统计学习方法

第十章　隐马尔可夫模型

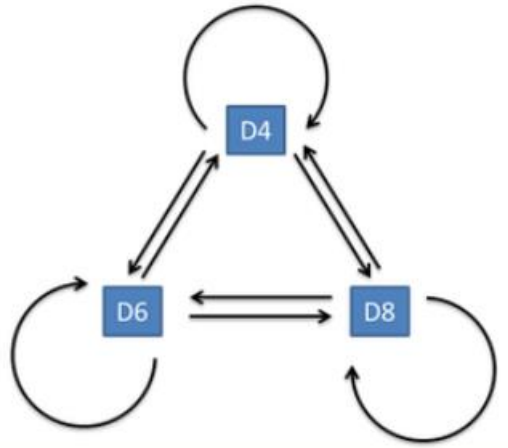
沈祺

2019.04.23

第十章 隐马尔可夫模型

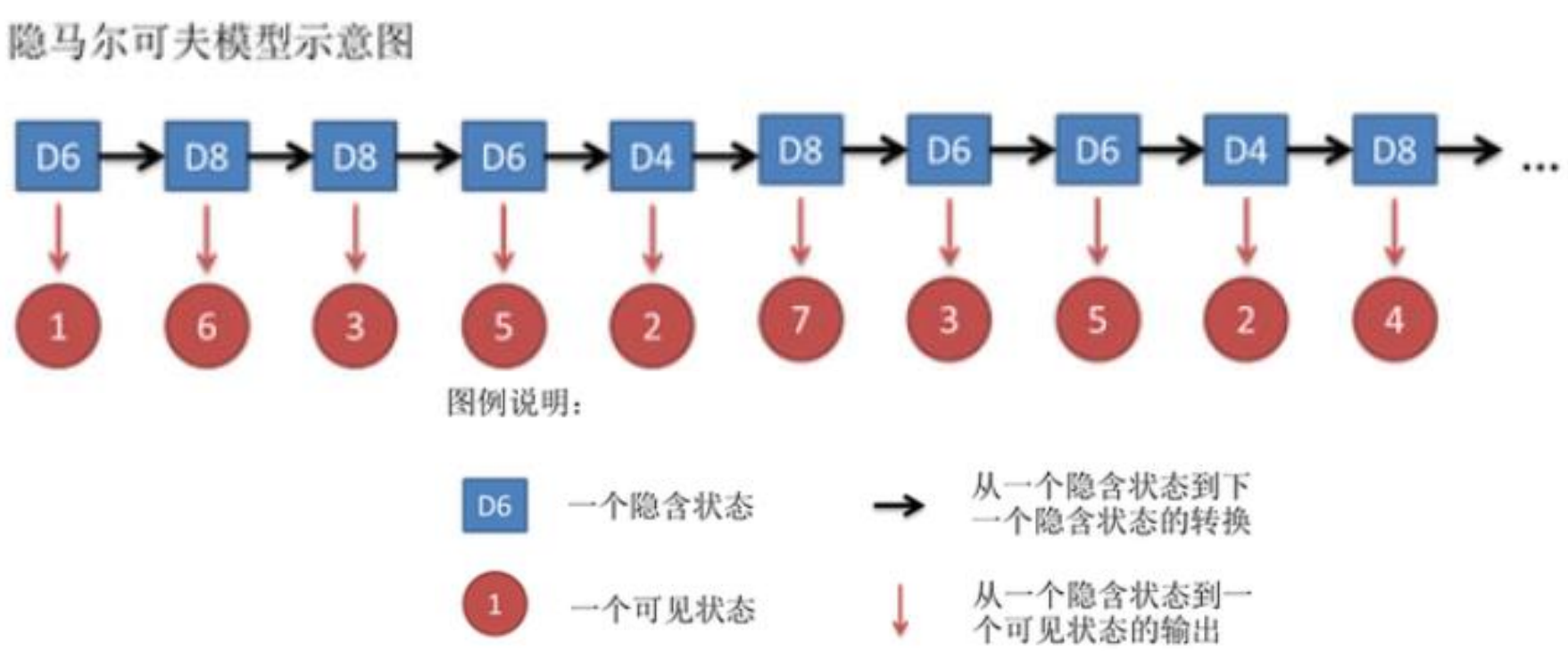
10.1.1 隐马尔可夫模型的定义

• **马尔可夫链：**为状态空间中经过从一个状态到另一个状态的转换的随机过程。该过程要求具备“无记忆”的性质，即下一状态的概率分布只能由当前状态决定，在时间序列中它前面的事件均与之无关。这种特定类型的“无记忆性”称作马尔可夫性质。



10.1.1 隐马尔可夫模型的定义(结构部分)

• 隐马尔可夫模型是关于时序的概率模型，描述由一个隐藏的马尔可夫链随机生成不可观测的随机状态序列**(隐状态序列)**，再由这个状态序列生成一个观测而产生观测随机序列**(观测序列)**的过程。



10.1.1 隐马尔可夫模型的定义(前提假设)

•**齐次马尔可夫性假设：**即假设隐藏的马尔可夫链在任意时刻t的状态只依赖于其前一时刻的状态，与其他时刻的状态及观测无关，也与时刻t无关。

$$P(i_t|i_{t-1},o_{t-1},\cdots,i_1,o_1)=P(i_t|i_{t-1}),\ t=1,2,\cdots,T$$

•**观测独立性：**即假设任意时刻的观测只依赖于该时刻的马尔可夫链的状态，与其他观测及状态无关。

$$P(o_t|i_T,o_T,i_{T-1},o_{T-1},\cdots,i_{t+1},o_{t+1},i_t,i_{t-1},o_{t-1},\cdots,i_1,o_1)=P(o_t|i_t),\ t=1,2,\cdots,T$$

i_t表示t时刻的隐状态，o_t表示t时刻的观测状态

10.1.1 隐马尔可夫模型的定义(参数部分)

- Q(query)是所有可能的隐状态的集合$Q = \{q_1, q_2, \cdots, q_N\}$
- V(viewing)是所有可能的观测的集合$V = \{v_1, v_2, \cdots, v_M\}$

- I(implicit) 是长度为T的隐状态序列 $I = (i_1, i_2, \cdots, i_T)$
- O(observable)是对应的观测序列 $O = (o_1, o_2, \cdots, o_T)$

- A是隐状态转移概率矩阵:

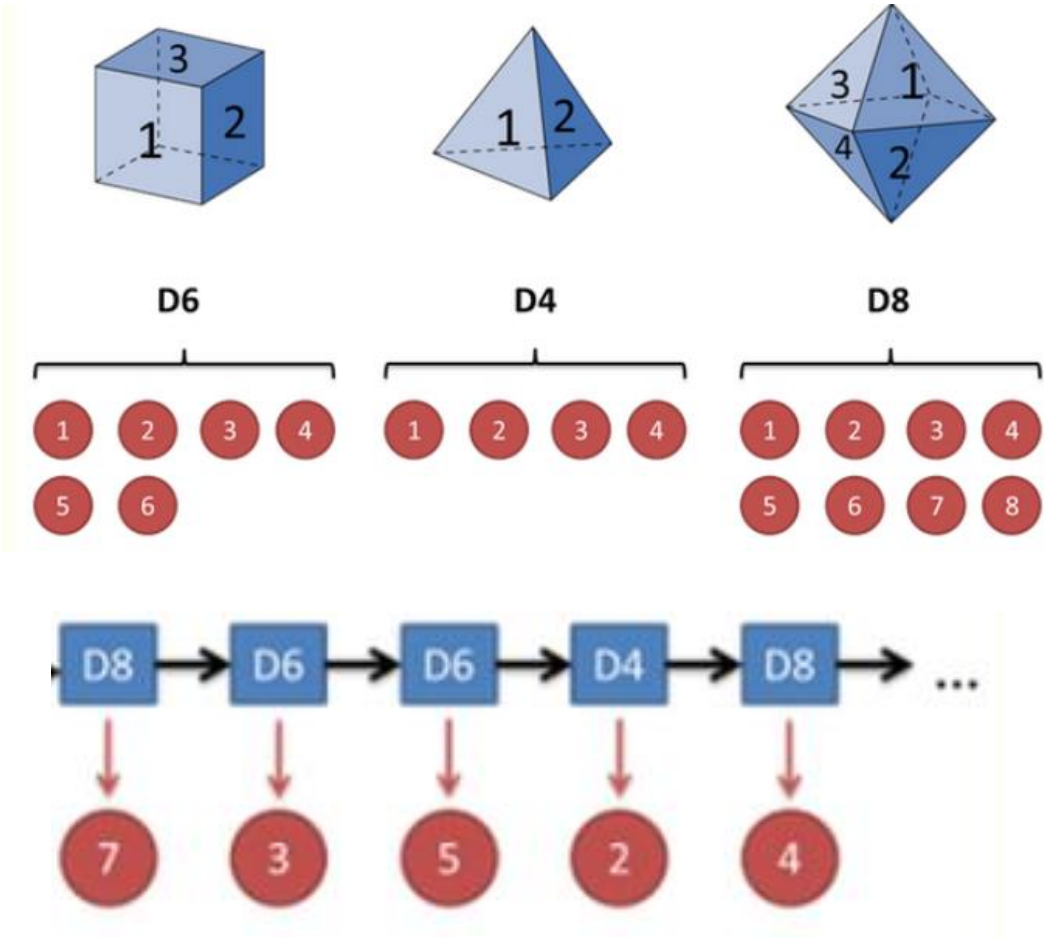
  $A = [a_{ij}]_{N\times N}$ $a_{ij}=P(i_{t+1} = q_j | i_t = q_i), i = 1,2,\cdots,N;\ j = 1,2,\cdots,N$

- B是观测概率矩阵:

  $B = [b_j(k)]_{N\times M}$ $b_j(k) = P(o_t = v_k | i_t = q_j), k = 1,2,\cdots,M;\ j = 1,2,\cdots,N$

- π是初始概率矩阵:$\pi_i = P(i_1 = q_i), i = 1,2,\cdots,N$

10.1.1 隐马尔可夫模型的定义(举例)



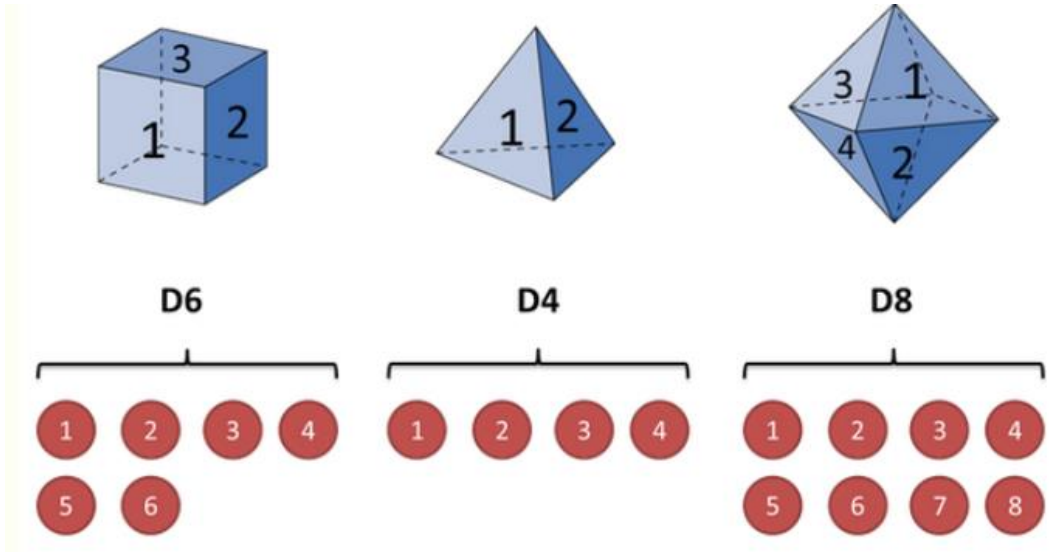
隐状态集合Q = {D6,D4,D8}

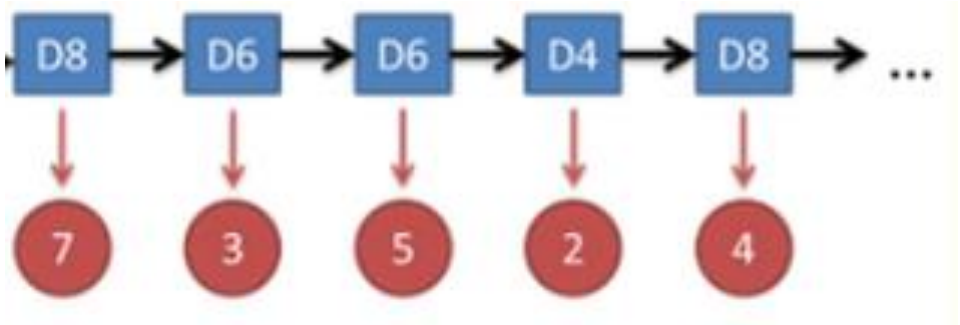
观测状态集合V = {1,2,3,4,5,6,7,8}

隐状态序列I = {D8,D6,D6,D4,D8}

对应观测序列O = {7,3,5,2,4}

10.1.1 隐马尔可夫模型的定义(举例)





	D6	D4	D8
D6	1/3	1/3	1/3
D4	1/3	1/3	1/3
D8	1/3	1/3	1/3

隐状态概率转移矩阵A

	1	2	3	4	5	6	7	8
D6	1/6	1/6	1/6	1/6	1/6	1/6	0	0
D4	1/4	1/4	1/4	1/4	0	0	0	0
D8	1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8

观测概率矩阵B

D6	D4	D8
1/3	1/3	1/3

初始状态矩阵π

10.1.2 观测序列的生成过程

- 输入：隐马尔可夫模型 λ=(A,B,π)
- 输出：观测序列O = ($o_1, o_2, \cdots, o_T$)

(1)按照初始状态分布π产生状态i_1

(2)令t=1

(3)按照状态i_t的观测概率分布$b_{i_t}(k)$生成o_t

(4)按照状态i_t的隐状态转移概率矩阵$[a_{ij}]_{N\times N}$选取最大概率转移产生状态i_{t+1}

(5)令t=t+1;如果t≤T，转(3);否则，终止

10.1.3 隐马尔可夫模型的三个基本问题

- 概率计算问题
- 学习问题
- 预测问题

问题1：概率计算问题

问题描述：给定模型λ=(A,B,π) 和观测序列O = (o_1, o_2,…, o_T),计算在模型λ下观测序列O出现的概率P(O|λ)

问题1：概率计算问题

10.2.1 直接计算法

- $P(O|\lambda) = \sum_I P(O|I,\lambda)P(I|\lambda)$
- $P(I|\lambda) = \pi_{i_1} a_{i_1 i_2} a_{i_2 i_3} \cdots a_{i_{T-1} i_T}$
- $P(O|I,\lambda) = b_{i_1}(o_1) b_{i_2}(o_2) \cdots b_{i_T}(o_T)$
- $P(O|\lambda) = \sum_{i_1, i_2, \cdots, i_T} \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} b_{i_2}(o_2) \cdots a_{i_{T-1} i_T} b_{i_T}(o_T)$

但是这种方式计算量很大，是$O(TN^T)$阶，这种算法不可行!

问题1：概率计算问题

10.2.2 前向算法(1)

- 前向概率：给定隐马尔可夫模型λ，定义到时刻t部分观测序列$o_1, o_2, \cdots, o_t$且状态为q_j的概率为前向概率，记作$\alpha_t(j) = P(o_1, o_2, \cdots, o_t, i_t = q_j|\lambda)$

递推过程：

(1)初值

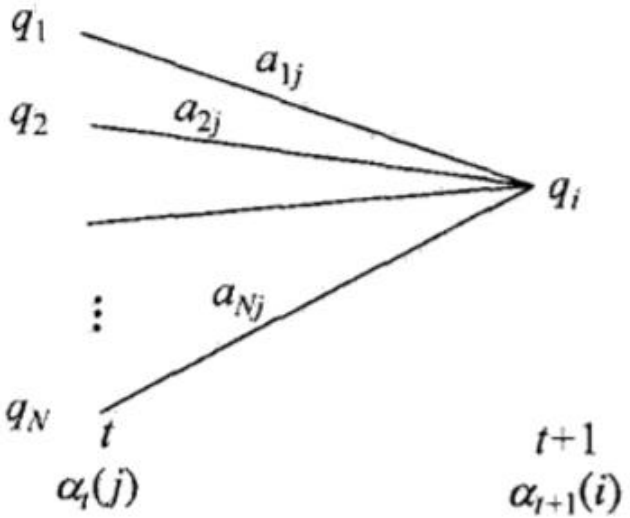
$$\alpha_1(i) = \pi_i b_i(o_1), \quad i = 1,2,\cdots,N$$

(2) 递推 求t=1,2,…,T-1,

$$\alpha_{t+1}(i) = \left[\sum_{j=0}^{N} \alpha_t(j) a_{ji}\right] b_i(o_{t+1}), \quad i = 1,2,\cdots,N$$

(3) 终止

$$P(O \mid \lambda) = \sum_{i=0}^{N} \alpha_T(i)$$



计算量为仅$O(N^2T)$

10.2.3 后向算法(1)

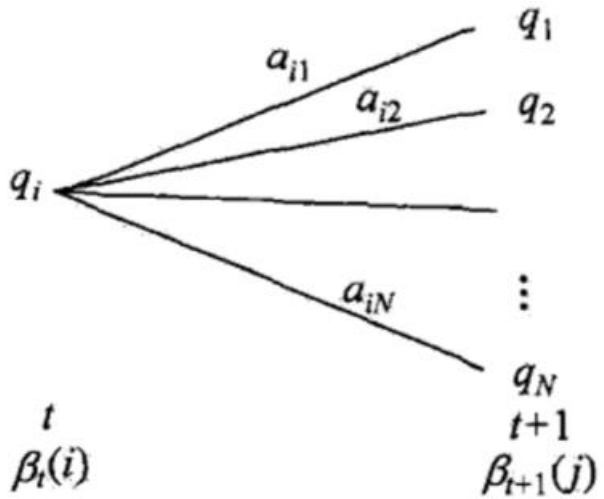
• 后向概率：给定隐马尔可夫模型λ，定义在时刻t状态为q_j的条件下，从t+1到T的部分观测序列为$o_{t+1}, o_{t+2}, \cdots, o_T$的概率为后向概率,记作

$$\beta_t(j) = P(o_{t+1}, o_{t+2}, \cdots, o_T | i_t = q_j, \lambda)$$

递推过程：

(1) 初值

$$\beta_T(i) = 1, \qquad i = 1,2,\cdots,N$$



(2) 对t=T-1,T-2,…, 1,

$$\beta_t(i) = \sum_{j=0}^{N} a_{ij}\, b_i(o_{t+1})\beta_{t+1}(j), \quad i = 1,2,\cdots,N$$

(3) 终止

$$P(O \mid \lambda) = \sum_{i=0}^{N} \pi_i b_i(o_1)\, \beta_1(i)$$

10.2.3 后向算法(2)

- 利用前向概率和后向概率的定义可以将观测序列概率P(O|λ) 统一写成

$$P(O|\lambda) = \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(t),\ t = 1,2,\cdots,T-1$$

其中t取[1,T)中任意一个值结果都是一样的

当t=1 时

$$P(O \mid \lambda) = \sum_{i=0}^{N}\pi_i b_i(o_1)\,\beta_1(i)$$

当t=T-1时

$$P(O \mid \lambda) = \sum_{i=0}^{N}\alpha_T(i)$$

10.2.4 一些概率和期望值的计算(1)

给定模型λ和观测O，在时刻t处于状态q_i的概率记

$$\gamma_t(i) = P(i_t = q_i|O,\lambda)$$

可以通过前向后向概率计算. 事实上，

$$\gamma_t(i) = P(i_t = q_i|O,\lambda) = \frac{P(i_t=q_i,O|\lambda)}{P(O|\lambda)}$$

由前向概率$\alpha_t(i) = P(o_1, o_2, \cdots, o_t, i_t = q_i|\lambda)$

后向概率$\beta_t(i) = P(o_{t+1}, o_{t+2}, \cdots, o_T|i_t = q_i, \lambda)$

可得$\alpha_t(i)\beta_t(i) = P(i_t = q_i, O|\lambda)$

于是想到：

$$\boldsymbol{\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}}$$

10.2.4 一些概率和期望值的计算(2)

给定模型λ和观测O，在时刻t处于转来q_i且在时刻t+1处于状态q_j的概率，记

$$\xi_t(i,j)=P(i_t=q_i,i_{t+1}=q_j|O,\lambda)$$

可以通过前向后向概率计算：

$$\xi_t(i,j)=\frac{P(i_t=q_t,i_{t+1}=q_j,O|\lambda)}{P(O|\lambda)}=\frac{P(i_t=q_t,i_{t+1}=q_j,O|\lambda)}{\sum_{i=1}^{N}\sum_{j=1}^{N}P(i_t=q_t,i_{t+1}=q_j,O|\lambda)}$$

$$P(i_t=q_t,i_{t+1}=q_j,O|\lambda)=\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)$$

$$\boldsymbol{\xi_t(i,j)=\frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}}$$

10.2.4 一些概率和期望值的计算(3)

- 将$\gamma_t(i)$和$\xi_t(i,j)$对各个时刻t求和，可以得到一些有用的期望值
  - 在观测O下状态i出现的期望值
    $\sum_{t=1}^{T}\gamma_t(i)$
  - 在观测O下由状态i转移的期望值
    $\sum_{t=1}^{T-1}\gamma_t(i)$
  - 在观测O下由状态i转移到状态j的期望值
    $\sum_{t=1}^{T-1}\xi_t(i,j)$

问题2：学习问题

问题描述：已知观测序列$O = (o_1, o_2, \cdots, o_T)$，估计模型$\lambda=(A,B,\pi)$的参数，使得在该模型下观测序列概率$P(O|\lambda)$最大.即用极大似然估计的方法估计参数

问题2：学习问题

10.3.1 监督学习方法

1、隐状态转移概率的估计

设样本中时刻t处于状态i，时刻t+1转移到状态j的频数为A_{ij}，那么隐状态转移概率a_{ij}的估计是

$$a_{ij} = \frac{A_{ij}}{\sum_{j=1}^{N} A_{ij}}$$

2、观测概率$b_j(k)$的估计

设样本中状态为j并观测为k的频数是B_{jK}，那么状态为j观测为k的概率$b_j(k)$的估计是

$$b_j(k) = \frac{B_{jK}}{\sum_{k=1}^{M} B_{jk}}$$

3、初始状态概率π_i的估计$\hat{\pi}_i$为S个样本中初始状态为q_i的频率

问题2：学习问题

10.3.2 Baum-Welch算法（EM算法）（1）

1.确定参数

(1) 观测状态$O = (o_1, o_2, \cdots, o_T)$

(2) 隐状态$I = (i_1, i_2, \cdots, i_T)$

(3) 完全数据$(O,I)=(o_1, o_2, \cdots, o_T, i_1, i_2, \cdots, i_T)$

(4) 完全数据的对数似然函数是$\log P(O,I|\lambda)$

2.EM算法的E步： 求Q函数$Q(\lambda,\bar{\lambda})$,其中，$\bar{\lambda}$是隐马尔可夫模型参数的当前估计值，λ是要极大化的隐马可夫模型参数

$$Q(\lambda,\bar{\lambda}) = \sum_I [\log P(O,I|\lambda)]P(I|O,\bar{\lambda})$$

$$P(I|O,\bar{\lambda})=\frac{P(O,I|\bar{\lambda})}{P(O,\bar{\lambda})}$$

对于固定的模型$P(O,\bar{\lambda})$ 是常数,所以：

$$Q(\lambda,\bar{\lambda}) = \sum_I [\log P(O,I|\lambda)]P(O,I|\bar{\lambda})$$

因为$P(O,I|\lambda) = \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} b_2(o_2) a_{i_2 i_3} \cdots a_{i_{T-1} i_T} b_{i_T}(o_T)$

于是函数$Q(\lambda,\bar{\lambda})$ 可以写成

$$Q(\lambda,\bar{\lambda}) = \sum_I \log \pi_{i_1} P\left(O, I\middle|\bar{\lambda}\right) + \sum_I\left(\sum_{t=1}^{T-1} \log a_{i_t i_{t+1}}\right)P(O,I|\bar{\lambda}) + \sum_I\left(\sum_{t=1}^{T} \log b_{i_t}(o_t)\right)P(O,I|\bar{\lambda}) \quad (10.34)$$

问题2：学习问题

10.3.2 Baum-Welch算法（EM算法）（2）

3.EM算法的M步： 极大化Q函数$Q(\lambda,\bar{\lambda})$求模型A,B,π，由于要极大化的参数式子在(10.34)中独立地出现在三个项中，所以只需对各项分别极大化.

(1) 式(10.34)的第一项可以写成：

$$\sum_I[\log \pi_{i_1}]P(O,I|\bar{\lambda}) = \sum_{i=1}^{N}[\log \pi_i]\, P(O,i_1 = i|\bar{\lambda})$$

注意$\sum_{i=1}^{n}\pi_i = 1$,利用拉格朗日乘子法，写出拉格朗日函数

$$\sum_{i=1}^{N}[\log \pi_i]\, P(O,i_1 = i|\bar{\lambda}) + \gamma(\sum_{i=1}^{n}\pi_i - 1)$$

对其结果求偏导并令结果为0

$$\frac{\partial}{\partial \pi_i}[\sum_{i=1}^{N}[\log \pi_i]\, P(O,i_1 = i|\bar{\lambda}) + \gamma(\sum_{i=1}^{N}\pi_i - 1)] = 0$$

得

$$P(O,i_1 = i|\bar{\lambda}) + \gamma\pi_i = 0 \qquad (10.35)$$

对 I 求和得到γ

$$\gamma = -P(O|\bar{\lambda})$$

带入式(10.35)即得

$$\pi_i = \frac{P(O,i_1=i|\bar{\lambda})}{P(O|\bar{\lambda})}$$

问题2：学习问题

10.3.2 Baum-Welch算法（EM算法）（2）

(2) 式(10.34)的第2项：

$$\sum_I\left(\sum_{t=1}^{T-1}\log a_{i_t i_{t+1}}\right)P(\mathrm{O},\mathrm{I}|\bar{\lambda})=\sum_{\mathrm{i}=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T-1}[\log a_{ij}]P(\mathrm{O},i_t=i,i_{t+1}=j|\bar{\lambda})$$

类似第1项，应用具有约束条件$\sum_{\mathrm{j}=1}^{N}a_{ij}=1$的拉格朗日乘子法可以求出

$$\mathrm{a}_{ij}=\frac{\sum_{t=1}^{T-1}P(O,i_t=i,i_{t+1}=j|\bar{\lambda})}{\sum_{t=1}^{T-1}P(O,i_t=i|\bar{\lambda})}$$

(3)式(10.34)的第3项：

$$\sum_I\left(\sum_{t=1}^{T}\log b_{i_t}(o_t)\right)P(\mathrm{O},\mathrm{I}|\bar{\lambda})=\sum_{j=1}^{N}\sum_{t=1}^{T}[\log b_j(o_t)]P(\mathrm{O},i_t=j|\bar{\lambda})$$

同样用拉格朗日乘子法，约束条件是$\sum_{\mathrm{k}=1}^{M}b_j(k)=1$. 只有在$\mathrm{o}_t=v_k$时$\mathrm{b}_j(o_t)$对$\mathrm{b}_j(\mathrm{k})$的偏导数才不为0，以$\mathrm{I}(\mathrm{o}_t=v_k)$表示. 求得

$$b_j(k)=\frac{\sum_{t=1}^{T}P(O,i_t=j|\bar{\lambda})\mathrm{I}(\mathrm{o}_t=v_k)}{\sum_{t=1}^{T}P(O,i_t=j|\bar{\lambda})}$$

10.3.3 Baum-Welch模型参数估计公式

将上式中的各概率分别用$\gamma_t(i),\xi_t(i,j)$表示，则可将相应的公式写成：

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

$$b_j(k)=\frac{\sum_{t=1,o_t=v_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$$

$$\pi_i=\gamma_1(i)$$

$$\boldsymbol{\xi_t(i,j)=P(i_t=q_i,i_{t+1}=q_j|O,\lambda)}$$

$$\boldsymbol{\gamma_t(i)=P(i_t=q_i|O,\lambda)}$$

问题2：学习问题

10.3.3 Baum-Welch模型参数估计过程

输入：观测数据O = $(o_1, o_2, \cdots, o_T)$

输出：隐马尔可夫模型

(1)初始化

对n=0, 选取$a_{ij}^{(0)}, b_j(k)^{(0)}, \pi_i^{(0)}$, 得到模型$\lambda^{(0)} = (A^{(0)}, B^{(0)}, \pi^{(0)})$.

(2)递推. 对n = 0,1,2,$\cdots$,

$$a_{ij}^{(\mathrm{n}+1)} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$b_j(k)^{n+1} = \frac{\sum_{t=1, o_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

$$\pi_i^{(n+1)} = \gamma_1(i)$$

右端各值按观测O = $(o_1, o_2, \cdots, o_T)$和模型$(A^{(n)}, B^{(n)}, \pi^{(n)})$计算。

(3) 终止，得到模型$\lambda^{(\mathrm{n}+1)} = (A^{(n+1)}, B^{(n+1)}, \pi^{(n+1)})$

问题3：预测问题

问题描述：也称为解码问题.已知模型λ=(A,B,π) 和观测序列O = ($o_1, o_2, \cdots, o_T$),求对给定观测序列条件概率P(I|O)最大的隐状态序列I=($i_1, i_2, \cdots, i_T$)。即给定观测序列，求最有可能的对应的状态序列

问题3：预测问题

10.4.1 近似算法

近似算法的思想是，在每个时刻t选择在该时刻最有可能出现的状态i_t^*,从而得到一个状态序列$I^* = (i_1^*, i_2^*, \cdots, i_T^*)$，将它作为预测的结果。

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}$$

$$\alpha_t(j) = P(o_1, o_2, \cdots, o_t, i_t = q_j|\lambda)$$

$$\beta_t(j) = P(o_{t+1}, o_{t+2}, \cdots, o_T|i_t = q_j, \lambda)$$

在每一时刻t最有可能的状态i_t^*是

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

从而得到状态序列$I^* = (i_1^*, i_2^*, \cdots, i_T^*)$

优点:计算简单

缺点:不能保证预测的状态序列整体是最有可能的状态序列，因为预测的状态序列可能有实际不发生的部分，比如会出现$a_{ij} = 0$但是仍预测出状态q_i到q_j。

问题3：预测问题

10.4.2 维比特算法（1）

- 维比特算法实际是用动态规划解隐马尔可夫模型的预测问题

- 最优路径有这样的特性：如果最优路径在时刻t通过结点i_t^*,那么这一路径从结点i_t^*到终点i_T^*的部分路径，对于从i_t^*到i_T^*的所有可能的部分路径来说,必须是最优的。

- 依据这一原理，我们只需要：

(1)时刻t=1开始，递推地计算在时刻t的状态为i的各条部分路径的最大概率,时刻t=T的最大概率即为最优路径的概率P^*，最优路径的终结点i_T^*也同时得到.

(2)从终结点i_T^*开始,由后向前逐步求得结点$i_{T-1}^*, \cdots, i_1^*$,得到最优路径$I^* = (i_1^*, i_2^*, \cdots, i_T^*)$.这就是维比特算法.

问题3：预测问题

10.4.2 维比特算法（2）

- 导入两个变量δ和ψ
- 定义在时刻t状态为i的所有单个路径$(i_1, i_2, \cdots, i_t)$中概率最大值为

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

- 由定义可得变量δ的递推公式

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

$$= \max_{1 \le j \le N} [\delta_t(j) a_{ji}] b_i(o_{t+1}) \ , i = 1,2, \cdots, n; t = 1,2, \cdots, T-1$$

- 定义在时刻t状态为i的所有单个路径$(i_1, i_2, \cdots, i_{t-1}, i)$中概率最大的路径的第t-1个节点为

$$\psi_t(i) = \arg \max_{1 \le j \le N} [\delta_{t-1}(j) a_{ji}] \ , i = 1,2, \cdots, N$$

注：由于有可能有多条最优路径 所以$\psi_t(i), \delta_t(i)$应该是list

问题3：预测问题

10.4.2 维比特算法（3）

输入: 模型λ=(A,B,π)和观测$O=(o_1, o_2, \cdots, o_T)$

输出： 最优路径$I^* = (\mathrm{i}_1^*, \mathrm{i}_2^*, \cdots, \mathrm{i}_T^*)$

(1)初始化

$$\delta_1(i)=\pi_i b_i(o_1),\ i = 1,2,\cdots,N$$

$$\psi_t(i) = 0,\ i = 1,2,\cdots,N$$

(2)递推。 对t=2, 3, $\cdots, T$

$$\delta_t(i) = \max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right]b_i(o_t)\,, i = 1,2,\cdots,N$$

$$\psi_t(i) = arg\ \max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right]\ \ ,i = 1,2,\cdots,N$$

(3)终止

$$P^* = \max_{1\le i\le N}\delta_T(\mathrm{i})$$

$$i_T^* = arg\ \max_{1\le i\le N}[\delta_T(i)]$$

(4)最优路径回溯。对t=$\mathrm{T}-1, \mathrm{T}-2, \cdots, 1$

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

求得最优路径$I^* = (\mathrm{i}_1^*, \mathrm{i}_2^*, \cdots, \mathrm{i}_T^*)$

例10.3 假设有三个盒子D1 D2 D3，每个盒子中都有一些红球和白球

已知

	D1	D2	D3
D1	0.5	0.2	0.3
D2	0.3	0.5	0.2
D3	0.2	0.3	0.6

隐状态概率转移矩阵A

	红	白
D1	0.5	0.5
D2	0.4	0.6
D3	0.7	0.3

观测概率转移矩阵B

D1	D2	D3
0.2	0.4	0.4

初始概率矩阵π

已知观测序列O = (红，白，红)， 试求最优状态序列，即最优路径$I^* = (i_1^*, i_2^*, \cdots, i_T^*)$

(1) 初始$\delta_1(i)$=$\pi_i b_i(o_1)$

$\delta_1(i)$=$\pi_i b_i$(红)$, i = 1,2,3$

带入实际数据

$\delta_1(D1)$=0. 10　$\delta_1(D2)$=0. 16　$\delta_1(D3)$=0. 28

$\psi_1(i) = 0,\ i = 1,2,\cdots,N$

(2)递推

$$\delta_t(i) = \max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right]b_i(o_t)\,, i = 1,2,\cdots,N$$

$$\delta_2(D1) = \max_{1\le j\le 3}\left[\delta_1(j)a_{j1}\right]b_1(o_2)$$
$$= \max_j(0.10\times 0.5, 0.16\times 0.3, 0.28\times 0.2)\times 0.5$$
$$= 0.028$$

$$\psi_t(i) = arg\max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right]\ \ ,i = 1,2,\cdots,N$$

$\psi_2(D1) = D3$
$\delta_2(D2) = 0.0504\ \ \psi_2(D2) = D3$
$\delta_2(D3) = 0.042\ \ \psi_2(D3) = D3$
$\delta_3(D1) = 0.00756\ \ \psi_3(D1) = D2$
$\delta_3(D2) = 0.01008\ \ \psi_3(D2) = D2$
$\delta_3(D3) = 0.0147\ \ \ \psi_3(D3) = D3$

	D1	D2	D3
D1	0.5	0.2	0.3
D2	0.3	0.5	0.2
D3	0.2	0.3	0.6

隐状态概率转移矩阵A

	红	白
D1	0.5	0.5
D2	0.4	0.6
D3	0.7	0.3

观测概率转移矩阵B

D1	D2	D3
0.2	0.4	0.4

初始概率矩阵π

O = (红，白，红)

(3) 以P^*表示最优路径的概率

$$P^* = \max_{1\le i\le 3} \delta_3(i) = 0.0147 = \delta_3(D3)$$

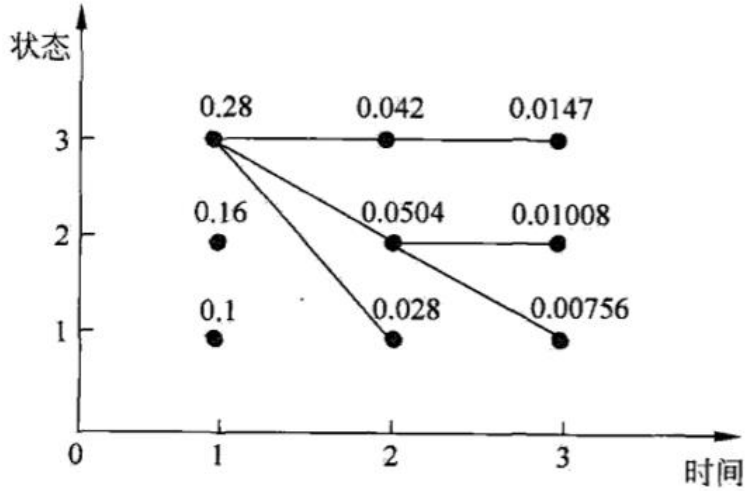
最优路径$i_3^* = arg \max_{1\le i\le 3}[\delta_3(i)] = D3$

(4) 由最优路径的终点 i_3^*，逆向找到i_2^*，i_1^*：

t = 2 时 ， $i_2^* = \psi_3(i_3^*) = \psi_3(D3) = D3$

t = 1 时 ， $i_1^* = \psi_2(i_2^*) = \psi_2(D3) = D3$

于是求得最优状态序列$I^* = (i_1^*, i_2^*, \cdots, i_T^*) = (D3, D3, D3)$



	D1	D2	D3
D1	0.5	0.2	0.3
D2	0.3	0.5	0.2
D3	0.2	0.3	0.6

隐状态概率转移矩阵A

	红	白
D1	0.5	0.5
D2	0.4	0.6
D3	0.7	0.3

观测概率转移矩阵B

D1	D2	D3
0.2	0.4	0.4

初始概率矩阵π

O = (红，白，红)

隐马尔可夫模型的应用：

- 词性标注
- 语音识别
- 金融领域
- 网络安全

.

.

.

经典隐马尔可夫模型的缺点：

- 模型仍过于简单，当前状态只与前一状态相关。(实际应该与序列长度和上下文等等都有关系)
- 观测值之间是相互独立的
- 目标函数和预测目标函数不匹配。
  HMM学到的是隐状态和观测序列的联合分布P(I,O),而预测问题中,我们需要的是条件概率P(I|O)

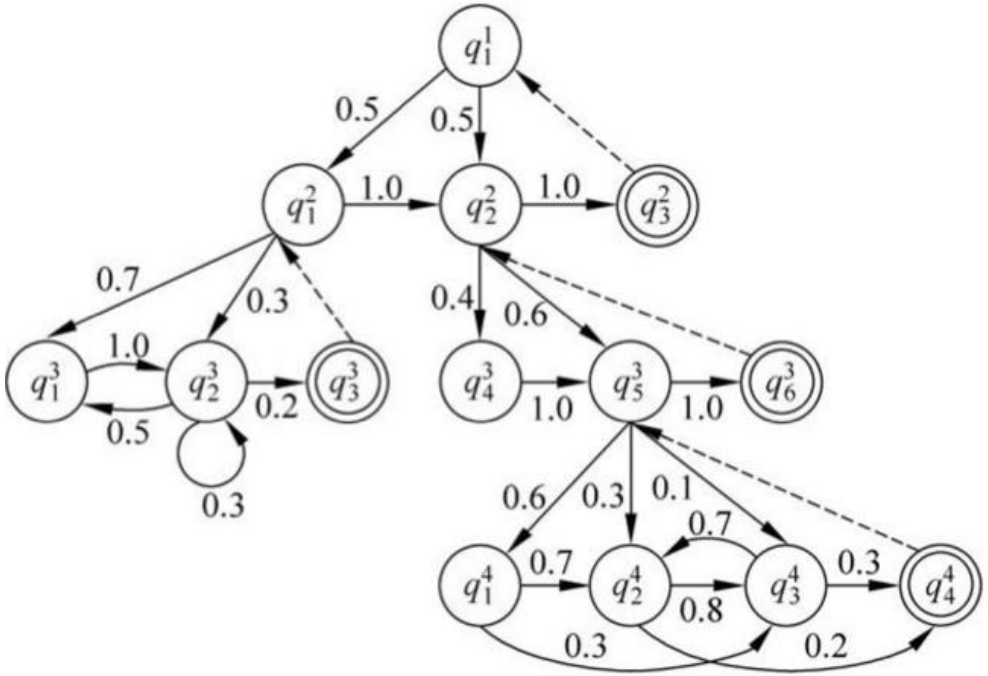
经典隐马尔可夫模型的一些变种：

- 和前一状态的观测值也相关
- 和前几个状态相关
- 观测状态是连续的
- 隐状态有多种

.

.

.



4 无限状态隐马尔可夫模型

虽然 HMM 是一种常用的概率统计模型，但在实际应用中，却受到很大的限制，主要体现在：(1)HMM 学习使用的经典 EM 算法中 M 步骤估计时没有考虑到模型的复杂度，不能解决模型的过适应或欠适应问题；(2)在使用前，必须确定 HMM 的结构，即模型中的状态数目，然而，由于实际数据的复杂性以及数据的动态更新性，人为地指定状态数目通常不能最佳地描述数据，这样，模型只能从给定的数据集中得到有限的信息。针对这些问题，Beal 等人[17]于 2002 年提出了

层次化隐马尔可夫(HHMM):

1.一个HHMM的状态就产生一个观察序列，而不是一个观察符号。

2.特殊的终止状态负责控制转移过程返回到激活该层状态转移的上层状态。

隐马尔可夫模型的下溢问题：

将alpha归一化

```
def forward_with_scale(self):
    T = len(self.O)
    alpha_raw = np.zeros((T, self.N), np.float)
    alpha = np.zeros((T, self.N), np.float)
    c = [i for i in range(T)]  # scaling factor; 0 or sequence doesn't matter

    for i in range(self.N):
        alpha_raw[0,i] = self.Pi[i] * self.B[i, self.O[0]]

    c[0] = 1.0 / sum(alpha_raw[0,i] for i in range(self.N))
    for i in range(self.N):
        alpha[0, i] = c[0] * alpha_raw[0,i]

    for t in range(T-1):
        for i in range(self.N):
            summation = 0.0
            for j in range(self.N):
                summation += alpha[t,j] * self.A[j, i]
            alpha_raw[t+1, i] = summation * self.B[i, self.O[t+1]]

        c[t+1] = 1.0 / sum(alpha_raw[t+1,i1] for i1 in range(self.N))

        for i in range(self.N):
            alpha[t+1, i] = c[t+1] * alpha_raw[t+1, i]
    return alpha, c
```

递推过程：

(1)初值

$$\alpha_1(i) = \pi_i b_i(o_1), \quad i = 1,2,\cdots,N$$

(2) 递推 求t=1,2,…,T-1,

$$\alpha_{t+1}(i) = \left[\sum_{j=0}^{N} \alpha_t(j)a_{ji}\right] b_i(o_{t+1}), \quad i = 1,2,\cdots,N$$

(3) 终止

$$P(O \mid \lambda) = \sum_{i=0}^{N} \alpha_T(i)$$

感谢聆听
欢迎指教

$t = 1$ 时

$$\alpha_1(i) = \pi_i b_i(o_1) \qquad \beta_t(i) = \sum_{j=0}^{N} a_{ij}\, b_i(o_{t+1})\beta_{t+1}(j)$$

$$P(O|\lambda) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_1(i) a_{ij} b_j(o_2)\beta_2(j)$$

$$= \sum_{i=1}^{N} \alpha_1(i) \sum_{j=1}^{N} a_{ij} b_j(o_2)\beta_2(j)$$

$$= \sum_{i=1}^{N} \alpha_1(i)\beta_1(i)$$

$$= \sum_{i=0}^{N} \pi_i b_i(o_1)\, \beta_1(i)$$

$t = T\text{-}1$ 时

$$\beta_T(i) = 1 \quad \alpha_{t+1}(i) = \left[\sum_{j=0}^{N} \alpha_t(j) a_{ji}\right] b_i(o_{t+1})$$

$$P(O|\lambda) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{T-1}(i) a_{ij} b_j(o_T)\beta_T(j)$$

$$= \sum_{j=1}^{N}\sum_{i=1}^{N} \alpha_{T-1}(i) a_{ij} b_j(o_T)$$

$$= \sum_{j=1}^{N} \alpha_T(j)$$

$$\mathrm{L} = \sum_{j=1}^{N} \sum_{t=1}^{T} [\log b_j(o_t)] P(\mathrm{O}, i_t = j|\bar{\lambda}) + \lambda(\sum_{\mathrm{k}=1}^{M} b_j(k) - 1)$$

$$\frac{\partial L}{\partial \mathrm{b}_j(\mathrm{k})} = 0$$

得

$$\sum_{t=1}^{T} P\left(\mathrm{O}, i_t = j|\bar{\lambda}\right) I(\mathrm{o}_t = v_k) + \lambda \mathrm{b}_j(\mathrm{k}) = 0 \quad (1)$$

对k 求和得

$$\lambda = -\sum_{t=1}^{T} P\left(\mathrm{O}, i_t = j|\bar{\lambda}\right)$$

带回(1)式得

$$b_j(k) = \frac{\sum_{t=1}^{T} P(O, i_t = j|\bar{\lambda}) \mathrm{I}(\mathrm{o}_t = v_k)}{\sum_{t=1}^{T} P(O, i_t = j|\bar{\lambda})}$$