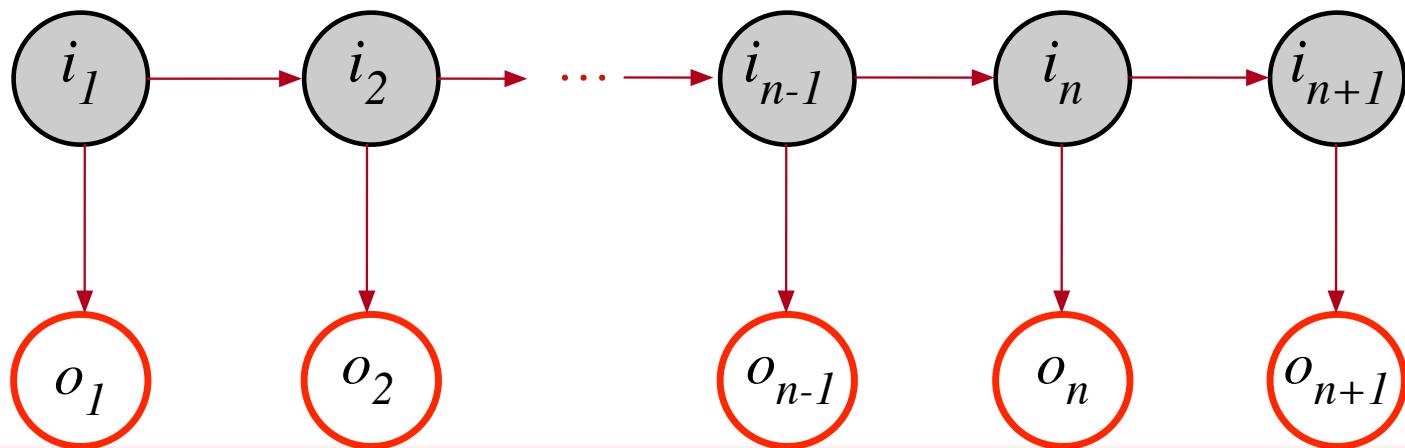


## 七、 HMM

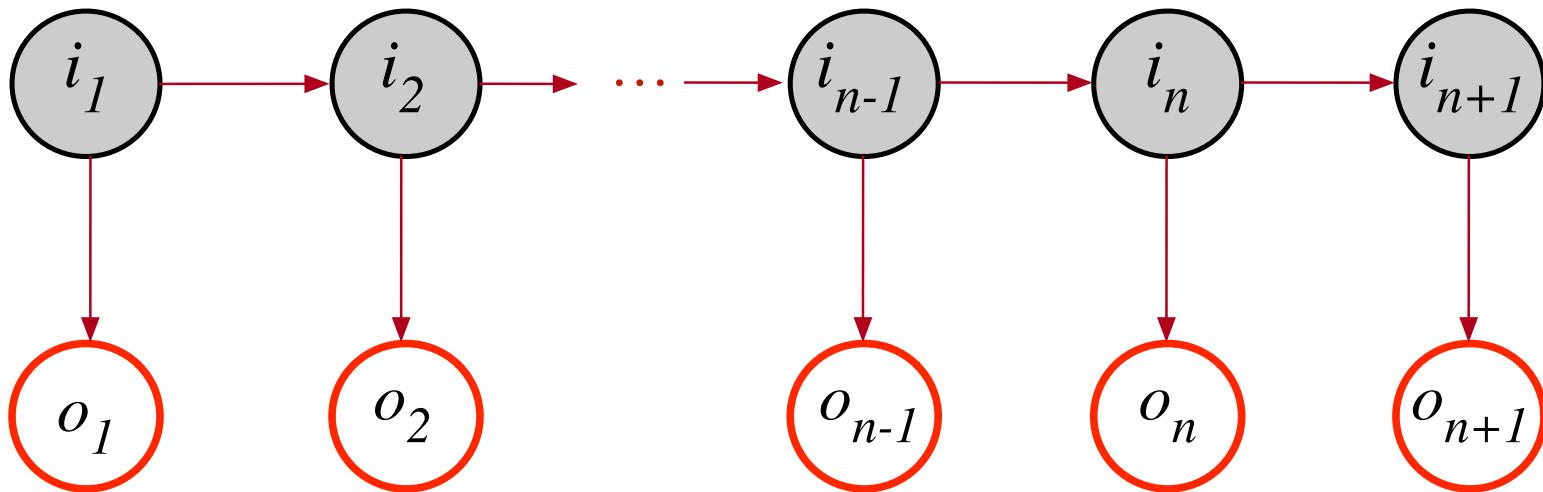
应用领域：语音识别，NLP，生物信息、模式识别，时序数据处理

# 隐马尔科夫模型的定义

- 隐马尔科夫模型(HMM, Hidden Markov Model)是关于时序的概率模型，描述由一个隐藏的马尔科夫链随机生成不可观测的状态随机序列，再由各个状态生成一个观测而产生观测随机序列的过程。
- 隐马尔科夫模型随机生成的状态的序列，称为**状态序列**；每个状态生成一个观测，由此产生的观测随机序列，称为**观测序列**。
  - 序列的每个位置可看做是一个时刻。



# 隐马尔科夫模型的贝叶斯网络



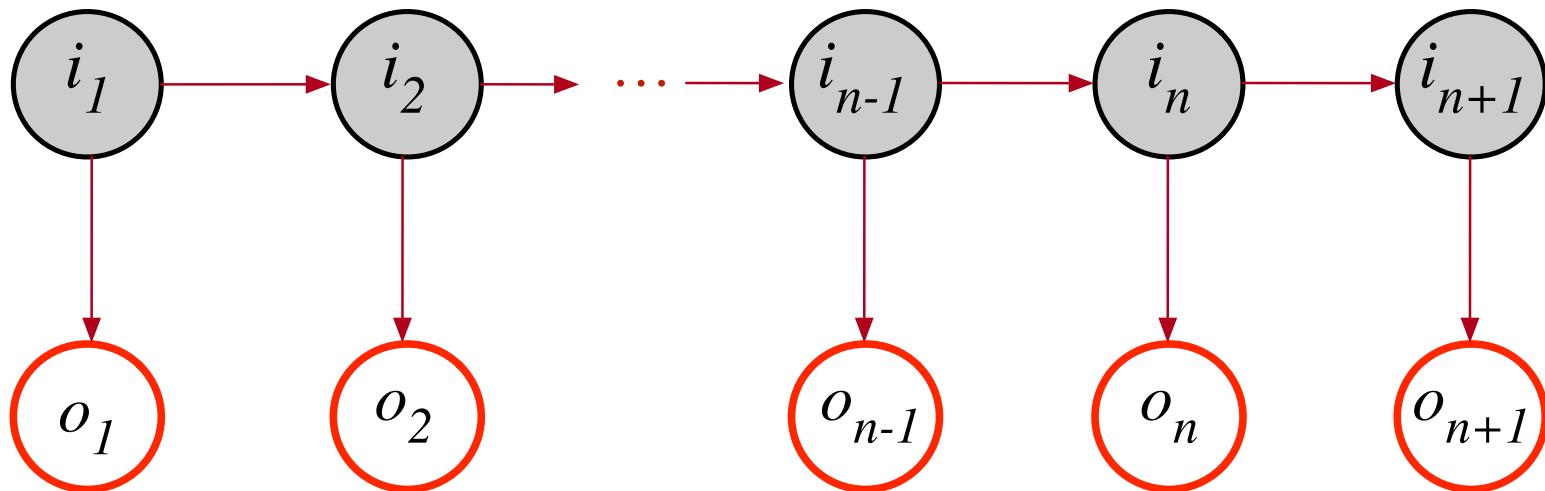
□ 请思考：

- 在  $i_1$  未给定的前提下， $o_1$  和  $i_2$  独立吗？ $o_1$  和  $o_2$  独立吗？

# HMM的确定

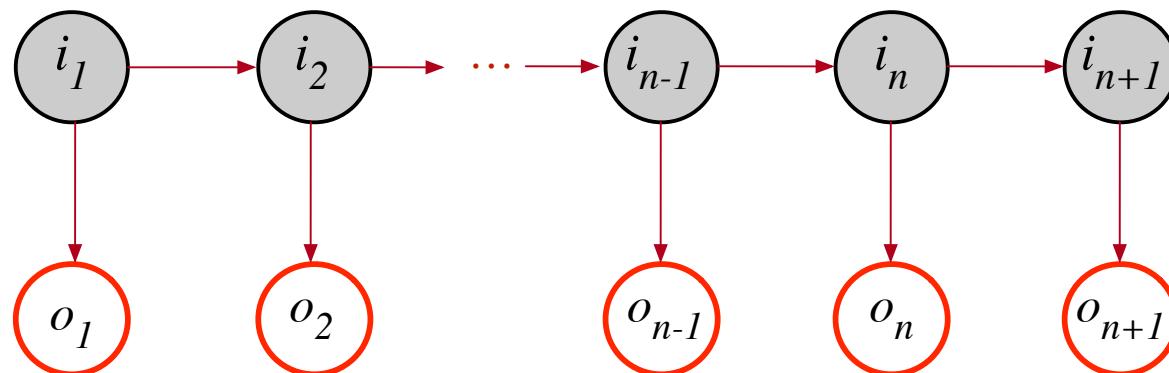
□ HMM由初始概率分布  $\pi$ 、状态转移概率分布  $A$  以及观测概率分布  $B$  确定。

$$\lambda = (A, B, \pi)$$

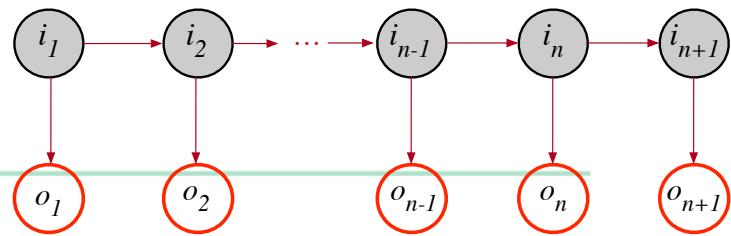


# HMM的参数

- $Q$  是所有可能的状态的集合
  - $N$  是可能的状态数  $Q = \{q_1, q_2, \dots, q_N\}$
- $V$  是所有可能的观测的集合
  - $M$  是可能的观测数  $V = \{v_1, v_2, \dots, v_M\}$



# HMM的参数



□ I 是长度为 T 的状态序列， O 是对应的观测序列

$$I = \{i_1, i_2, \dots, i_T\} \quad O = \{o_1, o_2, \dots, o_T\}$$

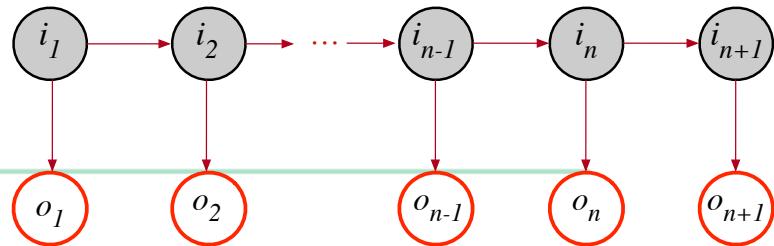
□ A 是状态转移概率矩阵

$$A = [a_{ij}]_{N \times N}$$

□ 其中  $a_{ij} = P(i_{t+1} = q_j | i_t = q_i)$

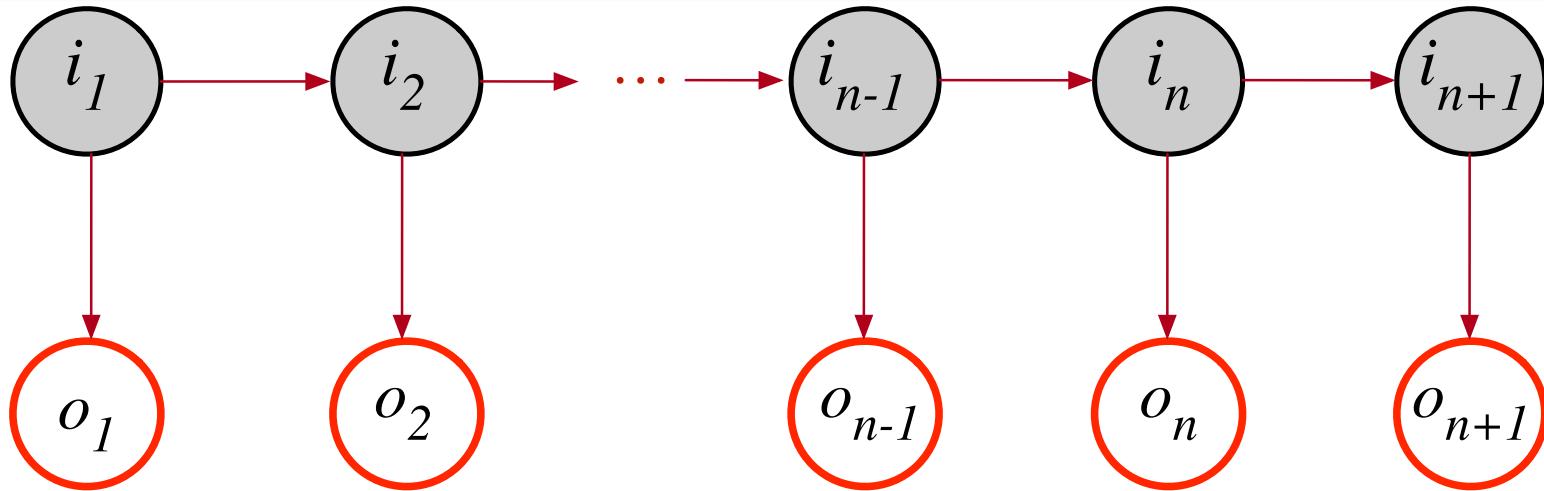
□  $a_{ij}$  是在时刻 t 处于状态  $q_i$  的条件下时刻 t+1 转移到状态  $q_j$  的概率。

# HMM的参数



- B是观测概率矩阵  $B = [b_{ik}]_{N \times M}$
- 其中,  $b_{ik} = P(o_t = v_k | i_t = q_i)$ 
  - $b_{ik}$  是在时刻 t 处于状态  $q_i$  的条件下生成观测  $v_k$  的概率。
- $\pi$  是初始状态概率向量 :  $\pi = (\pi_i)$
- 其中,  $\pi_i = P(i_1 = q_i)$ 
  - $\pi_i$  是时刻  $t=1$  处于状态  $q_i$  的概率。

# HMM的两个基本性质



□ 齐次假设 (状态) :

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

□ 观测独立性假设 (观测) :

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

# HMM举例

□ 假设有4个盒子， 编号为1、2、3、4， 每个盒子都装有红白两种颜色的小球， 数目如下：

盒子	1	2	3	4
红球数	5	3	6	8
白球数	5	7	4	2

□ 状态集合 :  $Q = \{\text{盒1}, \text{盒2}, \text{盒3}, \text{盒4}\}$ ,  $N = 4$

□ 观测集合 :  $V = \{\text{红}, \text{白}\}$

□ 观测概率矩阵  $B$  为 :

$$b_{ik} = P(o_t = v_k | i_t = q_i)$$

$$B = \begin{bmatrix} 0.5 & 0.5 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ 0.8 & 0.2 \end{bmatrix}$$

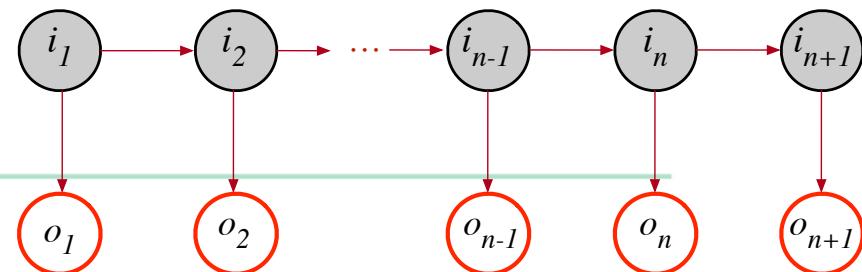
# HMM举例

---

盒子	1	2	3	4
红球数	5	3	6	8
白球数	5	7	4	2

- 按照下面的方法抽取小球， 得到球颜色的观测序列：
  - 按照  $\pi = (0.25, 0.25, 0.25, 0.25)^T$  的概率选择1个盒子， 从盒子随机抽出1个球， 记录颜色后放回盒子；

# HMM举例



盒子	1	2	3	4	$o_1$
红球数	5	3	6	8	$o_2$
白球数	5	7	4	2	$o_{n-1}$

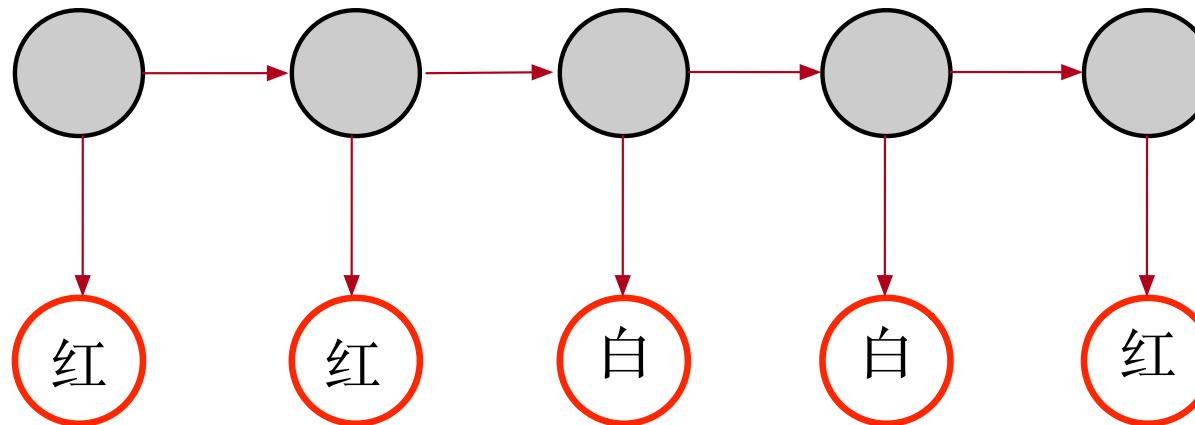
- 然后从当前盒子随机转移到下一个盒子，规则是：
  - ✓ 如果当前盒子是 1， 那么下一个盒子一定是2，
  - ✓ 如果当前盒子是2或3， 分别以概率 0.4 和 0.6 转移到左边或者右边的盒子，
  - ✓ 如果当前盒子是4， 那么各以概率0.5 的概率停留在盒子 4 或转移到盒子 3.
  - ✓ 可得到状态转移矩阵 A :

$$a_{ij} = P(i_{t+1} = q_j | i_t = q_i)$$

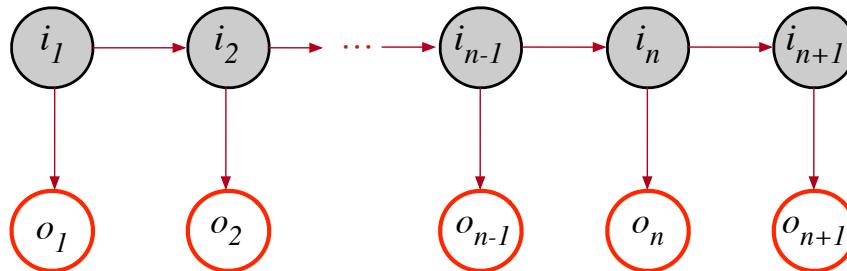
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0.4 & 0 & 0.6 & 0 \\ 0 & 0.4 & 0 & 0.6 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix}$$

# HMM举例

- 按照 **A** 给定的概率选择新的盒子，重复上述过程；最终得到观测序列：“**红红白白红**”。
- 在这个过程中，观察者只能观测到球的颜色的序列，观测不到球是从哪个盒子取出的，即观测不到盒子的序列。
- 状态序列和观测序列的长度 **T=5**



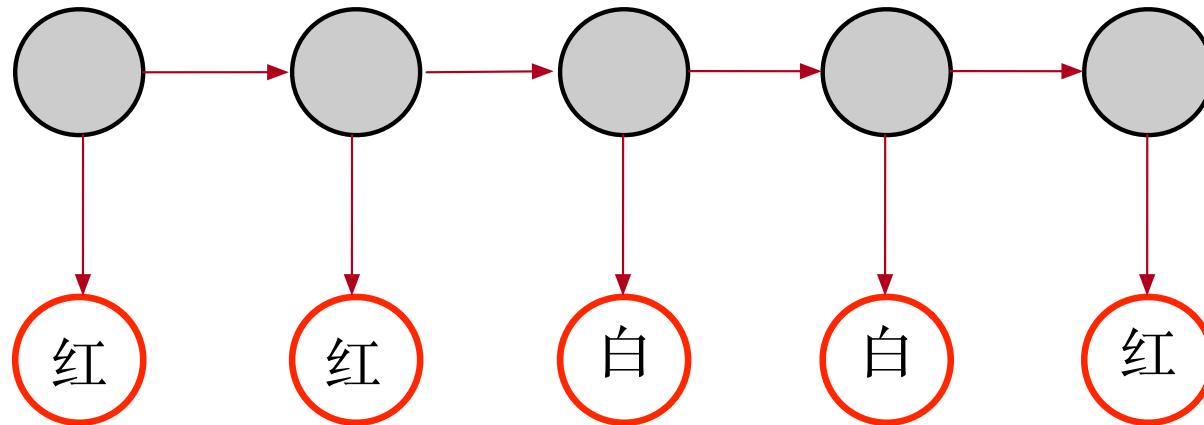
# HMM观测序列生成过程



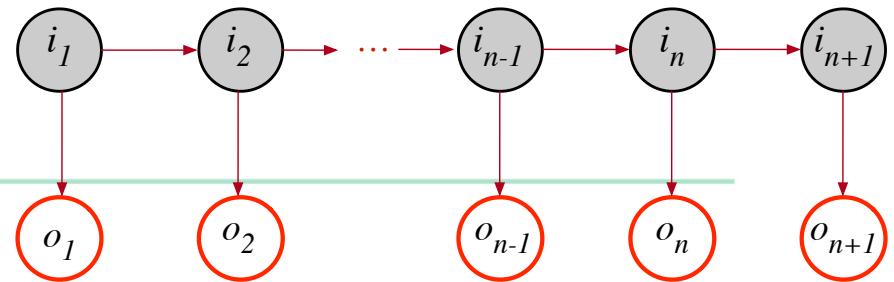
- 输入：HMM参数  $\lambda = \{\pi, A, B\}$ ， 观测序列长度 T
  - 输出：观测序列：  $O = \{o_1, o_2, \dots, o_T\}$
- (1) 按照初始状态分布  $\pi$  产生状态  $i_1$
  - (2) 令  $t = 1$
  - (3) 按照状态  $i_t$  的观测概率  $b_{ik} = P(o_t = v_k | i_t = q_i)$  生成观测  $o_t$
  - (4) 按照状态  $i_t$  的状态转移概率  $a_{ij} = P(i_{t+1} = q_j | i_t = q_i)$  产生状态  $i_{t+1}$
  - (5) 令  $t = t + 1$ ； 如果  $t < T$ , 转步(3)；否则，终止

# 思考：

□ 在给定参数  $\lambda = \{ \pi, A, B \}$  的前提下，得到观测序列“**红红白白红**”的概率是多少？



# HMM的3个基本问题



## □ 概率计算问题：前向-后项算法 – 动态规划

- 给定模型  $\lambda = (A, B, \pi)$  和观测序列  $O = \{o_1, o_2, \dots, o_T\}$ ，计算模型  $\lambda$  下观测序列  $O$  出现的概率  $P(O | \lambda)$

## □ 学习问题：Baum-Welch算法 (状态未知) – EM

- 已知观测序列  $O = \{o_1, o_2, \dots, o_T\}$  估计模型  $\lambda = (A, B, \pi)$  的参数，使得在该模型下观测序列  $P(O | \lambda)$  出现概率最大。参数估计问题！

## □ 预测问题：Viterbi 算法 – 动态规划

- 即解码(decoding)问题：已知模型  $\lambda = (A, B, \pi)$  和观测序列  $O = \{o_1, o_2, \dots, o_T\}$ ，求使条件概率  $P(I | O, \lambda)$  最大的状态序列  $I = \{i_1, i_2, \dots, i_T\}$ 。即给定观测序列，求最有可能的对应的状态序列。

# 概率计算问题

□ 给定模型  $\lambda = (A, B, \pi)$  和观测序列  $O = \{o_1, o_2, \dots, o_T\}$ ，计算模型  $\lambda$  下观测序列  $O$  出现的概率  $P(O | \lambda)$

- 直接算法 -- 暴力算法
- 前向算法
- 后向算法
- 后二者是理解HMM的重点

# 直接计算法

- 给定模型  $\lambda = (A, B, \pi)$  和观测序列  $O = \{o_1, o_2, \dots, o_T\}$ , 计算模型  $\lambda$  下观测序列  $O$  出现的概率  $P(O|\lambda)$ 。
- 按照概率公式, 列举所有可能的长度为  $T$  的状态序列  $I = \{i_1, i_2, \dots, i_T\}$ , 求各个状态序列  $I$  与观测序列  $O = \{o_1, o_2, \dots, o_T\}$  的联合概率  $P(O, I|\lambda)$ ,

$$P(O, I|\lambda) = P(O|I, \lambda)P(I|\lambda)$$

- 然后对所有可能的状态序列求和, 从而得到  $P(O|\lambda)$ 。

$$P(O|\lambda) = \sum_I P(O, I|\lambda) = \sum_I P(O|I, \lambda)P(I|\lambda)$$

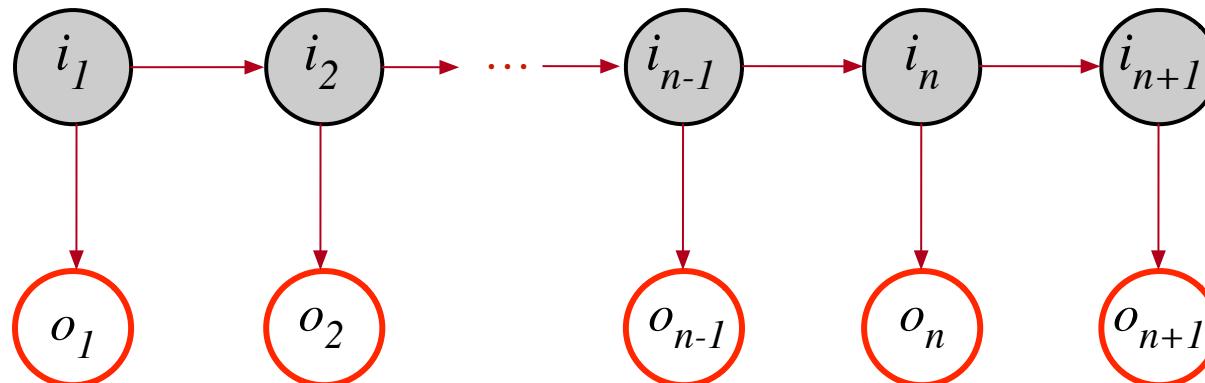
# 直接计算法

□ 状态序列  $I = \{i_1, i_2, \dots, i_T\}$  的概率是：

$$P(I|\lambda) = \pi_{i_1} a_{i_1 i_2} a_{i_2 i_3} \cdots a_{i_{T-1} i_T}$$

□ 对固定的状态序列 I, 观测序列 O 的概率是：

$$P(O|I, \lambda) = b_{i_1 o_1} b_{i_2 o_2} \cdots b_{i_T o_T}$$



# 直接计算法

$$P(I|\lambda) = \pi_{i_1} a_{i_1 i_2} a_{i_2 i_3} \cdots a_{i_{T-1} i_T}$$

□ **O** 和 **I** 同时出现的**联合概率**:  $P(O|I, \lambda) = b_{i_1 o_1} b_{i_2 o_2} \cdots b_{i_T o_T}$

$$P(O, I|\lambda) = P(O|I, \lambda)P(I|\lambda)$$

$$= \pi_{i_1} b_{i_1 o_1} a_{i_1 i_2} b_{i_2 o_2} \cdots a_{i_{T-1} i_T} b_{i_T o_T}$$

□ 对所有可能的状态序列 **I** 求和, 得到**观测序列 O** 的概率 **P(O|λ)**

$$P(O|\lambda) = \sum_I P(O, I|\lambda) = \sum_I P(O|I, \lambda)P(I|\lambda)$$

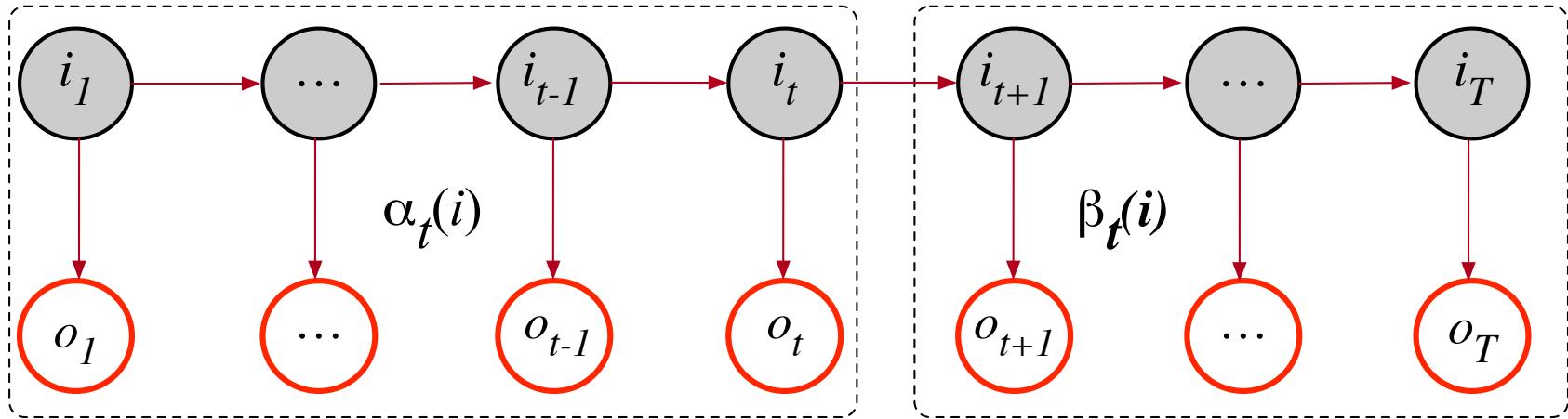
$$= \sum_{i_1, i_2, \dots, i_T} \pi_{i_1} b_{i_1 o_1} a_{i_1 i_2} b_{i_2 o_2} \cdots a_{i_{T-1} i_T} b_{i_T o_T}$$

□ 分析 : 加和符号中有  $2T$  个因子,  $i$  的可能个数是  $N$ ,  $i$  的遍历次数为  $T$ , 总数为  $N^T$ , 因此, 时间复杂度为  $O(T * N^T)$ , 过高。

# 前向算法

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, i_t = q_i | \lambda)$$

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



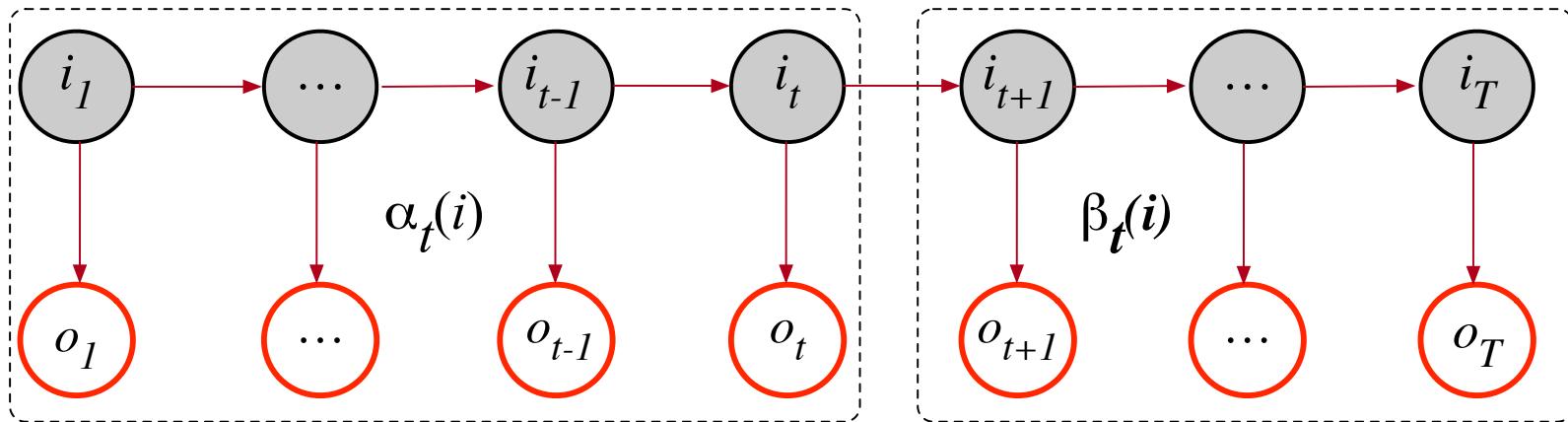
□ 定义：给定  $\lambda$ ，定义到时刻  $t$  部分观测序列为  $o_1, o_2, \dots, o_t$  且状态为  $q_i$  的概率为前向概率，记做：

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, i_t = q_i | \lambda)$$

□ 可以递推求得前向概率  $\alpha_t(i)$  及观测序列概率  $P(O | \lambda)$

# 前向算法

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, i_t = q_i | \lambda)$$



□ 初值 :  $\alpha_1(i) = \pi_i b_{io_1}$

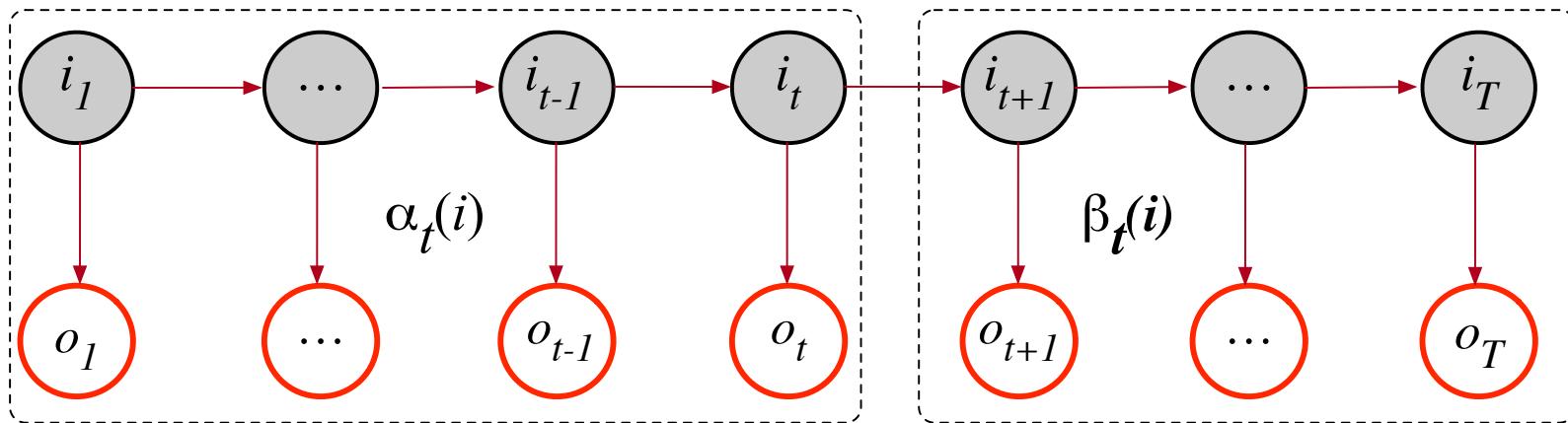
□ 递推 : 对于  $t=1, 2, \dots, T-1$   $\alpha_{t+1}(i) = \left( \sum_{j=1}^N \alpha_t(j) a_{ji} \right) b_{io_{t+1}}$

□ 最终 :  $P(O|\lambda) = \sum_{i=1}^N \underline{\alpha_T(i)}$

# 前向算法

思考：前向概率算法的时间复杂度是**O(TN<sup>2</sup>)**

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, i_t = q_i | \lambda)$$



□ 初值： $\alpha_1(i) = \pi_i b_{io_1}$

□ 递推：对于  $t=1, 2, \dots, T-1$

$$\alpha_{t+1}(i) = \underbrace{\left( \sum_{j=1}^N \alpha_t(j) a_{ji} \right)}_{\text{N个取值}} b_{io_{t+1}}$$

□ 最终： $P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$

**N**个取值      **N**次乘积加和

# 例

□ 考察盒子球模型，状态集合  $Q = \{1, 2, 3\}$ ，利用前向算法计算观测向量  $O = \text{“红白红”}$  的出现概率。

$$\pi = \begin{pmatrix} 0.2 \\ 0.4 \\ 0.4 \end{pmatrix} \quad A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}$$

解

$$\pi = \begin{pmatrix} 0.2 \\ 0.4 \\ 0.4 \end{pmatrix} \quad A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}$$

□ 计算初值  $\alpha_1(i) = \pi_i b_{io_1}$

$$\alpha_1(1) = \pi_1 b_{1o_1} = 0.2 \times 0.5 = 0.1$$

$$\alpha_1(2) = \pi_2 b_{2o_1} = 0.4 \times 0.4 = 0.16$$

$$\alpha_1(3) = \pi_3 b_{3o_1} = 0.4 \times 0.7 = 0.28$$

解

$$\pi = \begin{pmatrix} 0.2 \\ 0.4 \\ 0.4 \end{pmatrix} \quad A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}$$

□ 递推

$$\alpha_{t+1}(i) = \left( \sum_{j=1}^N \alpha_t(j) a_{ji} \right) b_{io_{t+1}}$$

$$\alpha_2(1) = \left( \sum_{j=1}^N \alpha_1(j) a_{j1} \right) b_{1o_2}$$

$$= (0.1 \times 0.5 + 0.16 \times 0.3 + 0.28 \times 0.2) \times 0.5 \\ = 0.077$$

$$\alpha_2(2) = 0.1104$$

$$\alpha_3(1) = 0.04187$$

$$\alpha_2(3) = 0.0606$$

$$\alpha_3(2) = 0.03551$$

$$\alpha_3(3) = 0.05284$$

解

$O = \text{"红白红"}$

□ 最终

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

$$P(O|\lambda) = \sum_{i=1}^3 \alpha_3(i)$$

$$= 0.04187 + 0.03551 + 0.05284$$

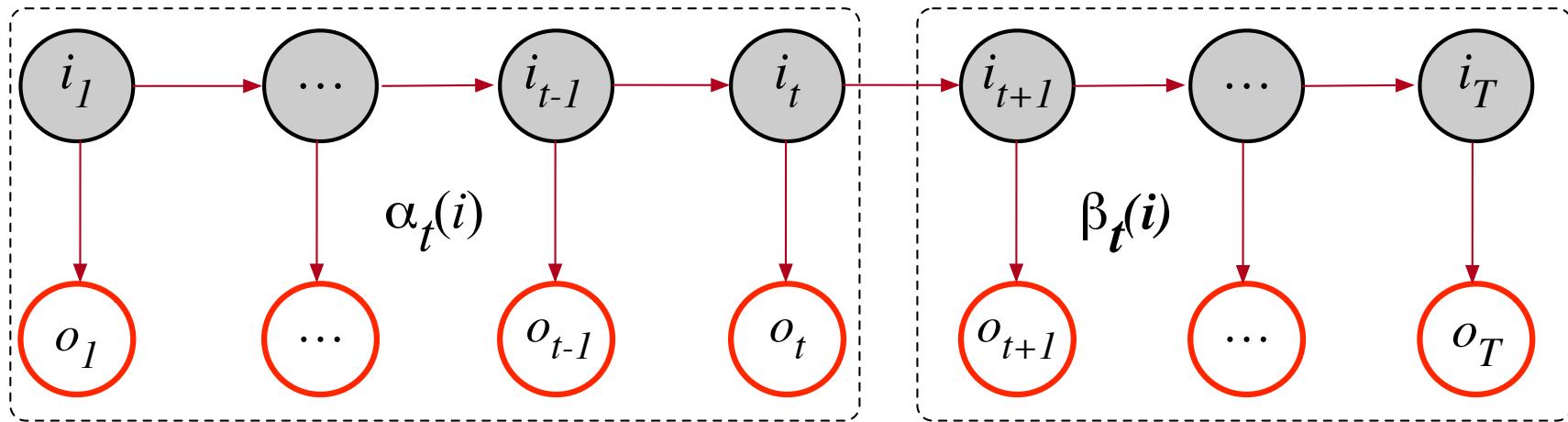
$$= 0.13022$$

观测向量  $O = \text{"红白红"}$  的出现概率

# 后向算法

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, i_t = q_i | \lambda)$$

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



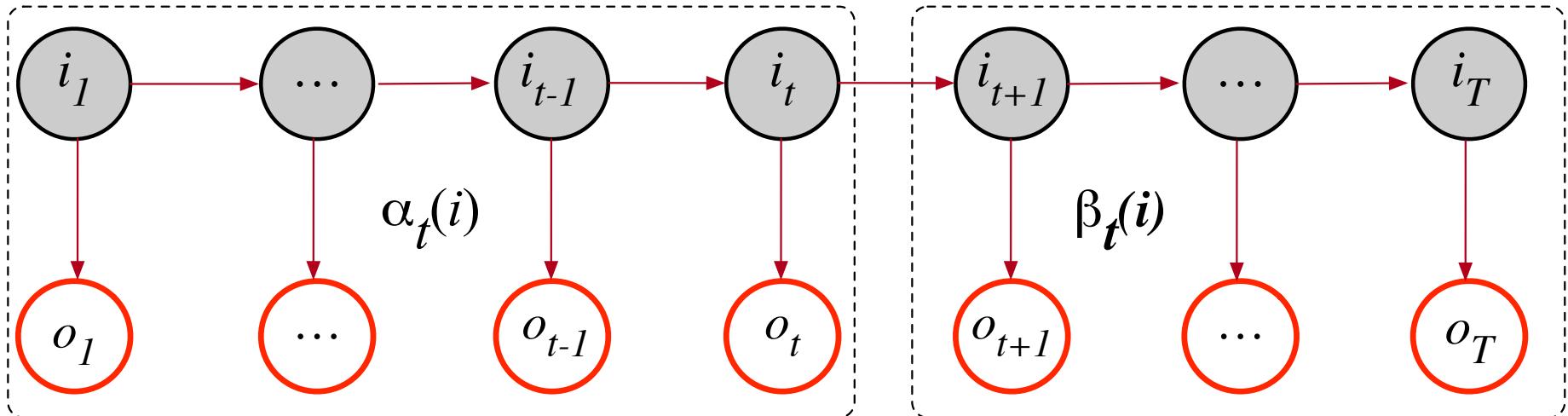
□ 定义：给定  $\lambda$ ，定义到时刻  $t$  状态为  $q_i$  的前提下，从  $t+1$  到  $T$  的部分观测序列为  $o_{t+1}, o_{t+2}, \dots, o_T$  的概率为后向概率，记做：

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

□ 可以递推求得后向概率  $\beta_t(i)$  及观测序列概率  $P(O|\lambda)$

# 后向算法

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



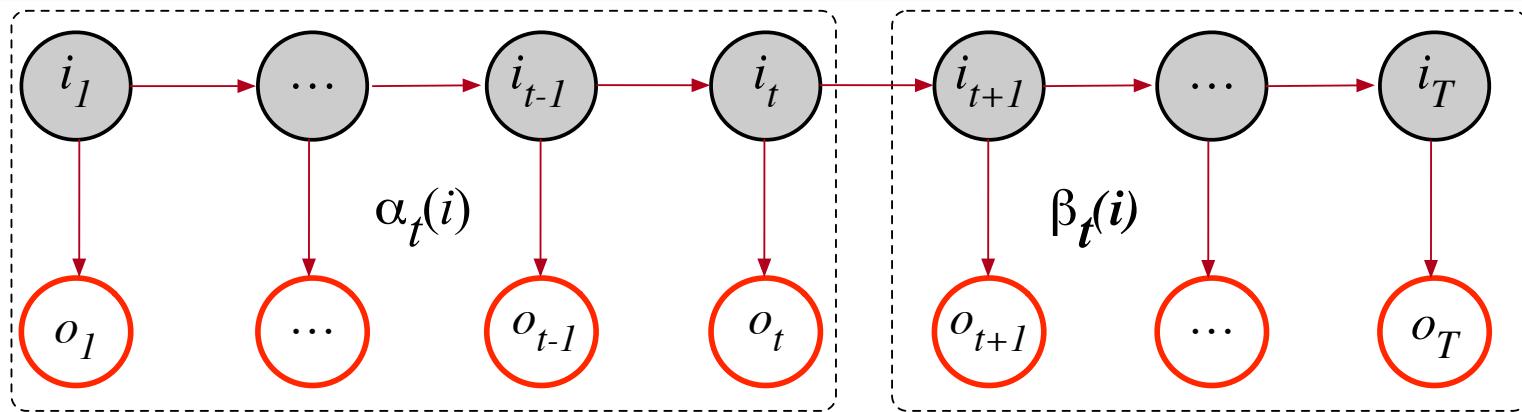
□ 初值 :  $\beta_T(i) = 1$  不要结论, 只要前提, 概率为1。

□ 递推 : 对于  $t = T-1, T-2, \dots, 1$   $\beta_t(i) = \left( \sum_{j=1}^N a_{ij} b_{j o_{t+1}} \beta_{t+1}(j) \right)$

□ 最终 :  $P(O|\lambda) = \sum_{i=1}^N \pi_i b_{i o_1} \beta_1(i)$

# 后向算法的说明

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



$$\beta_t(i) = \left( \sum_{j=1}^N a_{ij} b_{jo_{t+1}} \beta_{t+1}(j) \right)$$

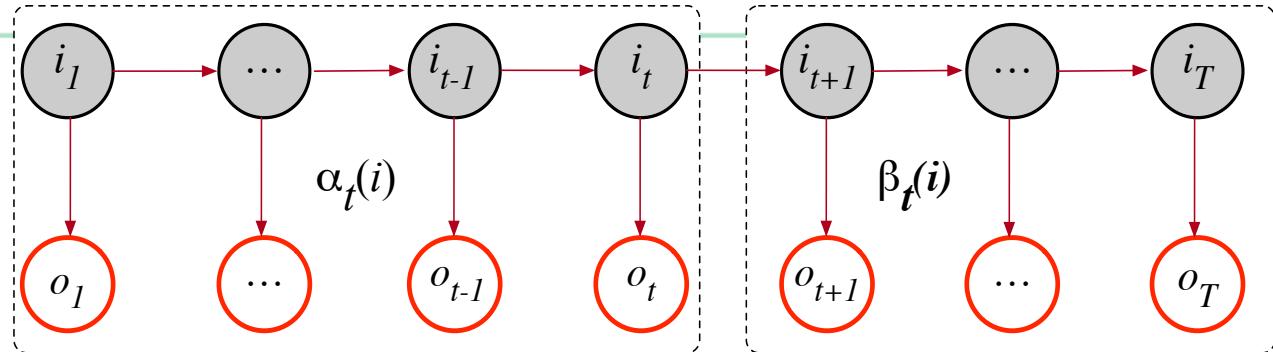
□ 为了计算在时刻  $t$  状态为  $q_i$  条件下时刻  $t+1$  之后的观测序列为  $o_{t+1}, o_{t+2}, \dots, o_T$  的后向概率  $\beta_t(i)$ , 只需要考虑在时刻  $t+1$  所有可能的  $N$  个状态  $q_j$  的转移概率( $a_{ij}$ 项), 以及在此状态下的观测  $o_{t+1}$  的观测概率( $b_{jot+1}$ )项, 然后考虑状态  $q_j$  之后的观测序列的后向概率  $\beta_{t+1}(j)$ 。

# 前向后向关系

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, i_t = q_i | \lambda)$$

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

□根据定义



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

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

$$= P(o_1, \dots, o_t, o_{t+1}, \dots, o_T | i_t = q_i, \lambda) P(i_t = q_i | \lambda)$$

$$= \underline{P(o_1, \dots, o_t | i_t = q_i, \lambda)} P(o_{t+1}, \dots, o_T | i_t = q_i, \lambda) \underline{P(i_t = q_i | \lambda)}$$

$$= P(o_1, \dots, o_t, i_t = q_i | \lambda) P(o_{t+1}, \dots, o_T | i_t = q_i, \lambda)$$

$$= \alpha_t(i) \beta_t(i)$$

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

# 单个状态的概率

- 求给定模型  $\lambda$  和观测  $O$ , 在时刻  $t$  处于状态  $q_i$  的概率。记为 :

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

- 根据前向后向概率的关系:  $P(i_t = q_i, O | \lambda) = \alpha_t(i) \beta_t(i)$

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

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

# $\gamma$ 的意义

□ 给定模型和观测序列，时刻  $t$  处于状态  $q_i$  的概率为：

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

□ 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  $I^* = \{i_1^*, i_2^*, \dots, i_T^*\}$ ，将它作为预测的结果。

# 两个状态的联合概率

□ 求给定模型  $\lambda$  和观测  $O$ , 在时刻  $t$  处于状态  $q_i$  并且时刻  $t+1$  处于状态  $q_j$  的**联合概率**。

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

□ 根据前向后向概率的定义

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, i_t = q_i | \lambda)$$
$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | i_t = q_i, \lambda)$$

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

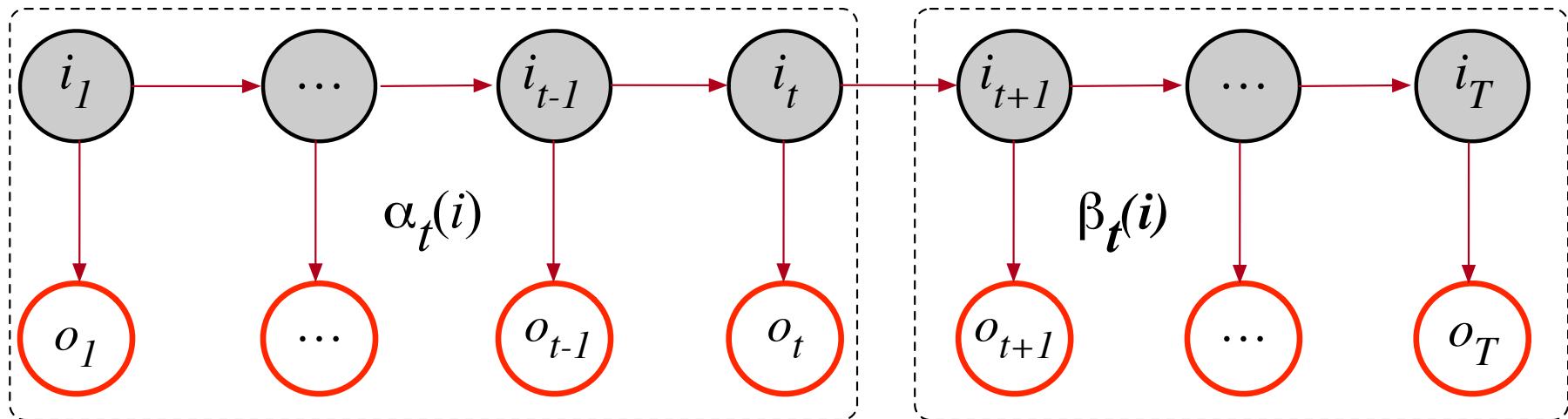
□ 其中,  $P(i_t = q_i, i_{t+1} = q_j, O | \lambda) = \alpha_t(i) a_{ij} b_{j o_{t+1}} \beta_{t+1}(j)$

# 两个状态的联合概率

□ 根据前向后向概率的定义

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, i_t = q_i | \lambda)$$
$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T | i_t = q_i, \lambda)$$

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



# 期望

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

---

□ 在观测  $O$  下状态  $i$  出现的期望：

$$\sum_{t=1}^T \gamma_t(i)$$

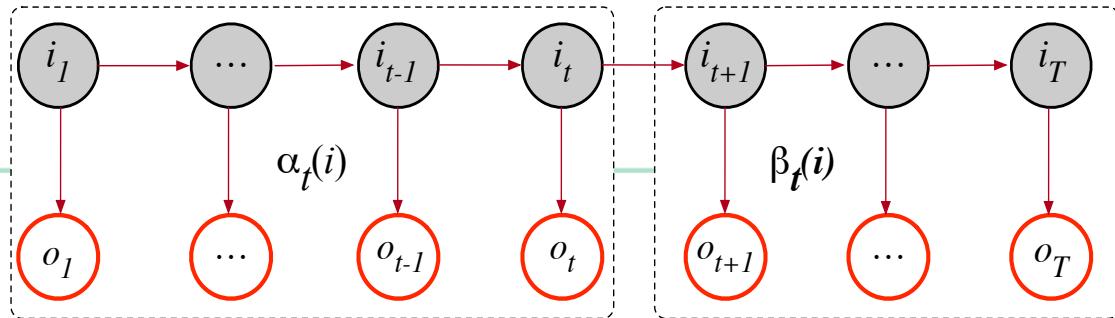
□ 在观测  $O$  下由状态  $i$  转移的期望：

$$\sum_{t=1}^{T-1} \gamma_t(i)$$

□ 在观测  $O$  下状态  $i$  转移到状态  $j$  的期望：

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

# 学习算法



## □ 学习问题：Baum-Welch算法 (状态未知) – EM

- 已知观测序列  $O = \{o_1, o_2, \dots, o_T\}$  估计模型  $\lambda = (A, B, \pi)$  的参数，使得在该模型下观测序列  $P(O|\lambda)$  出现概率最大。参数估计问题！

□ 若训练数据包括观测序列和状态序列，则HMM的学习非常简单，是监督学习；

□ 若训练数据只有观测序列，则HMM的学习需要使用EM算法，是非监督学习。

# 再次分析二项分布的参数估计

---

□ 极大似然估计

□ 简单的例子

- 10次抛硬币的结果是：正正反正正反反正正

□ 假设p是每次抛硬币结果为正的概率。则：

□ 得到这样的实验结果的概率是：

$$\begin{aligned}P &= pp(1-p)ppp(1-p)(1-p)pp \\&= p^7(1-p)^3\end{aligned}$$

# 极大似然估计MLE

□ 目标函数：

□ 最优解是： $p=0.7$

- 即：使用样本的均值可以作为全体的均值估计

$$\max P = \max_{0 \leq p \leq 1} p^7(1-p)^3$$

□ 一般形式：

$$L_p = \prod_x p(x)^{\bar{p}(x)}$$

$p(x)$ 模型是估计的概率分布

$\bar{p}(x)$ 是实验结果的分布

# 直接推广上述结论

---

$$L_p^- = \prod_x p(x)^{\bar{p}(x)}$$

□假设已给定训练数据包含  $S$  个长度相同的观  
测序列和对应的状态序列  $\{(O_1, I_1), (O_2, I_2) \dots (O_s, I_s)\}$ ，那么，可以利用极大似然估计  
的上述结论，给出HMM的参数估计。

# 监督学习方法 – 大数定律

## □ 转移概率 $a_{ij}$ 的估计：

- 设样本中时刻  $t$  处于状态  $i$  且时刻  $t+1$  转移到状态  $j$  的频数为  $A_{ij}$ , 则

$$\hat{a}_{ij} = \frac{A_{ij}}{\sum_{j=1}^N A_{ij}}$$

## □ 观测概率 $b_{ik}$ 的估计：

- 设样本中状态  $i$  且观测为  $k$  的频数为  $B_{ik}$ , 则

$$\hat{b}_{ik} = \frac{B_{ik}}{\sum_{k=1}^M B_{ik}}$$

## □ 初始状态概率 $\pi_i$ 的估计为 $S$ 个样本中初始状态为 $q_i$ 的概率。

# Baum-Welch算法 – late 1960

- 若训练数据只有观测序列，则HMM的学习需要使用EM算法，是非监督学习。
- 所有观测数据写成  $O = (o_1, o_2 \dots o_T)$ ，所有隐数据写成  $I = (i_1, i_2 \dots i_T)$ ，完全数据是  $(O, I) = (o_1, o_2 \dots o_T, i_1, i_2 \dots i_T)$ ，完全数据的对数似然函数是  $\ln P(O, I | \lambda)$
- 假设  $\bar{\lambda}$  是HMM参数的当前估计值， $\lambda$  为待求的参数。

$$\begin{aligned} Q(\lambda, \bar{\lambda}) &= \sum_I \underbrace{\ln P(O, I | \lambda)}_{Q_i} \underbrace{P(I | O, \bar{\lambda})}_{\text{ }} \\ &= \sum_I \ln P(O, I | \lambda) \frac{P(O, I | \bar{\lambda})}{P(O, \bar{\lambda})} \\ &\propto \sum_I \ln P(O, I | \lambda) P(O, I | \bar{\lambda}) \end{aligned}$$

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

# EM算法

## EM算法过程

输入：观测变量数据  $\mathbf{X} = (x_1, \dots, x_n)$ , 隐变量  $\mathbf{Z} = (z_1, \dots, z_n)$ ,

联合分布  $P(\mathbf{X}, \mathbf{Z} | \theta)$ , 条件分布  $P(\mathbf{Z} | \mathbf{X}, \theta)$

输出：模型参数  $\theta$ .

写出似然函数： $L(\theta) = \sum_{i=1}^n \log p(x_i | \theta)$

$$= \sum_{i=1}^n \log \sum_z p(x_i, z_i | \theta)$$

$$\begin{aligned}\sum_i \log p(x_i | \theta) &= \sum_i \log \sum_{z_i} p(x_i, z_i | \theta) \\ &= \sum_i \log \sum_{z_i} Q_i(z_i) \frac{p(x_i, z_i | \theta)}{Q_i(z_i)} \\ &\geq \sum_i \sum_{z_i} Q_i(z_i) \log \frac{p(x_i, z_i | \theta)}{Q_i(z_i)}\end{aligned}$$

迭代地进行：(1) 构造  $L(\theta)$  的一个下界 (lower-bound); E-step

令： $Q_i(z_i) = p((z_i | x_i) | \theta)$

(2) 对这个下界求最大化。 (MLE) M-step

$$\theta := \arg \max_{\theta} \sum_i \sum_{z_i} Q_i(z_i) \log \frac{p(x_i, z_i | \theta)}{Q_i(z_i)}$$

重复这个过程，直到收敛到局部最大值。

# EM过程

□ 根据概率计算暴力公式： $P(O, I | \lambda) = P(O | I, \lambda)P(I | \lambda)$   
 $= \pi_{i_1} b_{i_1 o_1} a_{i_1 i_2} b_{i_2 o_2} \cdots a_{i_{T-1} i_T} b_{i_T o_T}$

□ 函数可写成

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

# 极大化

---

- 极大化Q, 求参数A, B, π
- 由于三个参数分别位于三个项中, 可分别极大化

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

- 注意到  $\pi_i$  满足加和为1, 利用拉格朗日乘子法, 得到

$$\sum_{i=1}^N \ln \pi_i P(O, i_1 = i | \bar{\lambda}) + \gamma \left( \sum_{i=1}^N \pi_i - 1 \right)$$

# 初始状态概率

□ 对上式相对于  $\pi_i$  求偏导，得到：

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

□ 对  $i$  求和，得到：  $\gamma = -P(O | \bar{\lambda})$

时刻  $t$  处于状  
态  $q_i$  的概率

□ 从而得到初始状态概率：

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

$$\pi_i = \frac{P(O, i_1 = i | \bar{\lambda})}{P(O | \bar{\lambda})} = P(i_1 = i | O, \bar{\lambda}) = \frac{\gamma_1(i)}{\sum_{i=1}^N \gamma_1(i)}$$

# 转移概率和观测概率

□ 仍然使用拉格朗日乘子法， 得到转移概率

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

在观测  $O$  下状态  $i$  转移到状态  $j$  的期望

在观测  $O$  下由状态  $i$  转移的期望

□ 同理， 得到

观测概率：

$$b_{ik} = \frac{\sum_{t=1}^T P(O, i_t = i | \bar{\lambda}) I(o_t = v_k)}{\sum_{t=1}^T P(O, i_t = i | \bar{\lambda})} = \frac{\sum_{t=1, o_t = v_k}^T \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)}$$

在观测  $O$  下状态  $i$  出现时观测到  $v_k$  的期望

在观测  $O$  下状态  $i$  出现的期望

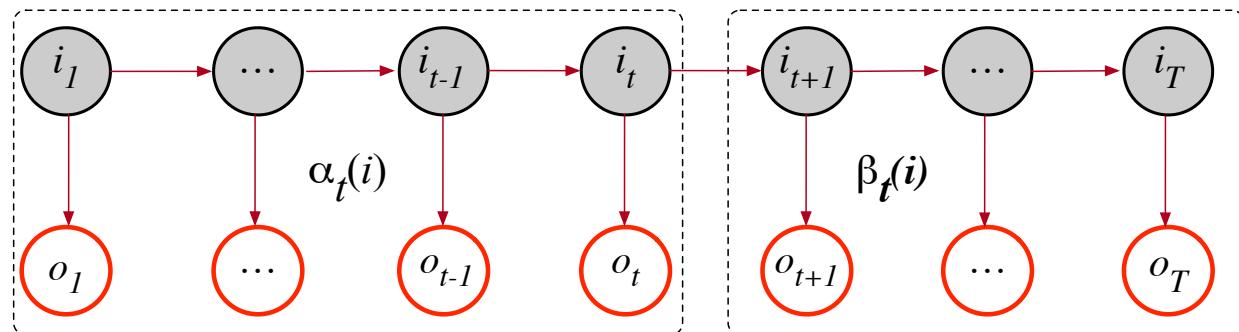
# 预测算法

## □ 预测问题 : Viterbi 算法 – 动态规划

- 即解码(**decoding**)问题 : 已知模型  $\lambda = (A, B, \pi)$  和观测序列  $O = \{o_1, o_2, \dots, o_T\}$  , 求使条件概率  $P(I|O, \lambda)$  最大的状态序列  $I = \{i_1, i_2, \dots, i_T\}$ 。即给定观测序列, 求最有可能的对应的状态序列。

## □ 近似算法

## □ Viterbi 算法



# 预测的近似算法

- 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  $I^* = \{i_1^*, i_2^*, \dots, i_T^*\}$ , 将它作为预测的结果。
- 给定模型和观测序列, 时刻  $t$  处于状态  $q_i$  的概率为 :

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

- 选择概率最大的  $i$  作为最有可能的状态

# 动态规划的经典题目：走棋盘

□ 给定  $m*n$  的矩阵，每个位置是一个非负整数，从左上角开始，每次只能朝右和下走，走到右下角，求总和最大的路径。

A	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	13	0	0	6	0	0
3	0	0	0	0	7	0	0	0
4	0	0	0	14	0	0	0	0
5	0	21	0	0	0	4	0	0
6	0	0	15	0	0	0	0	0
7	0	14	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

B

# 棋盘走法的状态转移函数

- 走的方向决定了：同一个格子一定不会经过两次。
- $dp[0,0]=a[0,0]$
- $dp[x,y] = \max($ 
  - $dp[x-1,y] + a[x,y]$
  - $dp[x,y-1] + a[x,y]$
  - )

A	1	2	3	4	5	6	7	8	B
1	0	0	0	0	0	0	0	0	
2	0	0	13	0	0	6	0	0	
3	0	0	0	0	7	0	0	0	
4	0	0	0	14	0	0	0	0	
5	0	21	0	0	0	4	0	0	
6	0	0	15	0	0	0	0	0	
7	0	14	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	

# Viterbi 算法

- Viterbi 算法实际是用动态规划解 HMM 预测问题，用 DP 求概率最大的路径(最优路径)，这是一条路径对应一个状态序列。
- 定义变量  $\delta_t(i)$ ：在时刻  $t$  状态为  $i$  的所有路径中，概率的最大值。

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

- 定义在时刻  $t$  状态为  $i$  的所有单个路径中概率最大的路径的第  $t-1$  个节点为：

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

# Viterbi算法

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

□ 递推 :  $\delta_1(i) = \pi_i b_{io_1} \quad \psi_1(i) = 0$

$$\begin{aligned}\delta_{t+1}(i) &= \max_{i_1, i_2, \dots, i_t} P(i_{t+1} = i, i_t, \dots, i_1, o_{t+1}, \dots, o_1 | \lambda) \\ &= \max_{1 \leq j \leq N} (\delta_t(j) a_{ji}) b_{io_{t+1}}\end{aligned}$$

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

□ 终止 :

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

# 例

---

□ 考察盒子球模型，观测向量  $O=“红白红”$ ，试求最优状态序列。

$$\pi = \begin{pmatrix} 0.2 \\ 0.4 \\ 0.4 \end{pmatrix} \quad A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}$$

# 解

$$\pi = \begin{pmatrix} 0.2 \\ 0.4 \\ 0.4 \end{pmatrix} \quad A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}$$

□ 初始化：

□ 在  $t=1$  时，对于每一个状态  $i$ ，求状态为  $i$  观测到  $O_1 = \text{红}$  的概率，记此概率为  $\delta_1(i)$

$$\delta_1(i) = \pi_i b_{io_1} = \pi_i b_{i\text{红}}$$

□ 求得  $\delta_1(1) = 0.2 \times 0.5 = 0.1 \quad \psi_1(1) = 0$

$$\delta_1(2) = 0.4 \times 0.4 = 0.16 \quad \psi_1(2) = 0$$

$$\delta_1(3) = 0.4 \times 0.7 = \underline{0.28} \quad \max \quad \psi_1(3) = 0$$

解

$$\pi = \begin{pmatrix} 0.2 \\ 0.4 \\ 0.4 \end{pmatrix} \quad A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix} \quad B = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}$$

□ 在  $t=2$  时，对每个状态  $i$ ，求在  $t=1$  时状态为  $j$  观测为红并且在  $t=2$  时状态为  $i$  观测为白的路径的最大概率，记此概率为  $\delta_2(i)$ ，则：

$$\begin{aligned}\delta_{t+1}(i) &= \max_{1 \leq j \leq 3} (\delta_1(j)a_{ji}) b_{io_2} \\ &= \max_{1 \leq j \leq 3} (\delta_1(j)a_{ji}) b_{i\text{白}}\end{aligned}$$

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

□ 求得  $\delta_2(1) = \max(\delta_1(1) \times a_{11}, \delta_1(2) \times a_{21}, \delta_1(3) \times a_{31}) \times b_{1\text{白}}$   
 $= \max(0.1 \times 0.5, 0.16 \times 0.3, 0.28 \times 0.2) \times 0.5 = 0.028$

$$\begin{aligned}\delta_2(2) &= \max \psi_2(2) = 3 & \psi_2(1) &= 3 \\ \delta_2(3) &= \max \psi_2(3) = 3 & \psi_2(2) &= 3\end{aligned}$$

# 解

---

□ 同理，求得

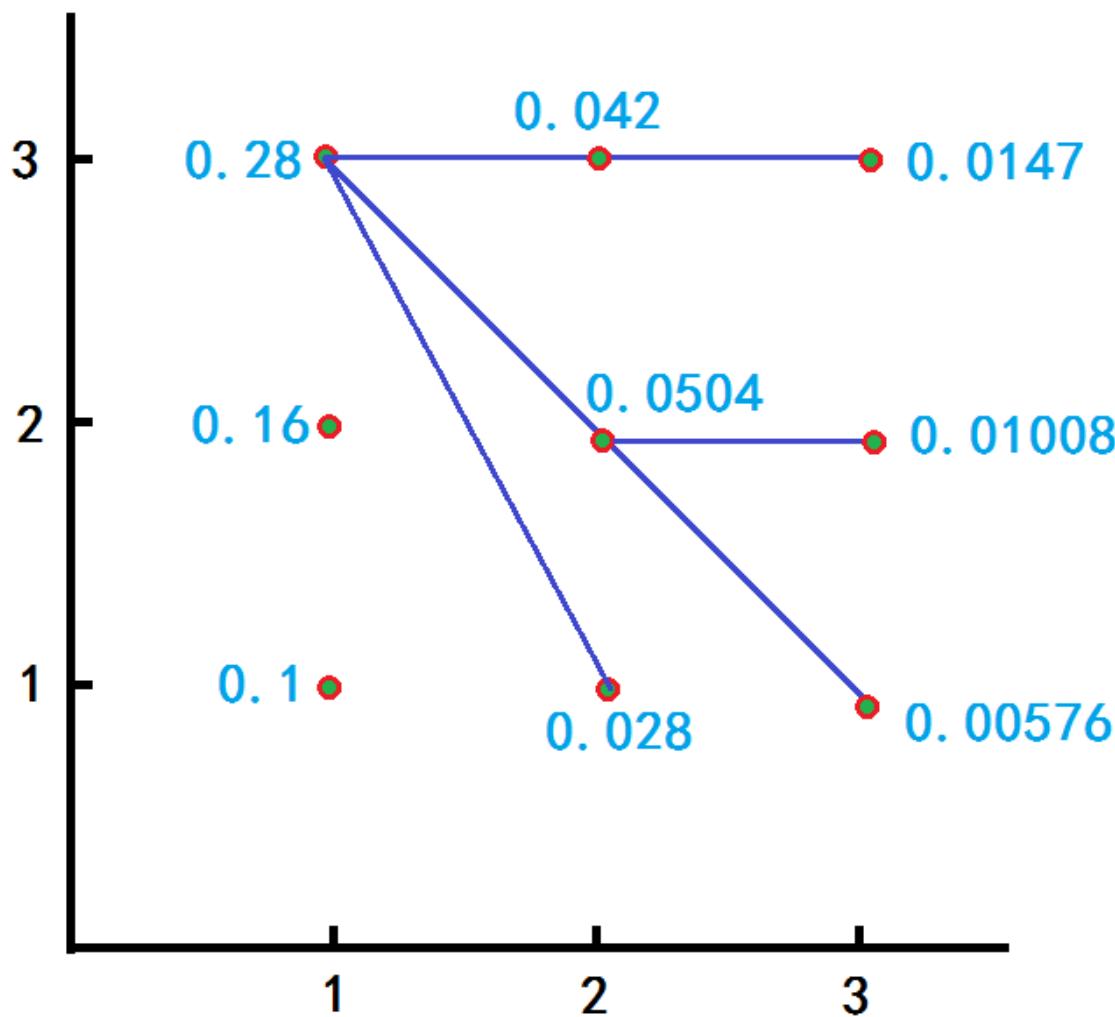
$$\delta_3(1) = 0.00756 \quad \psi_3(1) = 2$$

$$\delta_3(2) = 0.01008 \quad \psi_3(2) = 2$$

$$\delta_3(3) = \underline{0.0147} \quad \max \quad \psi_3(3) = 3$$

□ 从而，最大是  $\delta_3(3) = 0.0147$ ，根据每一步的最大，得到序列是 (3,3,3)

# 求最优路径图解



# 参考文献

---

- 统计学习方法， 李航著， 清华大学出版社， 2012年
- Pattern Recognition and Machine Learning  
Chapter 13, M. Jordan, J. Kleinberg, ect,  
2006
- A Tutorial on Learning With Bayesian  
Networks, David Heckerman, 1996
- Radiner L, Juang B. An introduction of hidden  
markov Models. IEEE ASSP Magazine,  
January 1986