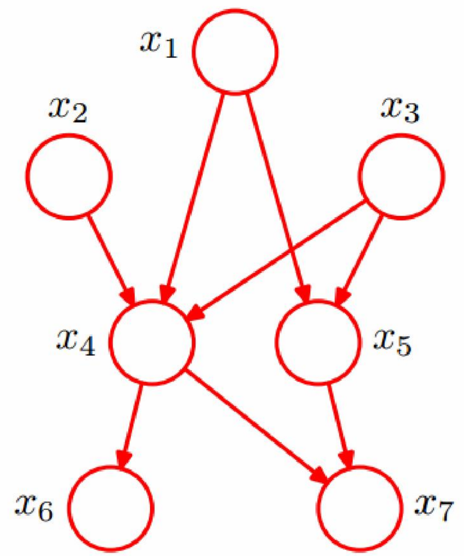
# 隐马尔科夫模型HMM

七月算法 邹博

2015年5月10日

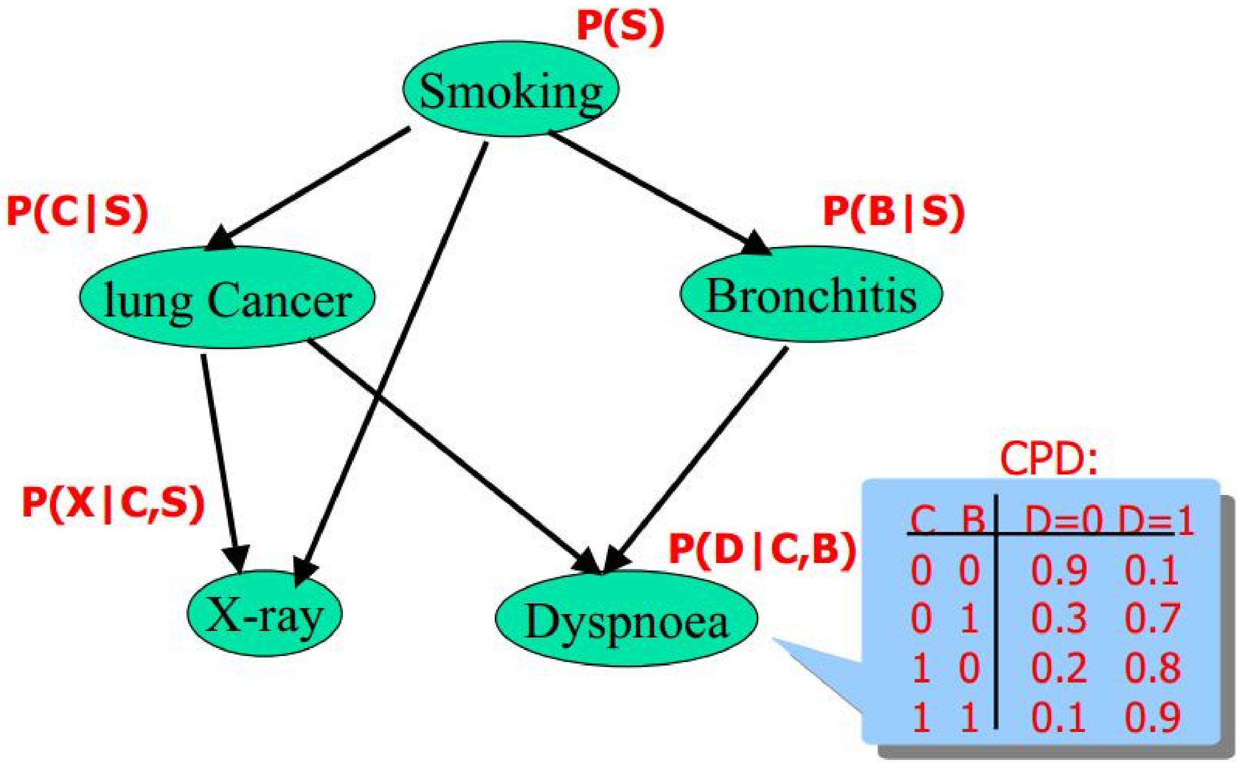
# 复习：贝叶斯网络

- x1和x2独立
- x6和x7在x4给定的条件下独立
- x1,x2,…x7的联合分布：




$$p(x_1)p(x_2)p(x_3)p(x_4|x_1,x_2,x_3)p(x_5|x_1,x_3)p(x_6|x_4)p(x_7|x_4,x_5)$$

# 贝叶斯网络分析



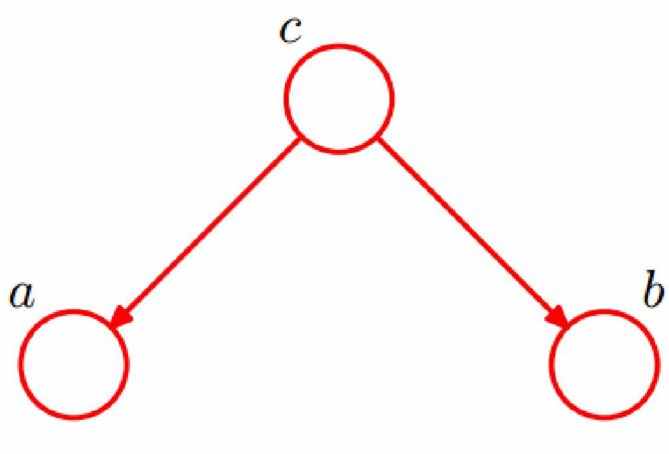

$1+2+2+4+4=13$ vs $2^5$

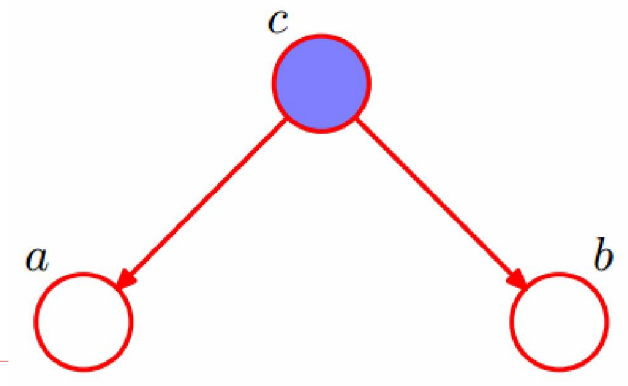
# 复习：特殊的贝叶斯网络


- M个离散结点形成一条链，每一个结点有K个状态，则需要K-1+(M-1)K(K-1)个参数。这是关于长度M的线性函数。
  - 别忘了，如果是全连接，需要$K^M$-1个参数，是关于M的指数函数。
- 这个网络被称作马尔科夫模型。

# 复习：通过贝叶斯网络判定条件独立—1

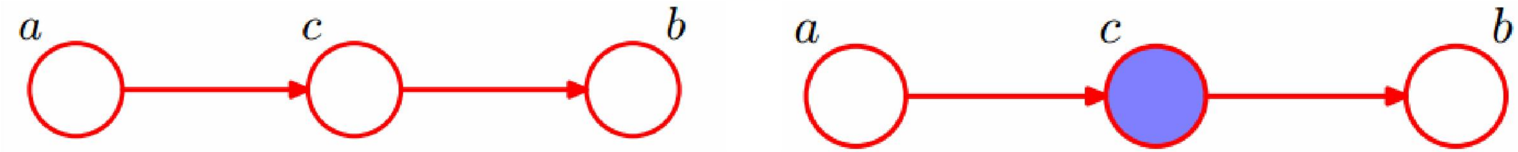
- P(a,b,c)=P(c)*P(a|c)*P(b|c)
- 则：P(a,b|c)=P(a,b,c)/P(c)
- 带入，得到：
- P(a,b|c)=P(a|c)*P(b|c)
- 即：在c给定的条件下，a，b被阻断(blocked)，是独立的。
  - 条件独立：tail-to-tail



# 复习：通过贝叶斯网络判定条件独立—2

- P(a,b,c)=P(a)*P(c|a)*P(b|c)




$$
\begin{aligned}
& P(a, b \mid c) \\
= \ & P(a, b, c) / P(c) \\
= \ & P(a) * P(c \mid a) * P(b \mid c) \ / \ P(c) \\
= \ & P(a, c) * P(b \mid c) \ / \ P(c) \\
= \ & P(a \mid c) * P(b \mid c)
\end{aligned}
$$

- 即：在c给定的条件下，a，b被阻断(blocked)，是独立的。
  - 条件独立：head-to-tail

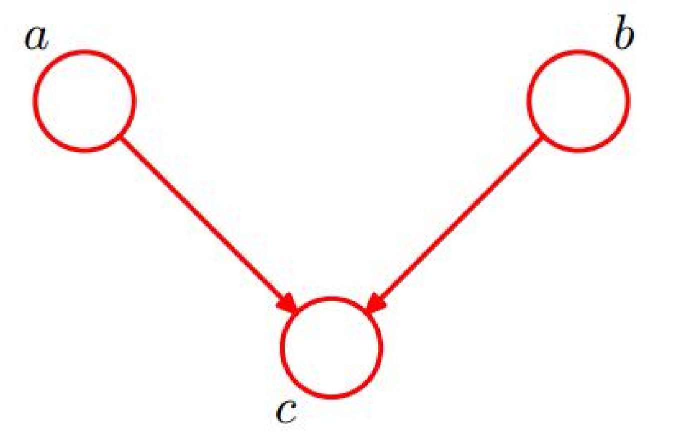
# 复习：通过贝叶斯网络判定条件独立—3

- P(a,b,c) = P(a)*P(b)*P(c|a,b)

$$\sum_c P(a,b,c) = \sum_c P(a)*P(b)*P(c|a,b)$$

$$\Rightarrow P(a,b) = P(a)*P(b)$$

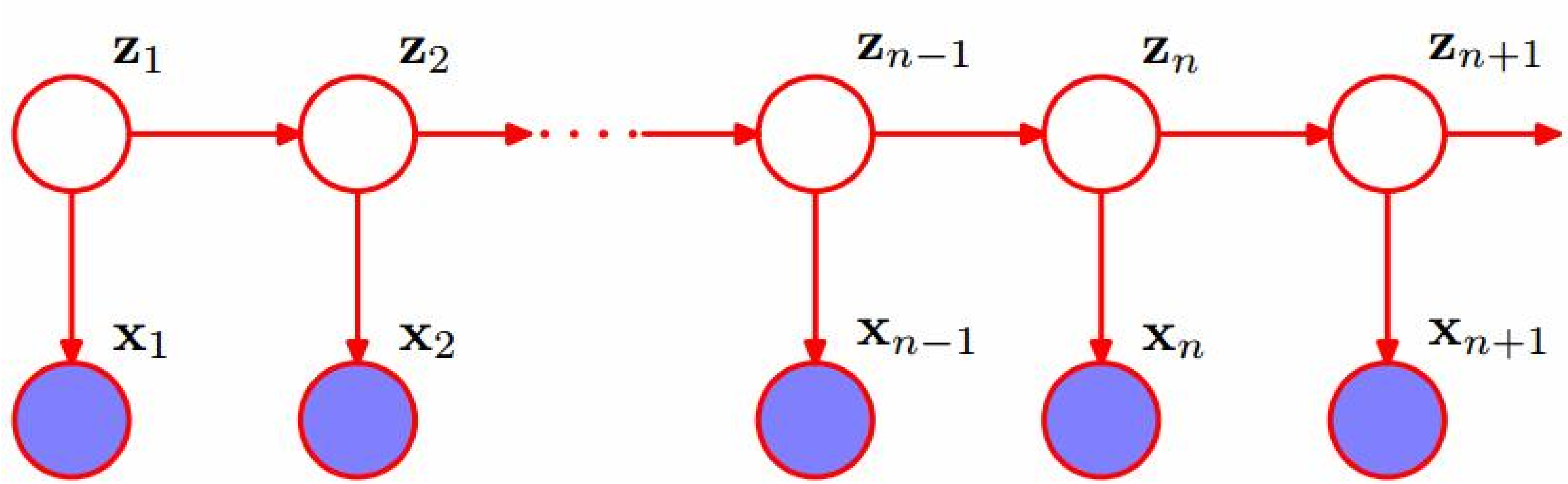
- 在c未知的条件下，a，b被阻断(blocked)，是独立的：head-to-head



# 隐马尔科夫模型的定义

- 隐马尔科夫模型(HMM, Hidden Markov Model)可用标注问题，在语音识别、NLP、生物信息、模式识别等领域被实践证明是有效的算法。
- HMM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z1给定的前提下，x1和z2独立吗？x1和x2独立吗？

# HMM的确定

- HMM由初始概率分布π、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
- V是所有可能的观测的集合
  - M是可能的观测数

$$Q = \{q_1, q_2, \ldots q_N\}$$

$$V = \{v_1, v_2, \ldots v_M\}$$

# HMM的参数

- I是长度为T的状态序列，O是对应的观测序列

$$I = \{i_1, i_2, \ldots i_T\} \qquad O = \{o_1, o_2, \ldots o_T\}$$

- A是状态转移概率矩阵

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

- 其中 $a_{ij} = P\left(i_{t+1} = q_j \middle| i_t = q_i\right)$
- aij是在时刻t处于状态qi的条件下时刻t+1转移到状态qj的概率。

# HMM的参数

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

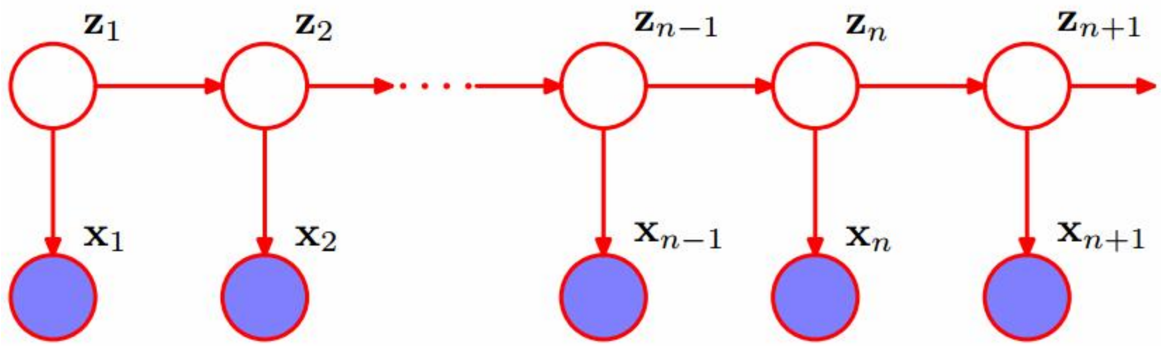
# HMM的参数总结

- HMM由初始概率分布π、状态转移概率分布A以及观测概率分布B确定。π和A决定状态序列，B决定观测序列。因此，HMM可以用三元符号表示，称为HMM的三要素：

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

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

# HMM的两个基本性质




- 齐次假设：

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

- 观测独立性假设：

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

# HMM举例

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

| 盒子号 | 1 | 2 | 3 |
|---|---|---|---|
| 红球数 | 5 | 4 | 7 |
| 白球数 | 5 | 6 | 3 |

- 按照下面的方法抽取小球，得到球颜色的观测序列：按照(0.2,0.4,0.4)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；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"。

# 该示例的各个参数

- 状态集合：Q={盒子1，盒子2，盒子3}
- 观测集合：V={红，白}
- 状态序列和观测序列的长度T=5
- 初始概率分布 $\pi$ ：
- 状态转移概率分布A：
- 观测概率分布B：

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

## 思考：

- 在给定参数π、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
  - 给定模型 $\lambda=(A,B,\pi)$ 和观测序列 $O=\{o_1,o_2,\ldots o_T\}$，计算模型λ下观测序列O出现的概率P(O| λ)
- 学习问题
  - 已知观测序列 $O=\{o_1,o_2,\ldots o_T\}$，估计模型 $\lambda=(A,B,\pi)$ 的参数，使得在该模型下观测序列P(O| λ)最大
- 预测问题
  - 即解码问题：已知模型 $\lambda=(A,B,\pi)$ 和观测序列 $O=\{o_1,o_2,\ldots o_T\}$，求对给定观测序列条件概率P(I|O)最大的状态序列I

# 概率计算问题

- 直接算法
  - 暴力算法
- 前向算法
- 后向算法
  - 这二者是理解HMM的算法重点

# 直接计算法

- 按照概率公式，列举所有可能的长度为T的状态序列 $I=\{i_1,i_2,...i_T\}$，求各个状态序列I与观测序列 $O=\{o_1,o_2,...o_T\}$的联合概率P(O,I|λ)，然后对所有可能的状态序列求和，从而得到P(O|λ)

# 直接计算法

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

$$P(I|\lambda)=\pi_{i_1}a_{i_1i_2}a_{i_2i_3}\cdots a_{i_{T-1}i_T}$$

- 对固定的状态序列I，观测序列O的概率是：

$$P(O|I,\lambda)=b_{i_1o_1}b_{i_2o_2}\cdots b_{i_To_T}$$

# 直接计算法

- O和I同时出现的联合概率是：

$$P(O,I|\lambda)=P(O,I|\lambda)P(I|\lambda)$$
$$=\pi_{i_1}b_{i_1o_1}a_{i_1i_2}b_{i_2o_2}\cdots a_{i_{T-1}i_T}b_{i_To_T}$$

- 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)

$$P(O|\lambda)=\sum_I P(O,I|\lambda)=\sum_I P(O|I,\lambda)P(I|\lambda)$$
$$=\sum_{i_1,i_2,\cdots i_T}\pi_{i_1}b_{i_1o_1}a_{i_1i_2}b_{i_2o_2}\cdots a_{i_{T-1}i_T}b_{i_To_T}$$

# 直接计算法

- 对于最终式

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

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

- 分析：加和符号中有2T个因子，I的遍历个数为$N^T$，因此，时间复杂度为$O(T N^T)$，过高。

# 借鉴算法的优化思想

- 最长递增子序列
- KMP中next数组的计算

# 前向算法

- 定义：给定λ，定义到时刻t部分观测序列为o1,o2…ot且状态为qi的概率为前向概率，记做：

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

- 可以递推的求得前向概率 $\alpha_t(i)$ 及观测序列概率P(O| λ )

# 前向算法

- 初值：$\alpha_1(i)=\pi_i b_{io_1}$

- 递推：对于t=1,2…T-1

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

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

# 前向算法

- 思考：前向概率算法的时间复杂度是$O(TN^2)$

# 例

- 考察盒子球模型，计算观测向量O=“红白红”的出现概率。

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

# 解

- 计算初值

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

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

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

# 解

## 递推

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

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

$$= 0.077$$

$$\alpha_2(2) = 0.1104$$

$$\alpha_2(3) = 0.0606$$

$$\alpha_3(1) = 0.04187$$

$$\alpha_3(2) = 0.03551$$

$$\alpha_3(3) = 0.05284$$

# 解

- 最终

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

$$=0.04187+0.03551+0.05284$$

$$=0.13022$$

# 后向算法

- 定义：给定λ，定义到时刻t状态为qi的前提下，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，记做：

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

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

# 后向算法

- 初值： $\beta_T(i)=1$
- 递推：对于t=T-1,T-2…,1

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

- 最终： $P(O|\lambda)=\sum_{i=1}^{N}\pi_i b_{io_1}\beta_1(i)$

# 后向算法的说明

- 为了计算在时刻t状态为qi条件下时刻t+1之后的观测序列为$o_{t+1},o_{t+2}\ldots o_T$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qj的转移概率(aij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qj之后的观测序列的后向概率$\beta_{t+1}(j)$

# 前向后向概率的关系

- 根据定义，证明下列等式

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

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

# 单个状态的概率

- 求给定模型λ和观测O，在时刻t处于状态qi的概率。
- 记：$\gamma_t(i) = P(i_t = q_i | O, \lambda)$

# 单个状态的概率

- 根据前向后向概率的定义，

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

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

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

# γ的意义

- 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。
- 给定模型和观测序列，时刻t处于状态qi的概率为：

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

# 两个状态的联合概率

- 求给定模型λ和观测O，在时刻t处于状态qi并且时刻t+1处于状态qj的概率。

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

# 两个状态的联合概率

- 根据前向后向概率的定义，

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

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

# 期望

- 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

- 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，是监督性学习；
- 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：$\max P = \max\limits_{0 \le p \le 1} p^7 (1-p)^3$
- 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
- 一般形式：

$$L_{\overline{p}} = \prod_x p(x)^{\overline{p}(x)}$$

$p(x)$模型是估计的概率分布

$\overline{p}(x)$是实验结果的分布

# 直接推广上述结论

- 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$\{(O_1,I_1), (O_2,I_2)\ldots(O_s,I_s)\}$，那么，可以直接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

- 转移概率aij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Aij，则

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

- 观测概率bik的估计：
  - 设样本中状态i并观测为k的频数为Bik，则

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

- 初始状态概率πi的估计为S个样本中初始状态为qi的概率。

# Baum-Welch算法

- 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。

# Baum-Welch算法

- 所有观测数据写成$O=(o_1,o_2 \ldots o_T)$，所有隐数据写成$I=(i_1,i_2 \ldots i_T)$，完全数据是$(O,I)=(o_1,o_2 \ldots o_T ,i_1,i_2 \ldots i_T)$，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，λ为待求的参数。

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

# EM过程

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

$$=\pi_{i_1}b_{i_1o_1}a_{i_1i_2}b_{i_2o_2}\cdots a_{i_{T-1}i_T}b_{i_To_T}$$

- 函数可写成 $Q(\lambda,\overline{\lambda})=\sum_I \ln P(O,I|\lambda)P(O,I|\overline{\lambda})$

$$=\sum_I \ln \pi_{i_1}P(O,I|\overline{\lambda})$$

$$+\sum_I\left(\sum_{t=1}^{T-1}\ln a_{i_ti_{t+1}}\right)P(O,I|\overline{\lambda})$$

$$+\sum_I\left(\sum_{t=1}^{T}\ln b_{i_to_t}\right)P(O,I|\overline{\lambda})$$

# 极大化

- 极大化Q，求的参数A,B,π
- 由于该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πi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πi求偏导，得到：

$$P\left(O, i_1 = i \middle| \overline{\lambda}\right) + \gamma \pi_i = 0$$

- 对i求和，得到：

$$\gamma = -P\left(O \middle| \overline{\lambda}\right)$$

- 从而得到初始状态概率：

$$\pi_i = \frac{P\left(O, i_1 = i \middle| \overline{\lambda}\right)}{P\left(O \middle| \overline{\lambda}\right)}$$

# 转移概率和观测概率

- 第二项可写成：

$$\sum_{I}\left(\sum_{t=1}^{T-1}\ln a_{i_t i_{t+1}}\right)P\left(O,I\middle|\overline{\lambda}\right)=\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T-1}\ln a_{ij}P\left(O,i_t=i,i_{t+1}=j\middle|\overline{\lambda}\right)$$

- 仍然使用拉格朗日乘子法，得到

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

- 同理，得到：

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

# 预测算法

- 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$I^*=\{i_1^*, i_2^* \cdots i_T^*\}$，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qi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
  - 会出现此状态在实际中可能不会发生的情况

# 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。

向右

向下

| 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]
  - )

# 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δi(t)：在时刻t状态为i的所有路径中，概率的最大值。

# Viterbi算法

- 定义：

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

- 递推：

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

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

$$=\max_{1\le j\le N}\left(\delta_t(j)a_{ji}\right)b_{io_{t+1}}$$

- 终止：

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

# 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

- 初始化：
- 在t=1时，对于每一个状态i，求状态为i观测到o1=红的概率，记此概率为 $\delta_1(t)$

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

- 求得 $\delta_1(1)=0.1$
- $\delta_1(2)=0.16$
- $\delta_1(3)=0.28$

# 解

- 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，记次概率为δ$_2$(t)，则：

$$\delta_{t+1}(i)=\max_{1\le j\le 3}\left(\delta_1(j)a_{ji}\right)b_{io_2}$$
$$=\max_{1\le j\le 3}\left(\delta_1(j)a_{ji}\right)b_{i白}$$

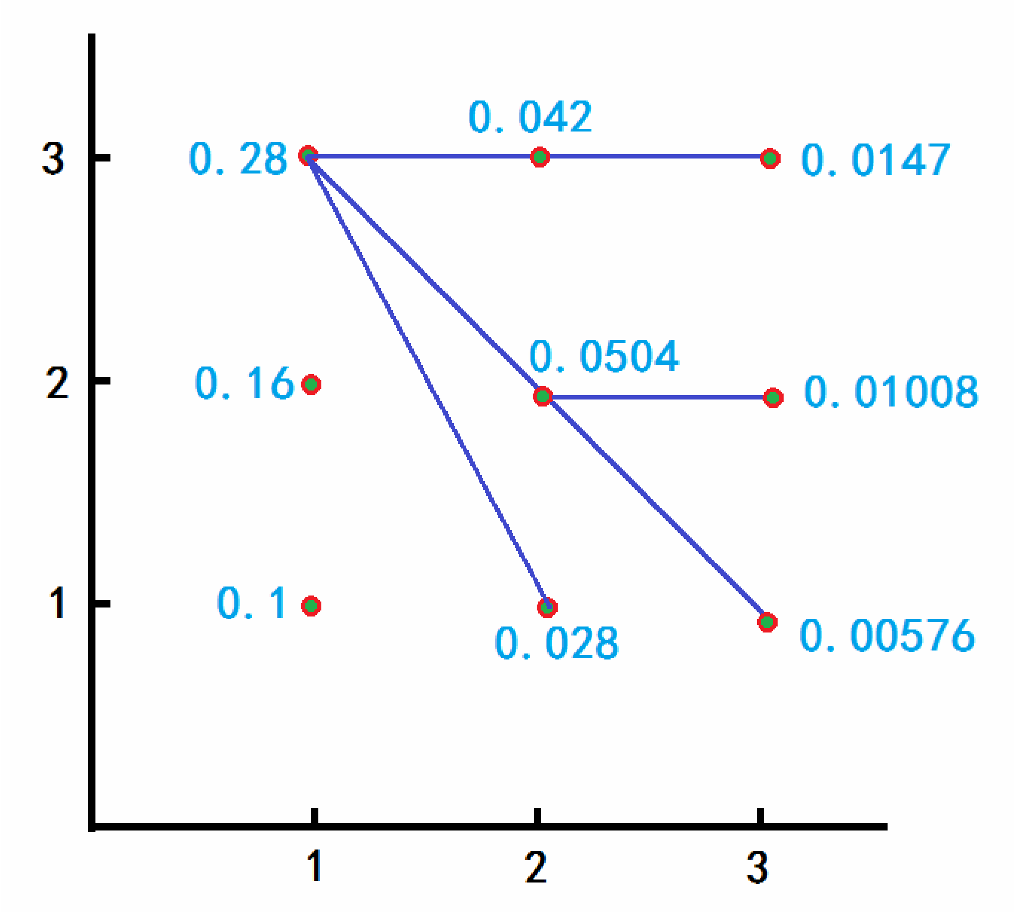
- 求得δ$_2$(1)=0.028
- δ$_2$(2)=0.0504
- δ$_2$(3)=0.042

# 解

- 同理，求得
- $\delta_3(1)=0.00756$
- $\delta_3(2)=0.01008$
- $\delta_3(3)=0.0147$

- 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, Bishop M, Springer-Verlag, 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

# 我们在这里

- 更多算法面试题在 7 七月算法
  - http://www.julyedu.com/
    - 免费视频
    - 直播课程
    - 问答社区

- contact us：微博
  - @研究者July
  - @七月问答
  - @邹博_机器学习

感谢大家！

恳请大家批评指正！