lambda}\sum_I P(O,I|\overline{\lambda}) logP(O,I|\lambda)$$

我们将上面$P(O,I|\lambda)$的表达式带入我们的极大化式子，得到的表达式如下：

$$\overline{\lambda} = arg\;\max_{\lambda}\sum_{d=1}^D\sum_I P(O,I|\overline{\lambda})(log\pi_{i_1} + \sum_{t=1}^{T-1}log\;a_{i_t}a_{i_{t+1}} + \sum_{t=1}^T b_{i_t}(o_t))$$

我们的隐藏模型参数$\lambda = (A,B,\Pi)$,因此下面我们只需要对上式分别对A,B,Π求导即可得到我们更新的模型参数$\overline{\lambda}$

首先我们看看对模型参数Π的求导。由于Π只在上式中括号里的第一部分出现，因此我们对于Π的极大化式子为：

$$\overline{\pi_i} = arg\;\max_{\pi_{i_1}} \sum_{d=1}^D\sum_I P(O,I|\overline{\lambda})log\pi_{i_1} = arg\;\max_{\pi_i} \sum_{d=1}^D\sum_{i=1}^N P(O,i_1^{(d)} =i|\overline{\lambda})log\pi_i$$

由于π_i还满足$\sum_{i=1}^N\pi_i =1$, 因此根据拉格朗日子乘法，我们得到π_i要极大化的拉格朗日函数为：

$$arg\;\max_{\pi_i}\sum_{d=1}^D\sum_{i=1}^N P(O,i_1^{(d)} =i|\overline{\lambda})log\pi_i + \gamma(\sum_{i=1}^N\pi_i -1)$$

其中，γ为拉格朗日系数。上式对π_i求偏导数并令结果为0， 我们得到：

$$\sum_{d=1}^D P(O,i_1^{(d)} =i|\overline{\lambda}) + \gamma\pi_i = 0$$

令i分别等于从1到N，从上式可以得到N个式子，对这N个式子求和可得：

$$\sum_{d=1}^D P(O|\overline{\lambda}) + \gamma = 0$$

从上两式消去γ,得到π_i的表达式为：

$$\pi_i = \frac{\sum\limits_{d=1}^D P(O,i_1^{(d)} =i|\overline{\lambda})}{\sum\limits_{d=1}^D P(O|\overline{\lambda})} = \frac{\sum\limits_{d=1}^D P(O,i_1^{(d)} =i|\overline{\lambda})}{DP(O|\overline{\lambda})} = \frac{\sum\limits_{d=1}^D P(i_1^{(d)} =i|O, \overline{\lambda})}{D} = \frac{\sum\limits_{d=1}^D P(i_1^{(d)} =i|O^{(d)}, \overline{\lambda})}{D}$$

利用我们在隐马尔科夫模型HMM（二）前向后向算法评估观察序列概率里第二节中前向概率的定义可得：

$$P(i_1^{(d)} =i|O^{(d)}, \overline{\lambda}) = \gamma_1^{(d)}(i)$$

因此最终我们在M步π_i的迭代公式为：

$$\pi_i = \frac{\sum\limits_{d=1}^D\gamma_1^{(d)}(i)}{D}$$

现在我们来看看A的迭代公式求法。方法和Π的类似。由于A只在最大化函数式中括号里的第二部分出现，而这部分式子可以整理为：

$$\sum_{d=1}^D\sum_I\sum_{t=1}^{T-1}P(O,I|\overline{\lambda})log\;a_{i_t}a_{i_{t+1}} = \sum_{d=1}^D\sum_{i=1}^N\sum_{j=1}^N\sum_{t=1}^{T-1}P(O,i_t^{(d)} = i, i_{t+1}^{(d)} = j|\overline{\lambda})log\;a_{ij}$$

机器学习库
机器学习路线图
深度学习进阶书
深度学习入门书

积分与排名

积分 - 298466
排名 - 614

阅读排行榜

1. 梯度下降（Gradient Descent）小结(94585)
2. 梯度提升树(GBDT)原理小结(45049)
3. 线性判别分析LDA原理总结(30554)
4. scikit-learn决策树算法类库使用小结(27504)
5. 谱聚类（spectral clustering）原理总结(20504)

评论排行榜

1. 梯度提升树(GBDT)原理小结(79)
2. 谱聚类（spectral clustering）原理总结(62)
3. 梯度下降（Gradient Descent）小结(60)
4. 卷积神经网络(CNN)反向传播算法(56)
5. 集成学习之Adaboost算法原理小结(50)

推荐排行榜

1. 梯度下降（Gradient Descent）小结(41)
2. 集成学习原理小结(14)
3. 卷积神经网络(CNN)反向传播算法(14)
4. 集成学习之Adaboost算法原理小结(13)
5. 协同过滤推荐算法总结(11)

由于a_{ij}还满足$\sum\limits_{j=1}^{N}a_{ij} = 1$。和求解$\pi_i$类似，我们可以用拉格朗日子乘法并对$a_{ij}$求导，并令结果为0，可以得到$a_{ij}$的迭代表达式为：

$$a_{ij} = \frac{\sum\limits_{d=1}^D\sum\limits_{t=1}^{T-1}P(O^{(d)}, i_t^{(d)} = i, i_{t+1}^{(d)} = j|\overline{\lambda})}{\sum\limits_{d=1}^D\sum\limits_{t=1}^{T-1}P(O^{(d)}, i_t^{(d)} = i|\overline{\lambda})}$$

利用隐马尔科夫模型HMM（二）前向后向算法评估观察序列概率里第二节中前向概率的定义和第五节$\xi_t(i,j)$的定义可得们在M步a_{ij}的迭代公式为：

$$a_{ij} = \frac{\sum\limits_{d=1}^D\sum\limits_{t=1}^{T-1}\xi_t^{(d)}(i,j)}{\sum\limits_{d=1}^D\sum\limits_{t=1}^{T-1}\gamma_t^{(d)}(i)}$$

现在我们来看看B的迭代公式求法。方法和Π的类似。由于B只在最大化函数式中括号里的第三部分出现，而这部分式子可以整理为：

$$\sum\limits_{d=1}^D\sum\limits_{I}\sum\limits_{t=1}^{T}P(O,I|\overline{\lambda})log\;b_{i_t}(o_t) = \sum\limits_{d=1}^D\sum\limits_{j=1}^N\sum\limits_{t=1}^{T}P(O,i_t^{(d)} = j|\overline{\lambda})log\;b_{j}(o_t)$$

由于$b_{j}(o_t)$还满足$\sum\limits_{k=1}^{M}b_{j}(o_t =v_k) = 1$。和求解$\pi_i$类似，我们可以用拉格朗日子乘法并对$b_{j}(k)$求导，并令结果为0，得到$b_{j}(k)$的迭代表达式为：

$$b_{j}(k) = \frac{\sum\limits_{d=1}^D\sum\limits_{t=1}^{T}P(O,i_t^{(d)} = j|\overline{\lambda})I(o_t^{(d)}=v_k)}{\sum\limits_{d=1}^D\sum\limits_{t=1}^{T}P(O,i_t^{(d)} = j|\overline{\lambda})}$$

其中$I(o_t^{(d)}=v_k)$当且仅当$o_t^{(d)}=v_k$时为1，否则为0. 利用隐马尔科夫模型HMM（二）前向后向算法评估观察序列概率里第二节中前向概率的定义可得$b_{j}(o_t)$的最终表达式为：

$$b_{j}(k) = \frac{\sum\limits_{d=1}^D\sum\limits_{t=1, o_t^{(d)}=v_k}^{T}\gamma_t^{(d)}(i)}{\sum\limits_{d=1}^D\sum\limits_{t=1}^{T}\gamma_t^{(d)}(i)}$$

有了$\pi_i, a_{ij},b_{j}(k)$的迭代公式，我们就可以迭代求解HMM模型参数了。

4. 鲍姆-韦尔奇算法流程总结

这里我们概括总结下鲍姆-韦尔奇算法的流程。

输入： D个观测序列样本$\{(O_1), (O_2), ...(O_D)\}$

输出：HMM模型参数

1)随机初始化所有的$\pi_i, a_{ij},b_{j}(k)$

2) 对于每个样本$d = 1,2,...D$，用前向后向算法计算$\gamma_t^{(d)}(i)$，$\xi_t^{(d)}(i,j), t =1,2...T$

3）更新模型参数：

$$\pi_i = \frac{\sum\limits_{d=1}^D\gamma_1^{(d)}(i)}{D}$$

$$a_{ij} = \frac{\sum\limits_{d=1}^D\sum\limits_{t=1}^{T-1}\xi_t^{(d)}(i,j)}{\sum\limits_{d=1}^D\sum\limits_{t=1}^{T-1}\gamma_t^{(d)}(i)}$$

$$b_{j}(k) = \frac{\sum\limits_{d=1}^D\sum\limits_{t=1, o_t^{(d)}=v_k}^{T}\gamma_t^{(d)}(i)}{\sum\limits_{d=1}^D\sum\limits_{t=1}^{T}\gamma_t^{(d)}(i)}$$

4) 如果$\pi_i, a_{ij},b_{j}(k)$的值已经收敛，则算法结束，否则回到第2）步继续迭代。

以上就是鲍姆-韦尔奇算法的整个过程。

（欢迎转载，转载请注明出处。欢迎沟通交流： pinard.liu@ericsson.com）

« 上一篇：隐马尔科夫模型HMM（二）前向后向算法评估观察序列概率
» 下一篇：隐马尔科夫模型HMM（四）维特比算法解码隐藏状态序列

评论列表

#1楼 2017-06-10 23:46 ChokCoco

....厉害了

支持(0) 反对(0)

#2楼 2017-11-11 23:05 指尖的hacker

这个讲解的还是不错的。很好，博主，你的理论知识真的很棒，理解的很深刻。向你学习。

支持(0) 反对(0)

#3楼 2017-11-23 02:04 Zuozuohao

"我们的训练数据为{(O1,I1),(O2,I2),...(OD,ID)}，其中任意一个观测序列Od={o1(d),o2(d),...oT(d)},其对应的未知的隐藏状态序列表示为："

这个位置应该是观测序列是Id={o1(d),o2(d),...oT(d)}

支持(0) 反对(0)

#4楼 2017-11-23 02:09 Zuozuohao

”其对应的未知的隐藏状态序列表示为：“
博主这个位置的隐藏状态应该是Id而不是Od，还有就是上面那条请忽略哈，手滑了。

支持(0) 反对(0)

#5楼[楼主] 2017-11-23 10:11 刘建平Pinard

@ Zuozuohao
感谢指出错误，已更正。

支持(0) 反对(0)

#6楼 2017-12-13 14:21 肖同尧

你好，博主的博客十分精彩。对于本篇我有个问题，M步对π求偏导的时候，是不是还应该再对logπ求导然后相乘呢？

支持(0) 反对(0)

#7楼[楼主] 2017-12-13 16:29 刘建平Pinard

@ 肖同尧
对于这个式子：

$$\sum_{d=1}^{D}\sum_{i=1}^{N}P(O,i_1^{(d)}=i|\overline{\lambda})log\pi_i+\gamma(\sum_{i=1}^{N}\pi_i-1)$$

求导后并令导数为0得到的是：

$$\sum_{d=1}^{D}\sum_{i=1}^{N}P(O,i_1^{(d)}=i|\overline{\lambda})\frac{1}{\pi_i}+\gamma=0$$

转化下就是文中的求导后的式子。

里面已经对$log\pi$求导。

支持(0) 反对(0)

#8楼 2017-12-20 20:06 chenlf2010

博主写的很详细，但是有一个地方有疑问。
在M步我们要极大化上式。由于P(I|O,λ¯¯¯)=P(I,O|λ¯¯¯)/P(O|λ¯¯¯)P(I|O,λ¯)=P(I,O|λ¯)/P(O|λ¯),而P(O|λ¯¯¯)P(O|λ¯)是常数，因此我们要极大化的式子等价于：
λ¯¯¯=argmaxλ∑d=1D∑IP(O,I|λ¯¯¯)logP(O,I|λ)
这里虽然对于同一个Od来说P(O|λ¯¯¯)P(O|λ¯)是同一常数，但是不同的Od，P(O|λ¯¯¯)P(O|λ¯)应是不同常数，无法将P(O|λ¯¯¯)P(O|λ¯)提到∑外，因此，一般来说
L(λ,λ¯¯¯)=∑d=1D∑IP(I|O,λ¯¯¯)logP(O,I|λ)应该不等于
λ¯¯¯=argmaxλ∑d=1D∑IP(O,I|λ¯¯¯)logP(O,I|λ)

支持(0) 反对(0)

#9楼 2017-12-20 20:11 chenlf2010

不会编辑公式，乱码了。
我不知道我的理解对不对，请博主指正。

支持(0) 反对(0)

#10楼 2017-12-20 23:00 chenlf2010

看来博主确实错了，最后的结论是正确的，但是推导的过程中有一个瑕疵。问题就出在我所指的地方：
在M步我们要极大化上式。由于$P(I|O,\lambda)=P(I,O|\overline{\lambda})/P(O|\overline{\lambda})$而$P(O|\overline{\lambda})$是常数，因此我们要极大化的式子等价于：

$$\overline{\lambda} = arg\ \max_{\lambda}\sum_{d=1}^{D}\sum_{I}P(O,I|\overline{\lambda})logP(O,I|\lambda)$$

这篇文章是我看过的HMM学习算法中写的最详细的，有瑕疵确实有遗憾，还是希望博主修订一下。

支持(0) 反对(0)

#11楼[楼主] 2017-12-21 10:46 刘建平Pinard

@ chenlf2010
你好，这里的确写的有点混乱，本来O代表的是所有样本的观测值，但是我又在前面加了一个累加的$\sum_{d=1}^{D}$，这样就矛盾了，也会出现你说的不同的O_d，$P(O_d|\overline{\lambda})$不同。但是实际上此时由于O是一个整体，所以$P(O|\overline{\lambda})$是相同的。

感谢指正错误，已经修改。因为包括李航的书在内的大多数书都是单样本的B-W算法，我这里是自己推导的，的确有考虑不周的地方。

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:
· 拼多多为什么能爆红？
· ICO收割炒币者，交易所收割ICO，谁才是韭菜？
· 牙买加政府积极探索使用开源软件
· 微软Azure Stack混合云4月国内正式商用
· Linux基金会宣布开放物联网ACRN管理程序
» 更多新闻...