# 实验2. 隐马尔科夫模型实践

MG1733092, 张则君，826320663@qq.com

2017年12月6日

## 综述

隐马尔科夫模型是一种有向图模型，节点表示变量，边表示变量依赖关系。它的变量分为观测变量,状态变量两类。其中状态变量之间的转换满足马尔可夫性。因此变量间的概率分布可以表达为

$$P(o_0, s_0, , , o_{n-1}, s_{n-1}) = P(s_0)P(o_0|s_0)\prod_{i=1}^{n-1} P(s_i|s_{i-1})P(o_i|s_i) \tag{1}$$

设状态空间为$S$,观测空间$O$

为求解公式(1)我们定义三个参数：

矩阵A表示状态转移概率矩阵，其中A中元素 $a_{i,j}$=$P(s_j|s_i)$

矩阵B表示发射矩阵，其中 B中元素$b_{i,j}$=$P(o_j|s_i)$

向量$\pi$表示初始状态概率，其中A中元素$\pi_i$=$P(s = s_i)$

本次实验为解决股票涨跌时序问题，采用了隐马尔可夫模型，解决了2个基本问题。

一个问题是对于观测序列求得概率最大的状态序列，我们将在实验一种利用维特比算法根据公式(1)求解。其中S={0,1},O={0,1,2}（0：跌，1：涨，2：平）。

另一个问题是是给定观测序列，估计出参数使得观测序列出现概率最大。我们在实验中利用Baum–Welch算法求解，其中需要用到的前向后向算法将在实验二、三中分别介绍。

## 实验一.

维特比算法采用动态规划思想，求解最大优化问题。在给定股票的观测序列时，根据公式(1)可以得到变量的概率。为了求解最大概率的相应状态序列path，我们需要矩阵$\delta$存储每个观测变量下的每个状态的最大概率，矩阵$\phi$存储每个观测变量下的当前状态的最大概率的前一个状态，以便于回溯求得path。算法看表1。

表 1: viterbi algorithm

| |
|---|
| 输入：a,b,o,$\pi$ |
| 输出：path |
| 过程： |
| 对每一个状态 $i$ |
| $\quad\delta_{i,0}= \pi_i \cdot b_{i,o_0}$ |
| $\quad\phi_{i,0} = 0$ |
| 对每一个观测变量$t = 1,,T$ |
| $\quad$对每一个状态 $i = 0,,N$ |
| $\quad\quad\delta_{i,t}=b_{i,o_t} \cdot \max_j(\delta_{j,t-1} \cdot a_{j,i})$ |
| $\quad\quad\phi_{t,i} = \arg\max_j(\delta_{j,t-1} \cdot a_{j,i})$ |
| $path_t=\arg\max_j(\delta_{j,T-1})$ |
| 对每一个观测变量从t-2到0 |
| $\quad path_t= \phi_{t+1,path_{t+1}}$ |

## 实验二.

前向算法是通过不断由前向后递推得到矩阵$\alpha$其中$\alpha_{i,t}=P(o_0,,o_t,x_t = i|A,B)$,具体算法参见表2forward algorithm。

表 2: forward algorithm

| |
|---|
| 输入：a,b,o,pi |
| 输出：$\alpha$ |
| 过程： |
| 对每个状态$i$ |
| $\quad\alpha_{i,0} = \pi_i \cdot b_{i,o_0}$ |
| 对每一个观测变量$t = 1,,T$ |
| $\quad$对每一个状态 $i = 0,,N-1$ |
| $\quad\quad\alpha_{i,t}=\sum_{j=0}^{N-1} b_{i,o_t} \cdot \alpha_{j,t-1} \cdot a_{j,i}$ |

## 实验三.

后向算法是通过不断由后向前递推得到矩阵$\beta$其中$\beta_{i,t}=P(o_{t+1},,o_{T-1}|A,B,x_t = i)$具体算法参见表3backward algorithm。

表 3: backward algorithm

输入：$a, b, o$

输出：$\beta$

过程：

对每个状态$i$

  $\beta_{i,T-1} = 1$

对每一个观测变量$t = T-2, , 0$

  对每一个状态 $i = 0, , N-1$

    $\beta_{i,t}=\sum_{j=0}^{N-1} b_{j,o_{t+1}} \cdot \beta_{j,t+1} \cdot a_{i,j}$

# 参考文献


[1] 周志华. 机器学习. 清华学出版社, 2016.

[2] https://en.wikipedia.org/wiki/Viterbi_algorithm.

[3] https://en.wikipedia.org/wiki/Forward-backward_algorithm.

[4] https://en.wikipedia.org/wiki/Baum-Welch_algorithm.

[5] https://people.cs.umass.edu/ mccallum/courses/inlp2004a/lect10-hmm2.pdf.