实验2. 隐马尔科夫模型实践

MF1733071，严德美，1312480794@qq.com

2017 年 12 月 6 日

综述

本次实验是一步一步实现隐马尔科夫模型（Hidden Markov Model, HMM）,并应用到金融时序数据分析与预测方面。一般而言，对于一个HMM模型λ=[A,B,π],能根据观察序列推断出隐状态序列，简单的方法计算所有可能的状态序列时间复杂度太高，一种更有效率的利用动态规划的思想是Viterbi算法，Viterbi算法定义一个$\delta_t(i)$，指在时间t时，HMM沿着某一条路径到达S_i,并输出序列为$W_1W_2...W_t$的最大概率。如果HMM的参数未知，则需要通过数据进行学习与训练，对于有监督学习，使用最大似然估计可求出所需参数，对于无监督学习使用Baum–Welch_algorithm进行求解。

$$\delta_t(i) = maxP(Pos_1 \cdots Pos_{t-1}, Pos_t = s_i, w_1 \cdots w_t) \tag{1}$$

初始化：$\delta_1(i) = maxP(Pos_1 = s_i, w_1) = \pi_i b_i(w_1), 1 \leq i \leq N$
迭代：$\delta_{t+1}(j) = maxP(Pos_1 \cdots Pos_t, Pos_{t+1} = s_j, w_1 w_2 \cdots w_{t+1}) \quad 1 \leq j \leq N, 1 \leq t \leq T-1$
迭代结束： $max_i[\delta_T(i)]$
回溯记录最优路径

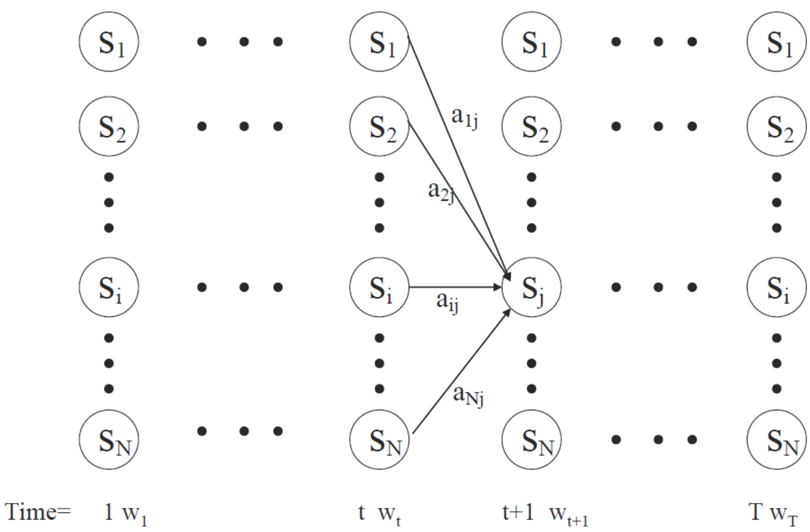

图 1: HMM

实验一.维特比算法

本实验实现维特比算法。

输入：a,b,o,pi

输出：path

function VITERBI(O,S,Π ,Y,A,B):X

for each state i∈ { 1,2,$\cdots$,K} do

$T_1[i,1] = \pi_i \cdot B_{iy_1}$

$T_2[i,1] = 0$

end for

for each observation i=2,3,$\cdots$,T do

for each state j ∈ {1,2, $\cdots$,K} do

$T_1[j,i] \leftarrow \max_k (T_1[k,i-1] \cdot A_{kj} \cdot B_{jy_i})$

$T_2[j,i] \leftarrow \arg\max_k (T_1[k,i-1] \cdot A_{kj} \cdot B_{jy_i})$

end for

end for

$z_T \leftarrow \arg\max_k (T_1[k,T])$

$x_T \leftarrow s_{z_T}$

for $i \leftarrow T, T-1, ..., 2$ do

$z_{i-1} \leftarrow T_2[z_i, i]$

$x_{i-1} \leftarrow s_{z_{i-1}}$

end for

return X

end function

实验二.Forward Algorithm

本实验实现 Forward Algorithm

输入：a,b,o,π

输出：α

Let $\alpha_i(t) = P(Y_1 = y_1, ..., Y_t = y_t, X_t = i|\theta)$, the probability of seeing the $y_1, y_2, ..., y_t$ and being in state at state i time t. This is found recursively:

1.$\alpha_i(1) = \pi_i b_i(y_1)$,

2.$\alpha_i(t+1) = b_i(y_{t+1}) \sum_{j=1}^{N} \alpha_j(t) a_{ji}$.

实验三.Backward Algorithm

本实验实现 Backward Algorithm

Let $\beta_i(t) = P(Y_{t+1} = y_{t+1}, ..., Y_T = y_T | X_t = i, \theta)$ that is the probability of the ending partial sequence $y_{t+1}, ..., y_T$ given starting state i at time t. We calculate $\beta_i(t)$ as, 1.$\beta_i(T) = 1$,

2.$\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1) a_{ij} b_j(y_{t+1})$.

运行结果.

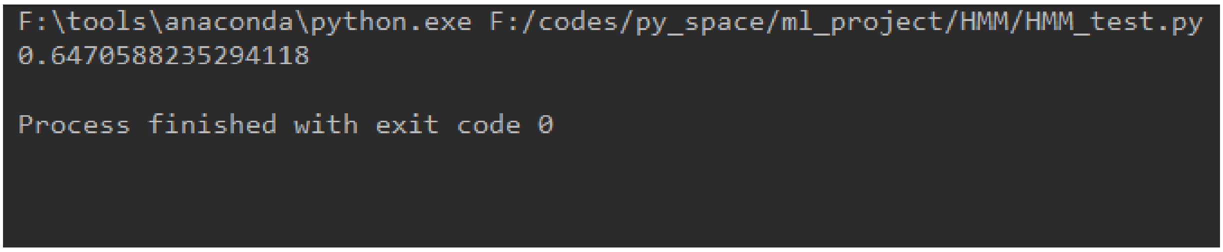

图 2: HMM