

刘建平Pinard

十年码农，对数学统计学，数据挖掘，机器学习，大数据平台，大数据平台应用开发，大数据可视化感兴趣。

博客园 首页 新随笔 联系 订阅 管理

隐马尔科夫模型HMM（四）维特比算法解码隐藏状态序列

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模型最后一个问题的求解，即给定模型和观测序列，求给定观测序列条件下，最可能出现的对应的隐藏状态序列。在阅读本篇前，建议先阅读这个系列的第一篇以熟悉HMM模型。

HMM模型的解码问题最常用的算法是维特比算法，当然也有其他的算法可以求解这个问题。同时维特比算法是一个通用的求序列最短路径的动态规划算法，也可以用于很多其他问题，比如之前讲到的文本挖掘的分词原理中我们讲到了单独用维特比算法来做分词。

本文关注于用维特比算法来解码HMM的的最可能隐藏状态序列。

1. HMM最可能隐藏状态序列求解概述

在HMM模型的解码问题中，给定模型 $\lambda = (A, B, \Pi)$ 和观测序列 $O = \{o_1, o_2, \dots, o_T\}$ 求给定观测序列O条件下，最可能出现的对应的状态序列 $I^* = \{i_1^*, i_2^*, \dots, i_T^*\}$ 即 $P(I^*|O)$ 要最大化。

一个可能的近似解法是求出观测序列O在每个时刻t最可能的隐藏状态 $i_t^*$ 然后得到一个近似的隐藏状态序列 $I^* = \{i_1^*, i_2^*, \dots, i_T^*\}$ 。要这样近似求解不难，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中第五节的定义：在给定模型 $\lambda$ 和观测序列O时，在时刻t处于状态 $q_i$ 的概率是 $\gamma_t(i)$ ，这个概率可以通过HMM的前向算法与后向算法计算。这样我们有：

$$i_t^* = \arg \max_{1 \leq i \leq N} [\gamma_t(i)], t = 1, 2, \dots, T$$

近似算法很简单，但是却不能保证预测的状态序列是整体是最可能的状态序列，因为预测的状态序列中某些相邻的隐藏状态可能存在转移概率为0的情况。

而维特比算法可以将HMM的状态序列作为一个整体来考虑，避免近似算法的问题，下面我们来看看维特比算法进行HMM解码的方法。

2. 维特比算法概述

维特比算法是一个通用的解码算法，是基于动态规划的求序列最短路径的方法。在文本挖掘的分词原理中我们已经讲到了维特比算法的一些细节。

既然是动态规划算法，那么就需要找到合适的局部状态，以及局部状态的递推公式。在HMM中，维特比算法定义了两个局部状态用于递推。

第一个局部状态是在时刻t隐藏状态为i所有可能的状态转移路径 $i_1, i_2, \dots, i_{t-1}$ 中的概率最大值。记为 $\delta_t(i)$ ：

$$\delta_t(i) = \max_{i_1, i_2, \dots, i_{t-1}} P(i_t = i, i_1, i_2, \dots, i_{t-1}, o_t, o_{t-1}, \dots, o_1 | \lambda), i = 1, 2, \dots, N$$

由 $\delta_t(i)$ 的定义可以得到 $\delta$ 的递推表达式：

$$\delta_{t+1}(i) = \max_{i_1, i_2, \dots, i_t} P(i_{t+1} = i, i_1, i_2, \dots, i_t, o_{t+1}, o_t, \dots, o_1 | \lambda) \tag{1}$$

$$= \max_{1 \leq j \leq N} [\delta_t(j) a_{ji}] b_i(o_{t+1}) \tag{2}$$

第二个局部状态由第一个局部状态递推得到。我们定义在时刻t隐藏状态为i的所有单个状态转移路径 $(i_1, i_2, \dots, i_{t-1}, i)$ 中概率最大的转移路径中第t-1个节点的隐藏状态为 $\Psi_t(i)$ 其递推表达式可以表示为：

$$\Psi_t(i) = \arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}]$$

有了这两个局部状态，我们就可以从时刻0一直递推到时刻T，然后利用 $\Psi_t(i)$ 记录的前一个最可能的状态节点回溯，直到找到最优的隐藏状态序列。

3. 维特比算法流程总结

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

现在我们来总结下维特比算法的流程：

输入：HMM模型 $\lambda = (A, B, \Pi)$  观测序列 $O = (o_1, o_2, \dots, o_T)$

输出：最有可能的隐藏状态序列 $I^* = \{i_1^*, i_2^*, \dots, i_T^*\}$

1) 初始化局部状态：

$$\delta_1(i) = \pi_i b_i(o_1), \quad i = 1, 2, \dots, N$$
$$\Psi_1(i) = 0, \quad i = 1, 2, \dots, N$$

2) 进行动态规划递推时刻 $t = 2, 3, \dots, T$ 时刻的局部状态：

$$\delta_t(i) = \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}] b_i(o_t), \quad i = 1, 2, \dots, N$$
$$\Psi_t(i) = \arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}], \quad i = 1, 2, \dots, N$$

3) 计算时刻 $T$ 最大的 $\delta_T(i)$ 即为最可能隐藏状态序列出现的概率。计算时刻 $T$ 最大的 $\Psi_T(i)$ 即为时刻 $T$ 最可能的隐藏状态。

$$P^* = \max_{1 \leq j \leq N} \delta_T(j)$$
$$i_T^* = \arg \max_{1 \leq j \leq N} [\delta_T(j)]$$

4) 利用局部状态 $\Psi(i)$ 开始回溯。对于 $t = T - 1, T - 2, \dots, 1$

$$i_t^* = \Psi_{t+1}(i_{t+1}^*)$$

最终得到最有可能的隐藏状态序列 $I^* = \{i_1^*, i_2^*, \dots, i_T^*\}$

## 4. HMM维特比算法求解实例

下面我们仍然用隐马尔科夫模型HMM（一）HMM模型中盒子与球的例子来看看HMM维特比算法求解。

我们的观察集合是：

$$V = \{\text{红}, \text{白}\}, \quad M = 2$$

我们的状态集合是：

$$Q = \{\text{盒子1}, \text{盒子2}, \text{盒子3}\}, \quad N = 3$$

而观察序列和状态序列的长度为3。

初始状态分布为：

$$\Pi = (0.2, 0.4, 0.4)^T$$

状态转移概率分布矩阵为：

$$A = \begin{pmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{pmatrix}$$

观测状态概率矩阵为：

$$B = \begin{pmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{pmatrix}$$

球的颜色的观测序列：

$$O = \{\text{红}, \text{白}, \text{红}\}$$

按照我们上一节的维特比算法，首先需要得到三个隐藏状态在时刻1时对应的各自两个局部状态，此时观测状态为1：

$$\delta_1(1) = \pi_1 b_1(o_1) = 0.2 \times 0.5 = 0.1$$
$$\delta_1(2) = \pi_2 b_2(o_1) = 0.4 \times 0.4 = 0.16$$
$$\delta_1(3) = \pi_3 b_3(o_1) = 0.4 \times 0.7 = 0.28$$
$$\Psi_1(1) = \Psi_1(2) = \Psi_1(3) = 0$$

现在开始递推三个隐藏状态在时刻2时对应的各自两个局部状态，此时观测状态为2：

$$\delta_2(1) = \max_{1 \leq j \leq 3} [\delta_1(j) a_{j1}] b_1(o_2) = \max_{1 \leq j \leq 3} [0.1 \times 0.5, 0.16 \times 0.3, 0.28 \times 0.2] \times 0.5 = 0.028$$
$$\Psi_2(1) = 3$$
$$\delta_2(2) = \max_{1 \leq j \leq 3} [\delta_1(j) a_{j2}] b_2(o_2) = \max_{1 \leq j \leq 3} [0.1 \times 0.2, 0.16 \times 0.5, 0.28 \times 0.3] \times 0.6 = 0.0504$$

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)

$$\Psi_2(2) = 3$$

$$\delta_2(3) = \max_{1 \leq j \leq 3} [\delta_1(j) a_{j3}] b_3(o_2) = \max_{1 \leq j \leq 3} [0.1 \times 0.3, 0.16 \times 0.2, 0.28 \times 0.5] \times 0.3 = 0.042$$

$$\Psi_2(3) = 3$$

继续递推三个隐藏状态在时刻3时对应的各自两个局部状态，此时观测状态为1：

$$\delta_3(1) = \max_{1 \leq j \leq 3} [\delta_2(j) a_{j1}] b_1(o_3) = \max_{1 \leq j \leq 3} [0.028 \times 0.5, 0.0504 \times 0.3, 0.042 \times 0.2] \times 0.5 = 0.00756$$

$$\Psi_3(1) = 2$$

$$\delta_3(2) = \max_{1 \leq j \leq 3} [\delta_2(j) a_{j2}] b_2(o_3) = \max_{1 \leq j \leq 3} [0.028 \times 0.2, 0.0504 \times 0.5, 0.042 \times 0.3] \times 0.4 = 0.01008$$

$$\Psi_3(2) = 2$$

$$\delta_3(3) = \max_{1 \leq j \leq 3} [\delta_2(j) a_{j3}] b_3(o_3) = \max_{1 \leq j \leq 3} [0.028 \times 0.3, 0.0504 \times 0.2, 0.042 \times 0.5] \times 0.7 = 0.0147$$

$$\Psi_3(3) = 3$$

此时已经到了最后的时刻，我们开始准备回溯。此时最大概率为 $\delta_3(3)$ ，从而得到 $i_3^* = 3$

由于 $\Psi_3(3) = 3$ 所以 $i_2^* = 3$  而又由于 $\Psi_2(3) = 3$ 所以 $i_1^* = 3$  从而得到最终的最可能的隐藏状态序列为：(3, 3, 3)

## 5. HMM模型维特比算法总结

如果大家看过之前写的文本挖掘的分词原理中的维特比算法，就会发现这两篇之中的维特比算法稍有不同。主要原因是在中文分词时，我们没有观察状态和隐藏状态的区别，只有一种状态。但是维特比算法的核心是定义动态规划的局部状态与局部递推公式，这一点在中文分词维特比算法和HMM的维特比算法是相同的，也是维特比算法的精华所在。

维特比算法也是寻找序列最短路径的一个通用方法，和dijkstra算法有些类似，但是dijkstra算法并没有使用动态规划，而是贪心算法。同时维特比算法仅仅局限于求序列最短路径，而dijkstra算法是通用的求最短路径的方法。

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

分类: 0083. 自然语言处理

标签: 自然语言处理

好文要顶

关注我

收藏该文







刘建平Pinard  
关注 - 13  
粉丝 - 1057

+加关注

1

0

« 上一篇: 隐马尔科夫模型HMM（三）鲍姆-韦尔奇算法求解HMM参数  
» 下一篇: 用hmmlearn学习隐马尔科夫模型HMM

posted @ 2017-06-12 16:57 刘建平Pinard 阅读(1457) 评论(4) 编辑 收藏

### 评论列表

# 1楼 2017-12-12 11:37 liguangchuang

看了很多次都不能理解回溯的过程，如果只是为了寻找最优解，直接正向每一步都找最大值应该是可以的，老师讲解逆向是为了减少噪声，但过程还是不能理解

支持(0) 反对(0)

# 2楼[楼主] 2017-12-12 16:50 刘建平Pinard

@ liguangchuang

你好，首先这是一个动态规划算法，所以首先你需要对动态规划算法有所了解，然后再看这个算法就简单了。

按你的说法，“直接正向每一步都找最大值应该是可以的”，这样就是贪婪法，在这里贪婪法是没法找到最优解的，因为这里每一步取最大值时对应的转移是不一样的。

举个简单的例子，有1->2->3的跳转。每个时间点有A，B两个状态。假设：  
1->2 (A->B:0.8, A->A:0.4, 其他省略)  
2->3(A->B:0.4, A->A:0.9, 其他省略)

如果按照你说的，直接正向每一步都找最大值，那么我们找到的是  
1->2 (A->B:0.8 ) 和2->3(A->A:0.9)

问题是这时我们得到的状态转移是A->B和A->A，这里第二步的状态B和A矛盾了。

支持(1) 反对(0)

# 3楼 2017-12-12 16:55 liguangchuang

@ 刘建平Pinard

我应该是明白了，首先是要找到全局最优的 $P(I|O)$ ，然后才能确定每个状态取值是什么，贪婪算法每一步找最大，最后的结果不一定能达到全局最大，也就是最终的 $P(I|O)$ 不一定最大。

支持(0) 反对(0)

# 4楼[楼主 ] 2017-12-12 17:11 刘建平Pinard

@ liguangchuang

你好，就是这个样子：)

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](#)。**

【推荐】超50万VC++源码：大型组态工控、电力仿真CAD与GIS源码库！

【缅怀】传奇谢幕，回顾霍金76载传奇人生

【推荐】腾讯云校园拼团福利，1核2G服务器10元/月！

【活动】2050 科技公益大会 - 年青人因科技而团聚



**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...](#)



**最新知识库文章：**

- [写给自学者的入门指南](#)
- [和程序员谈恋爱](#)
- [学会学习](#)
- [优秀技术人的管理陷阱](#)
- [作为一个程序员，数学对你到底有多重要](#)

» [更多知识库文章...](#)