第6章　马尔科夫模型与条件随机场

6.1 马尔可夫模型

6.1 马尔可夫模型

◆马尔可夫模型描述

存在一类重要的随机过程：如果一个系统有 N 个状态 $S1$, $S2$,…, SN, 随着时间的推移，该系统从某一状态转移到另一状态。

如果用q_t表示系统在时间 t 的状态变量，那么，t 时刻的状态取值为 Sj $(1\leq j\leq N)$ 的概率取决于前 t-1 个时刻 (1, 2, …, t-1) 的状态，该概率为：

$$p(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, \cdots)$$

简单说：马尔科夫过程就是指过程中的每个状态的转移只依赖于之前的 n个状态

6.1 马尔可夫模型

为控制复杂性，我们对其进行简化

● 假设1：

如果在特定情况下，系统在时间 *t* 的状态只与其在时间 *t*-1 的状态相关，则该系统构成一个离散的一阶马尔可夫链：

$$p(q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, \cdots) = \boxed{p(q_t = S_j \mid q_{t-1} = S_i)} \quad \ldots (6.1)$$

6.1 马尔可夫模型

- 假设2：

如果只考虑公式(6.1)独立于时间 t 的随机过程，即所谓的不动性假设，状态与时间无关，那么：

$$p(q_t = S_j \mid q_{t-1} = S_i) = a_{ij}, \qquad 1 \le i, j \le N \qquad \ldots (6.2)$$

该随机过程称为(一阶)马尔可夫模型(**Markov Model**)。

6.1 马尔可夫模型

在马尔可夫模型中，状态转移概率 a_{ij} 必须满足下列条件：

$$a_{ij} \geq 0 \qquad \ldots (6.3)$$

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad \ldots (6.4)$$

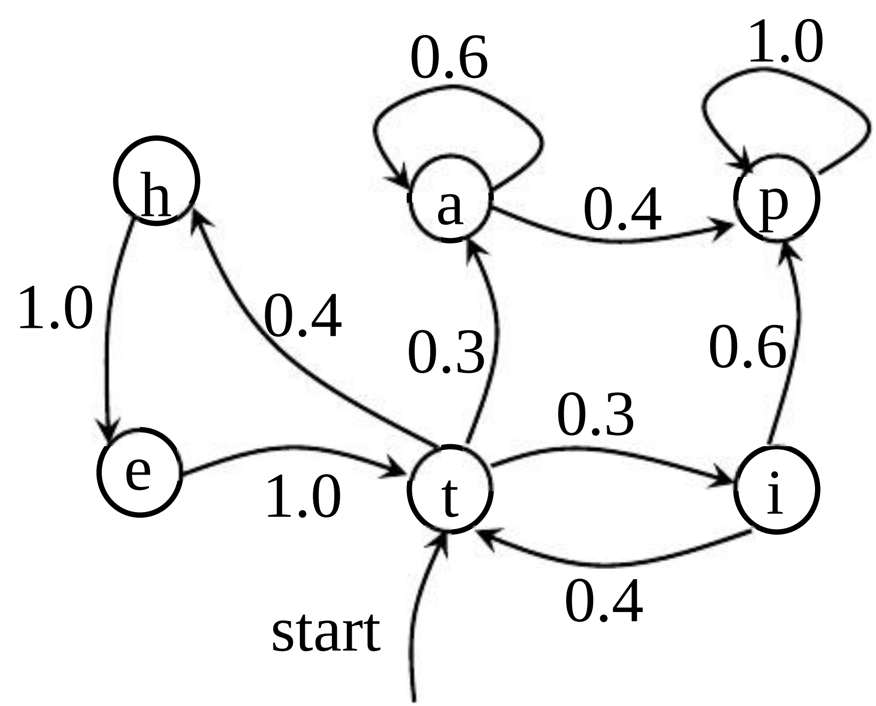
马尔可夫模型可视为随机的有穷状态自动机，该有穷状态自动机的每一个状态转换过程都有一个相应的概率，该概率表示自动机采用这一状态转换的可能性。

6.1 马尔可夫模型

◆马尔可夫链可以表示成状态图（转移弧上有概率的非确定的有穷状态自动机）

– 零概率的转移弧省略。

– 每个节点上所有**发出**弧的概率之和等于1。



X_{m+1}的状态

X_m的状态

$$\begin{array}{c|ccccc} & a_1 & a_2 & \cdots & a_j & \cdots \\ \hline a_1 & p_{11} & p_{12} & \cdots & p_{1j} & \cdots \\ a_2 & p_{21} & p_{22} & \cdots & p_{1j} & \cdots \\ \vdots & \vdots & \vdots & & \vdots & \\ a_i & p_{i1} & p_{i2} & \cdots & p_{ii} & \cdots \end{array}$$

6.1 马尔可夫模型

状态序列 $S_1, \ldots, S_T$ 的概率：

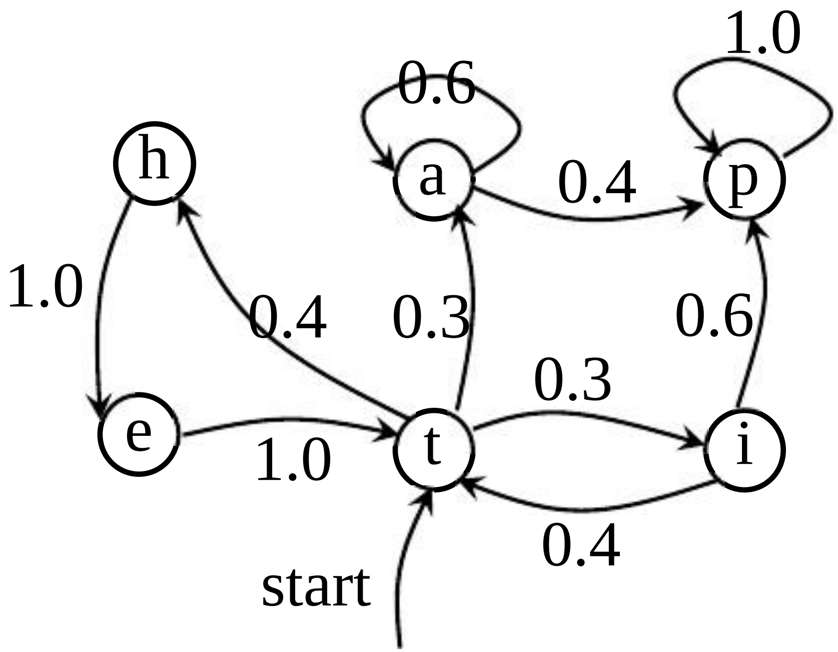
$$p(S_1,\cdots,S_T) = p(S_1)\times p(S_2 \mid S_1)\times p(S_3 \mid S_1,S_2)\times\cdots\times p(S_T \mid S_1,\cdots,S_{T-1})$$

$$= p(S_1)\times p(S_2 \mid S_1)\times p(S_3 \mid S_2)\times\cdots\times p(S_T \mid S_{T-1})$$

$$= \pi_{S_1}\prod_{t=1}^{T-1} a_{S_t S_{t+1}} \qquad \ldots (6.5)$$

其中，$\pi_i = p(q_1 = S_i)$，为初始状态的概率。

6.1 马尔可夫模型



$$p(\mathrm{t}, \mathrm{i}, \mathrm{p}) = p(S_1 = \mathrm{t}) \times p(S_2 = \mathrm{i} | S_1 = \mathrm{t}) \times p(S_3 = \mathrm{p} | S_2 = \mathrm{i})$$

$$= 1.0 \times 0.3 \times 0.6$$

$$= 0.18$$

6.2 隐马尔可夫模型

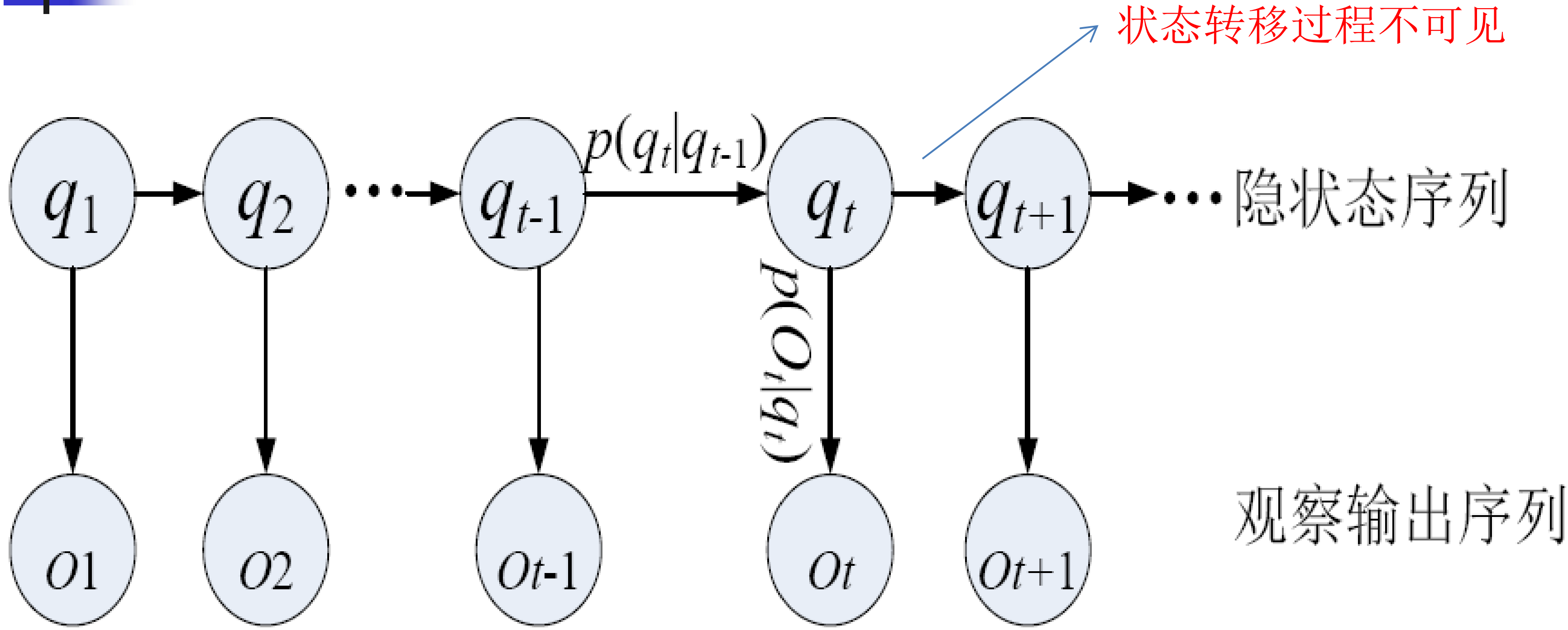
6.2 隐马尔可夫模型

◆隐马尔可夫模型(**Hidden Markov Model, HMM**)

描写：该模型是一个双重随机过程，我们不知道具体的状态序列（隐蔽的），只知道状态转移的概率，而可观察事件的随机过程是隐蔽状态转换过程的随机函数。

注意：马尔科夫模型和隐马尔科夫模型都是有向图

6.2 隐马尔可夫模型

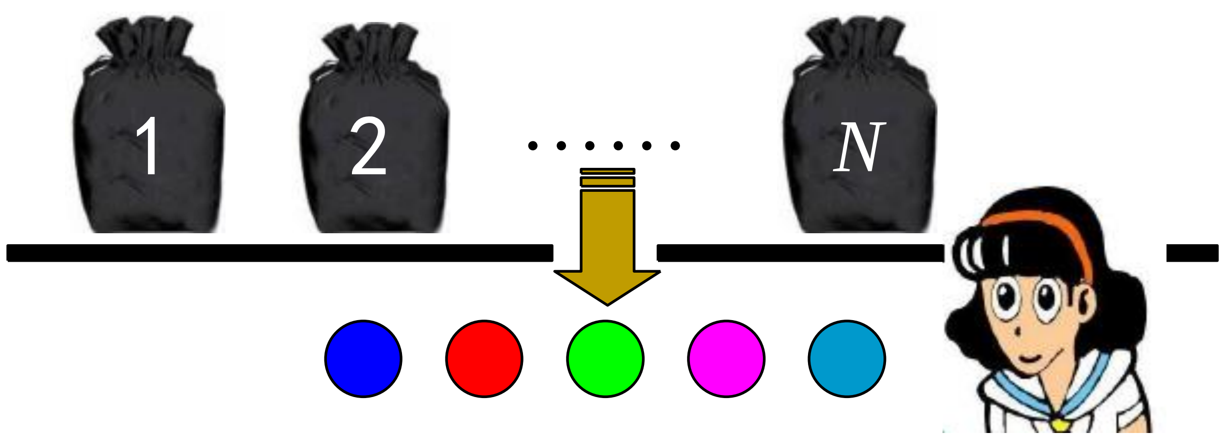


HMM 图解

6.2 隐马尔可夫模型

例如：N 个袋子，每个袋子中有 M 种不同颜色的球。一实验员根据某一概率分布选择一个袋子(对应HMM中的一个状态)，然后根据袋子中不同颜色球的概率分布随机取出一个球，并报告该球的颜色（球的颜色对应于 HMM 中的观察输出）。

对局外人：可观察的过程是不同颜色球的序列，而袋子的序列是不可观察的。



6.2 隐马尔可夫模型

◆**HMM** 的组成

1. 模型中的状态数为 N **(**袋子的数量**)**

2. 从每一个状态可能输出的不同的符号数为M **(**不同颜色球的数目**)**

6.2 隐马尔可夫模型

3. 状态转移概率矩阵 $A = a_{ij}$，a_{ij}为实验员从一只袋子(状态 S_i) 转向另一只袋子(状态 S_j) 取球的概率。其中,

$$\begin{cases} a_{ij} = p(q_{t+1} = S_j \mid q_t = S_i), \quad 1 \le i, j \le N \\ a_{ij} \ge 0 \\ \sum_{j=1}^{N} a_{ij} = 1 \end{cases} \qquad \ldots (6.6)$$

6.2 隐马尔可夫模型

4. 从状态 S_j 观察到某一特定符号 v_k的概率分布矩阵为：

$B=b_j(k)$

其中，$b_j(k)$为 实验员从第 j 个袋子中取出第 k 种颜色的球的概率,也称发射概率。那么，

$$\begin{cases} b_j(k)=p(O_t=v_k \mid q_t=S_j), \quad 1\le j\le N, \quad 1\le k\le M \\ b_j(k)\ge 0 \\ \sum_{k=1}^{M} b_j(k)=1 \end{cases} \quad \ldots (6.7)$$

6.2 隐马尔可夫模型

5. 初始状态的概率分布为：$\pi = \pi_i$，其中，

$$\begin{cases} \pi_i = p(q_1 = S_i), \quad 1 \le i \le N \\ \pi_i \ge 0 \\ \sum_{i=1}^{N} \pi_i = 1 \end{cases} \qquad \ldots (6.8)$$

转移概率　　发射概率　　初始状态

为了方便，一般将 HMM 记为：$\mu = (A, B, \pi)$

或者 $\mu = (S, O, A, B, \pi)$ 用以指出模型的参数集合。

6.2 隐马尔可夫模型

◆给定HMM$\boxed{\text{求观察序列}}$

给定模型 $\mu=(A,B,\pi)$，产生观察序列 $O=O_1O_2\ldots O_T$：

(1) 令 $t=1$;

(2) 根据初始状态分布 $\pi=\pi_i$ 选择初始状态 $q_1=S_i$;

(3) 根据状态 S_i 的输出概率分布 $b_i(k)$, 输出 $O_t=v_k$;

(4) 根据状态转移概率 a_{ij}，转移到新状态 $q_{t+1}=S_j$;

(5) $t=t+1$, 如果 $t<T$, 重复步骤 (3) (4), 否则结束。

6.2 隐马尔可夫模型

◆三个问题：

(1) 在给定模型$\mu=(A, B, \pi)$ 和观察序列 $O=O_1O_2\dots O_T$ 的情况下，怎样快速计算概率 $p(O|\mu)$？

如对于丢硬币(假定每个硬币均不相同, 其序号为状态)测试，上述问题对应：

给定HMM模型，观察结果（硬币的正反面）为O={H，T，H}的概率是多少？

6.2 隐马尔可夫模型

◆三个问题：

(2) 在给定模型 $\mu=(A, B, \pi)$ 和观察序列 $O=O_1O_2\ldots O_T$ 的情况下，如何选择在一定意义下"最优"的状态序列 $Q=q_1q_2\ldots q_T$，使得该状态序列"最好地解释"观察序列？

如对于丢硬币(假定其序号为内部状态)测试，上述问题对应：

若给定观察结果O={H，T，H}，那么最可能的状态序列（硬币序号）是什么？

6.2 隐马尔可夫模型

◆三个问题：

(3) 给定一个观察序列 $O = O_1 O_2 \ldots O_T$，如何根据最大似然估计来求模型的参数值？即如何调节模型的参数，使得 $p(O|\mu)$ 最大？

如对于丢硬币(假定每个硬币均不相同)测试，上述问题对应：

A、B、π未知的情况下，如何根据观察结果 O 得到它们？

6.3 前向算法

6.3 前向算法

◆ 求解问题1：

给定模型 $\mu=(A, B, \pi)$ 和观察序列 $O=O_1O_2 \ldots O_T$，快速计算 $p(O|\mu)$：

对于给定的状态序列 $Q = q_1 q_2 \ldots q_T, p(O|\mu) = ?$

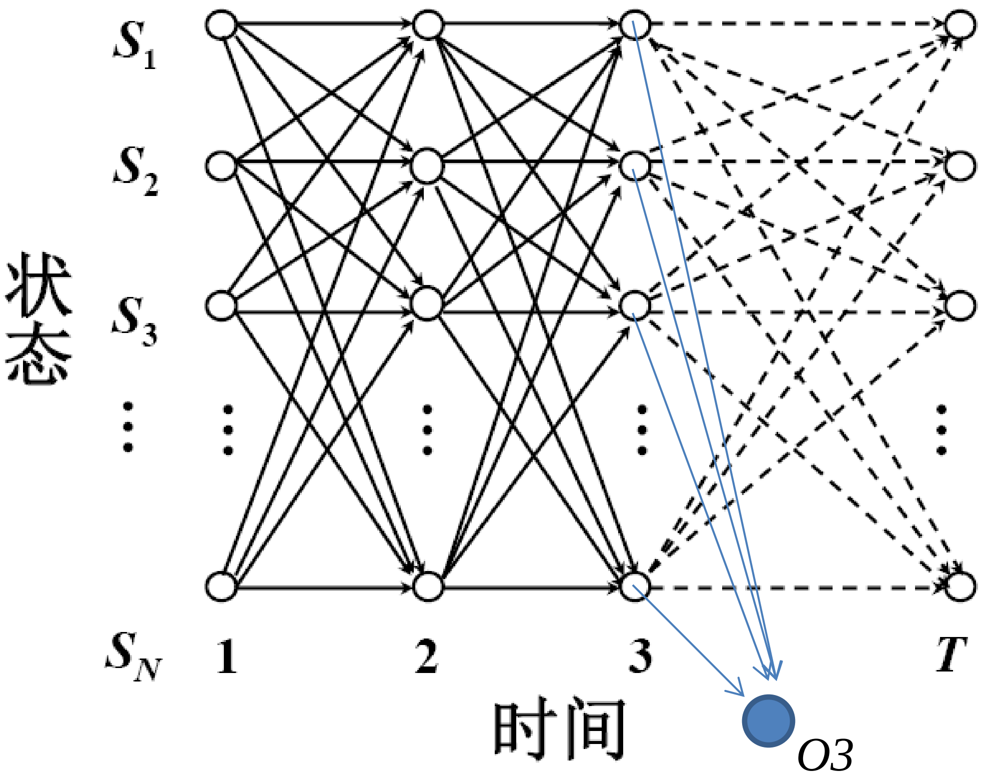
$$p(O|\mu) = \sum_Q p(O,Q|\mu) = \sum_Q \boxed{p(Q|\mu)} \times \boxed{p(O|Q,\mu)} \qquad \ldots (6.9)$$

$$p(Q|\mu) = \pi_{q_1} \times a_{q_1 q_2} \times a_{q_2 q_3} \times \cdots \times a_{q_{t-1} q_T}$$ 转移概率

$$p(O|Q,\mu) = b_{q_1}(O_1) \times b_{q_2}(O_2) \times \cdots \times b_{q_T}(O_T)$$ 发射概率

6.3 前向算法

对每个Oi，需要考虑所有可能路径下的概率 累加



- **困难：**
如果模型μ有 N 个不同的状态，时间长度为 T，那么有 N^T 个可能的状态序列，搜索路径成指数级组合爆炸。

6.3 前向算法

- <u>解决办法</u>：动态规划
  前向算法(The forward procedure)
- <u>基本思想</u>：定义前向变量（前向概率）$\alpha_t(i)$:

$$\alpha_t(i) = p(O_1 O_2 \cdots O_t, q_t = S_i \mid \mu) \qquad \ldots(6.12)$$

$\alpha_t(i)$: 当t时刻的状态为S_i时，且前面时刻观测到$O_1, O_2, \ldots, O_t$的概率

前向概率存储了“从初始到t时刻i状态每个子序列(状态路径)的累积概率”

6.3 前向算法

因为 $p(O|\mu)$ 是在到达状态 q_T 时观察到序列 $O = O_1 O_2 \ldots O_T$ 的概率(所有可能的概率之和):

N为状态总数

$$p(O|\mu) = \sum_{S_i} p(O_1 O_2 \cdots O_T, q_T = S_i | \mu) = \sum_{i=1}^{N} \alpha_T(i) \quad \ldots (6.13)$$

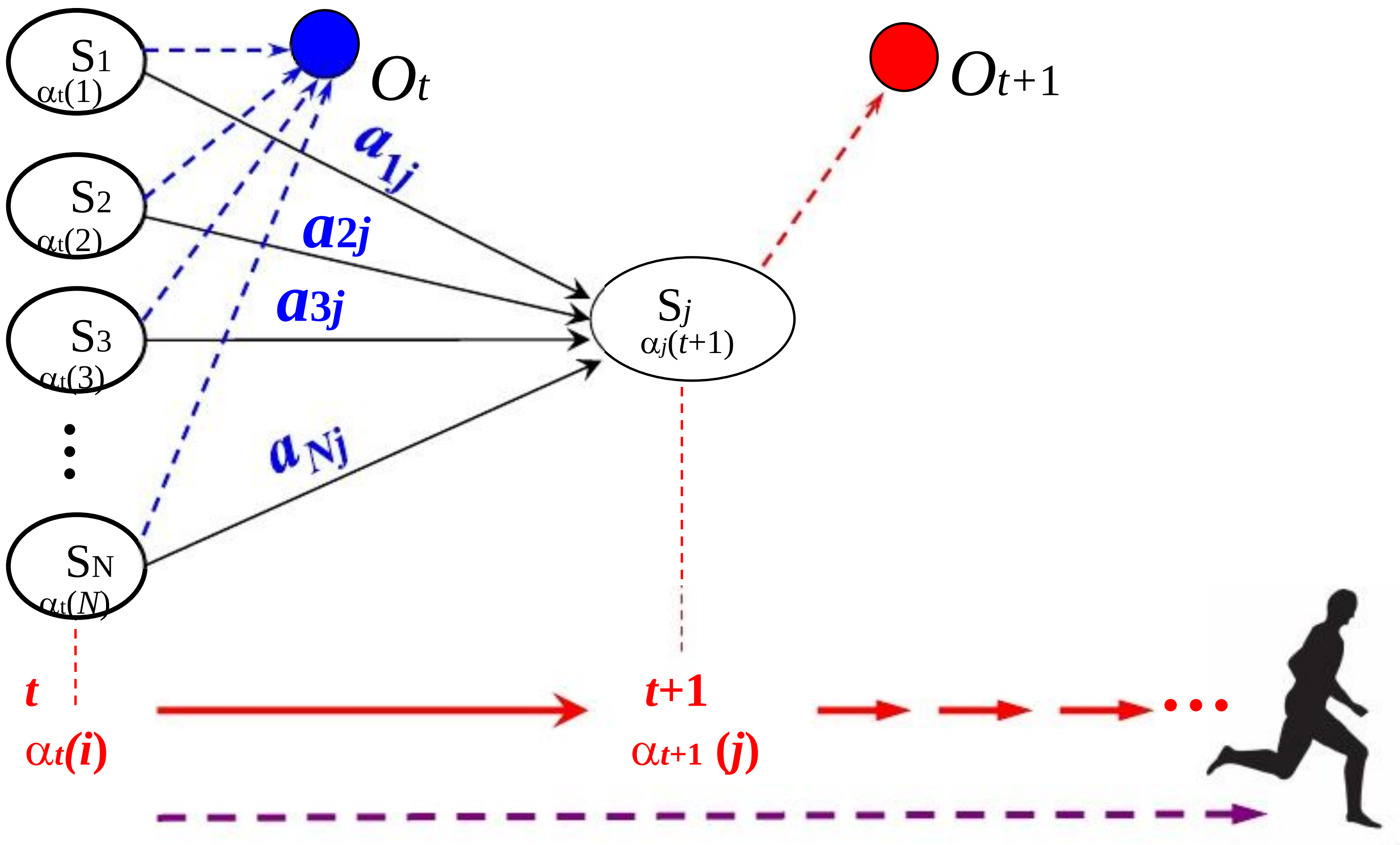
动态规划计算 $\alpha_t(i)$: 在时间 ***t*+1** 的前向变量可以根据时间 ***t*** 的前向变量 $\alpha_t(1),\cdots,\alpha_t(N)$ 的值递推计算:

状态j的发射概率

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] \times b_j(O_{t+1}) \quad \ldots (6.14)$$

上层所有状态到下层特定状态j的连接

6.3 前向算法



S1
αt(1)
S2
αt(2)
S3
αt(3)
SN
αt(N)
Ot
Ot+1
a1j
a2j
a3j
aNj
Sj
αj(t+1)
t
αt(i)
t+1
αt+1 (j)

6.3 前向算法

- 算法6.1：前向算法描述

(1) 初始化： $\alpha_1(i) = \pi_i b_i(O_1)$， $1 \le i \le N$

(2) 循环计算：

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] \times b_j(O_{t+1}), \quad 1 \le t \le T-1$$

(3) 结束，输出：

$$p(O \mid \mu) = \sum_{i=1}^{N} \alpha_T(i)$$

6.4 前向算法–实例分析

观察集合是：　V={红，白}，M=2

状态集合是：　Q={盒子1，盒子2，盒子3}，N=3

球的颜色的观测序列：　O={红，白，红}

初始状态分布为：　Π=(0.2,0.4,0.4)

其它转移概率、发射概率均已知。

（1）首先计算时刻1三个状态的前向变量：时刻1是红色球，

隐藏状态是盒子1的概率为：

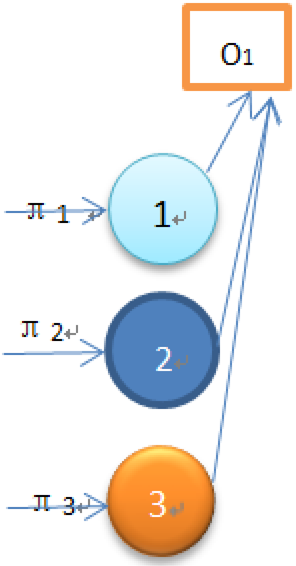
$$\alpha_1(1)=\pi_1 b_1(o_1)=0.2\times 0.5=0.1$$

隐藏状态是盒子2的概率为：

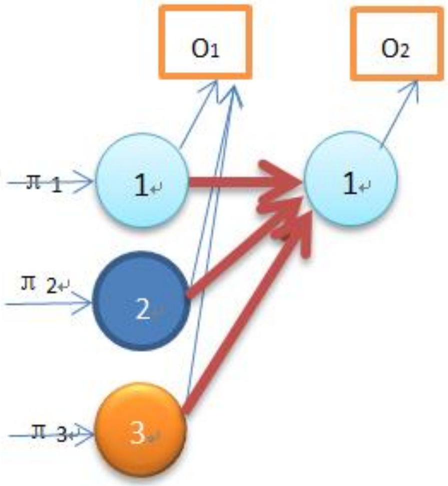
$$\alpha_1(2)=\pi_2 b_2(o_1)=0.4\times 0.4=0.16$$

隐藏状态是盒子3的概率为：

$$\alpha_1(3)=\pi_3 b_3(o_1)=0.4\times 0.7=0.28$$



6.4 前向算法-实例分析



球的颜色的观测序列： O={红，白，红}

（2）开始递推，时刻2三个状态的前向概率：时刻2是白色球隐藏状态是盒子1的概率为：

$$\alpha_2(1) = \left[\sum_{i=1}^{3} \alpha_1(i) a_{i1}\right] b_1(o_2) = [0.1 * 0.5 + 0.16 * 0.3 + 0.28 * 0.2] \times 0.5 = 0.077$$

隐藏状态是盒子2的概率为：

$$\alpha_2(2) = \left[\sum_{i=1}^{3} \alpha_1(i) a_{i2}\right] b_2(o_2) = [0.1 * 0.2 + 0.16 * 0.5 + 0.28 * 0.3] \times 0.6 = 0.1104$$

隐藏状态是盒子3的概率为：

$$\alpha_2(3) = \left[\sum_{i=1}^{3} \alpha_1(i) a_{i3}\right] b_3(o_2) = [0.1 * 0.3 + 0.16 * 0.2 + 0.28 * 0.5] \times 0.3 = 0.0606$$

6.4 后向算法

6.4 后向算法

- 后向算法 (The backward procedure)

后向变量 $\beta_t(i)$：是在给定了模型$\mu=(A,B,\pi)$和时间 t 时状态为 S_i 的条件下，模型输出观察序$O_{t+1}O_{t+2}\ldots\ldots O_T$的概率：

$$\beta_t(i)=p(O_{t+1}O_{t+2}\cdots O_T \mid q_t=S_i,\mu) \qquad \ldots (6.15)$$

后向变量(概率)存储了“从t时刻的状态i开始每个子序列(状态路径)的累积概率”

6.4 后向算法

与前向变量一样，运用动态规划计算后向变量：

(1)当t=T时，$\beta_T(i)=1,\quad 1\le i\le N$

(2)在时间 t=T-1时

$$\beta_t(i)=\sum_{j=1}^{N}a_{ij}b_j(O_{t+1})\times\beta_{t+1}(j)$$

由 β_{t+1}倒推β_t

归纳顺序： $\beta_T(x),\beta_{T-1}(x),\cdots,\beta_1(x)$

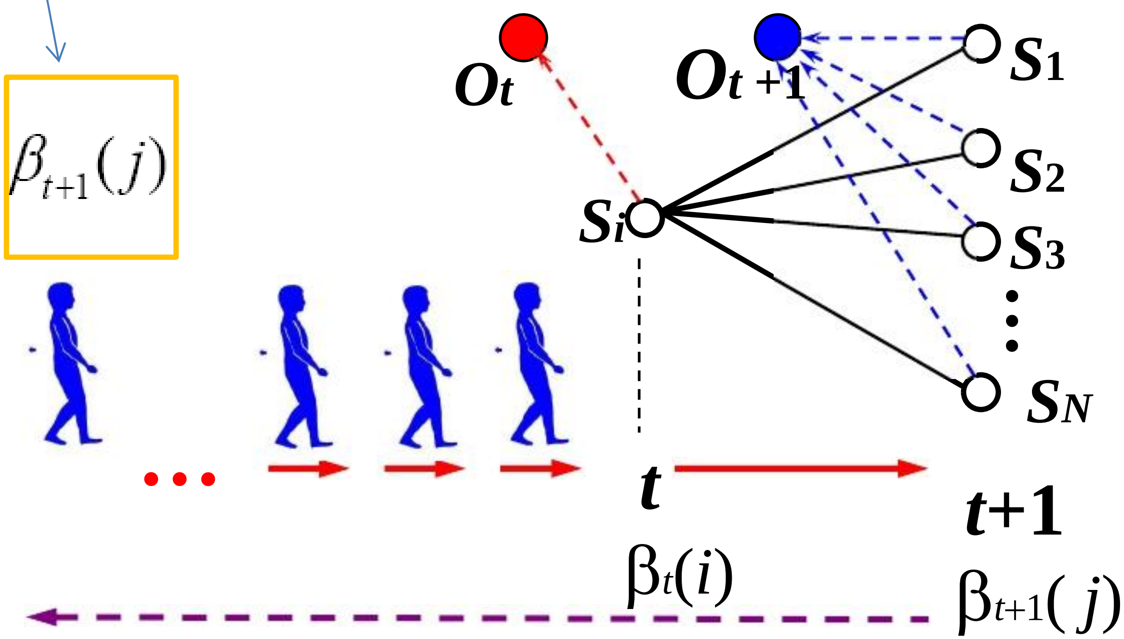
6.4 后向算法

算法图解：

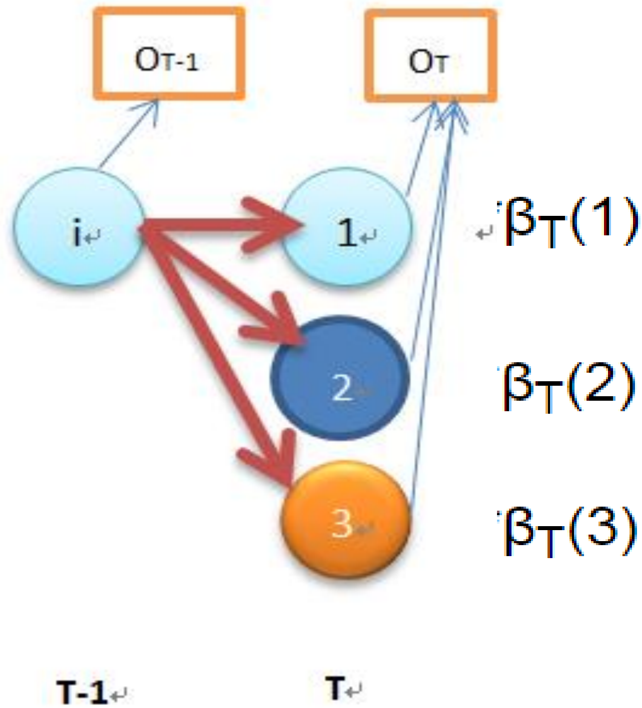
(1) 从时刻 t 到 t+1，模型由状态 S_i 转移到状态 S_j，并从 S_j 输出 O_{t+1}；

(2) 在时间 t+1，状态为 S_j 的条件下，模型输出观察序列 $O_{t+2}O_{t+3}\cdots O_T$ 。

$$\beta_t(i) = \boxed{\sum_{j=1}^{N} a_{ij} b_j(O_{t+1})} \times \boxed{\beta_{t+1}(j)}$$



6.4 后向算法



$$\beta_{T-1}(i)=P(O_T|q_{T-1}=s_i,\mu) =$$

$$a_{i1}{*}b_1(O_T){*}\beta_T(1) + a_{i2}{*}b_2(O_T){*}\beta_T(2) + a_{i3}{*}b_3(O_T){*}\beta_T(3)$$

6.4 后向算法

- 算法6.2：后向算法描述

(1) 初始化：$\beta_T(i)=1,\quad 1\le i\le N$

从最后时刻T开始

(2) 循环计算：

$$\beta_t(i)=\sum_{j=1}^{N}a_{ij}b_j(O_{t+1})\times\beta_{t+1}(j),\quad T-1\ge t\ge 1,\quad 1\le i\le N$$

(3) 输出结果：$p(O\mid\mu)=\sum_{i=1}^{N}\beta_1(i)\times\pi_i\times b_i(O_1)$

6.5 Viterbi搜索算法

6.5 Viterbi 搜索算法

◆问题2—如何发现“最优”状态序列
能够“最好地解释”观察序列

一种解释：在给定模型μ 和观察序列O的条件下求概率最大的状态序列：

$$\hat{Q} = \underset{Q}{\mathrm{argmax}}\ p(Q|O,\mu) \quad \ldots (6.21)$$

Viterbi 算法: 利用动态规划求解概率最大的路径，一条路径对应一个状态序列。

6.5 Viterbi 搜索算法

原理：从t=1时刻开始，不断向后递推到下一个状态路径的最大概率，直至最后到达最终的最优路径，然后依据终点回溯到起始点，这样就能得到最优路径。

定义：**Viterbi 变量**$\delta_t(i)$是在时间 t 时，模型沿着某一条路径到达 S_i，且输出观察序列 $O=O_1O_2\ldots O_t$的最大概率为：

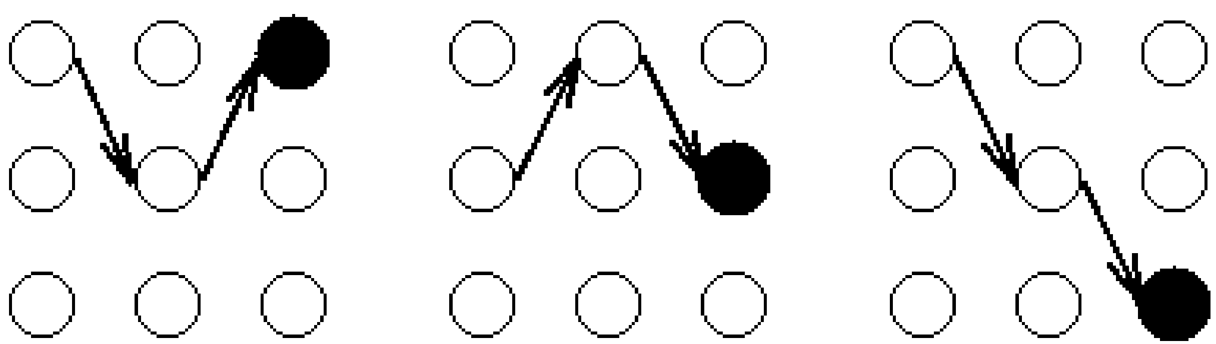
$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} p(q_1,q_2,\cdots,q_t=S_i,O_1O_2\cdots O_t \mid \mu) \quad \ldots (6.22)$$

6.5 Viterbi 搜索算法

$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} p(\underbrace{q_1,q_2,\cdots,q_t}_{} = S_i, O_1O_2\cdots O_t \mid \mu) \quad \ldots (6.22)$$

变量δt(*i*)存储了一条到达中间状态*Si*时的局部最优路径，且通过该路径到达状态*Si*的概率为δt(*i*)。

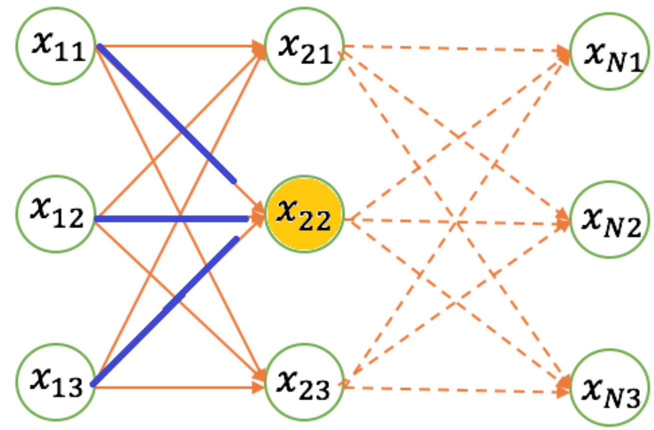
通常时刻t时，每个状态*Si*都有一个到达该状态的最可能路径，如第3时刻，每个状态*Si* 的最有路径：



6.5 Viterbi 搜索算法

递归计算： $\delta_{t+1}(i) = \max_{j}[\delta_t(j)\cdot a_{ji}]\cdot b_i(O_{t+1})$

三选一



t时刻　　t+1时刻

- **算法6.3：Viterbi 算法描述**

(1) 初始化： $\delta_1(i) = \pi_i b_i(O_1)$， $1 \le i \le N$

概率最大的路径变量： $\psi_1(i) = 0$

(2) 递推计算：

$$\delta_t(j) = \max_{1\le i\le N}[\delta_{t-1}(i)\cdot a_{ij}]\cdot b_j(O_t),\quad 2\le t\le T,\quad 1\le j\le N$$

$$\psi_t(j) = \arg\max_{1\le i\le N}[\delta_{t-1}(i)\cdot a_{ij}]\cdot b_j(O_t),\ 2\le t\le T,\ 1\le i\le N$$

6.5 Viterbi 搜索算法

(3) 结束：

T时刻，所有状态中δ最大的那个状态i

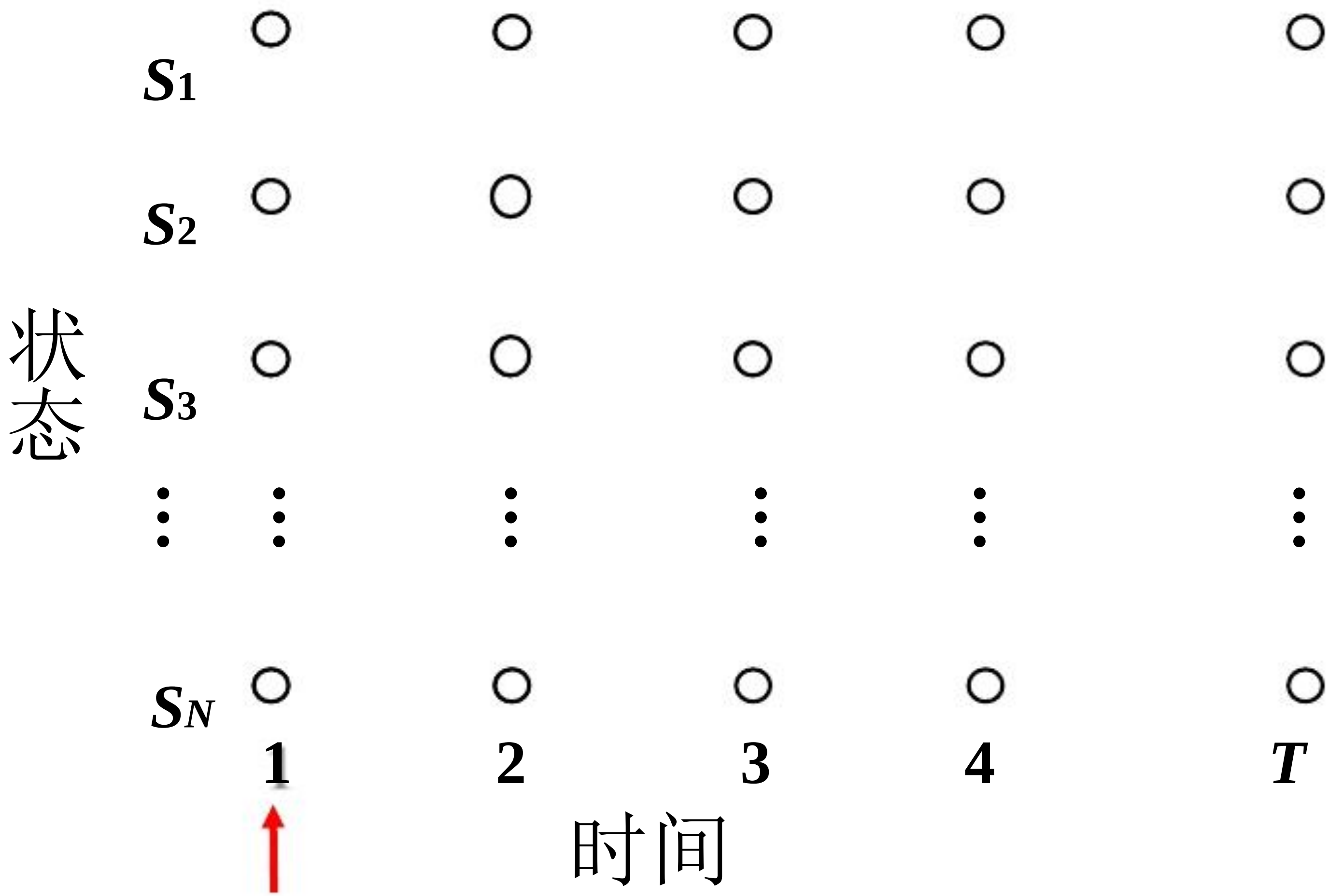
$$\widehat{Q}_T = \arg\max_{1 \le i \le N} [\delta_T(i)], \qquad \widehat{p}(\widehat{Q}_T) = \max_{1 \le i \le N} \delta_T(i)$$

(4) 通过回溯得到路径（状态序列）：

$$\widehat{q}_t = \psi_{t+1}(\widehat{q}_{t+1}), \quad t = T-1,\ T-2,\ \cdots,\ 1$$

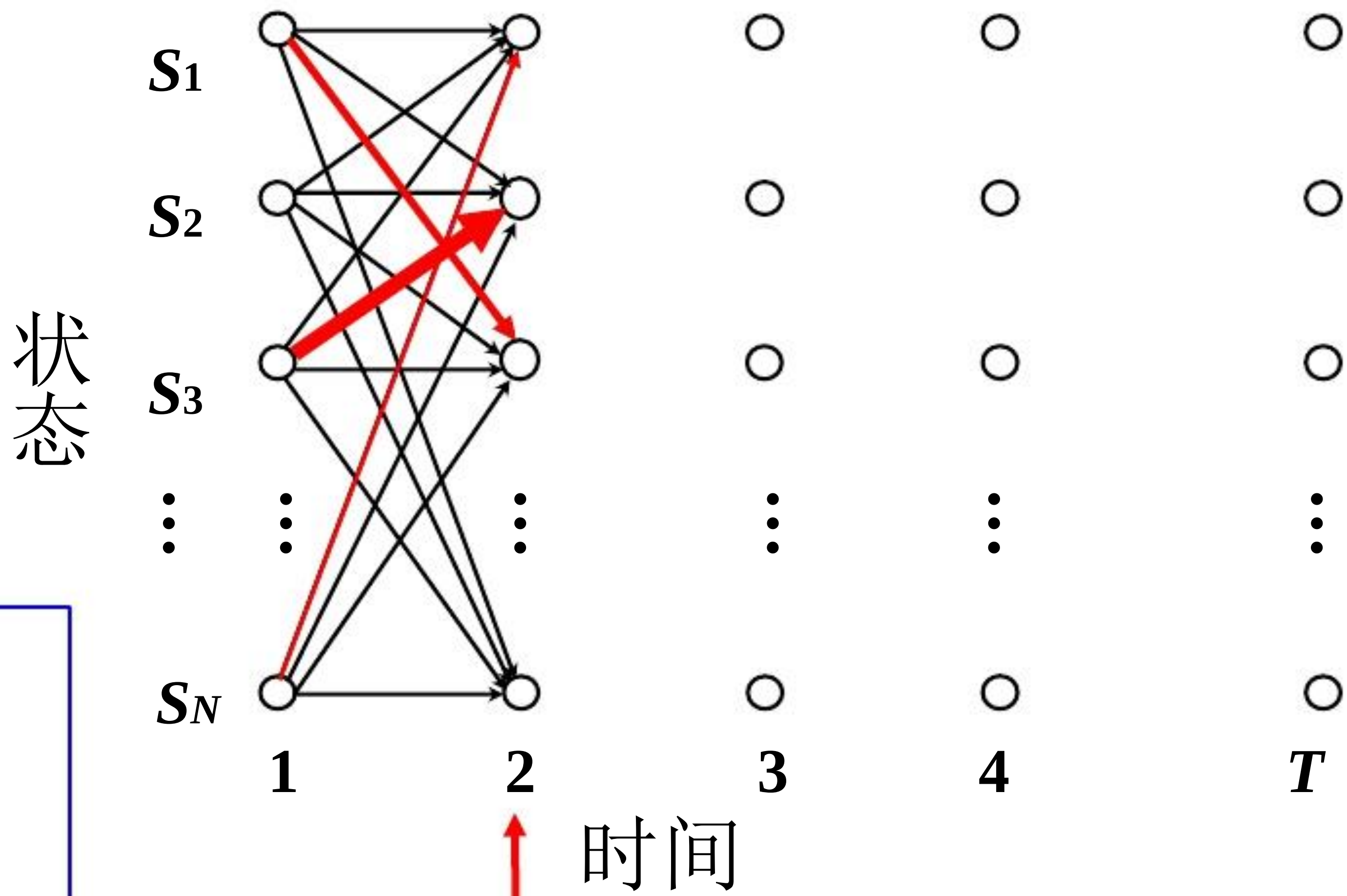
6.5 Viterbi 搜索算法

图解
Viterbi
搜索
过程



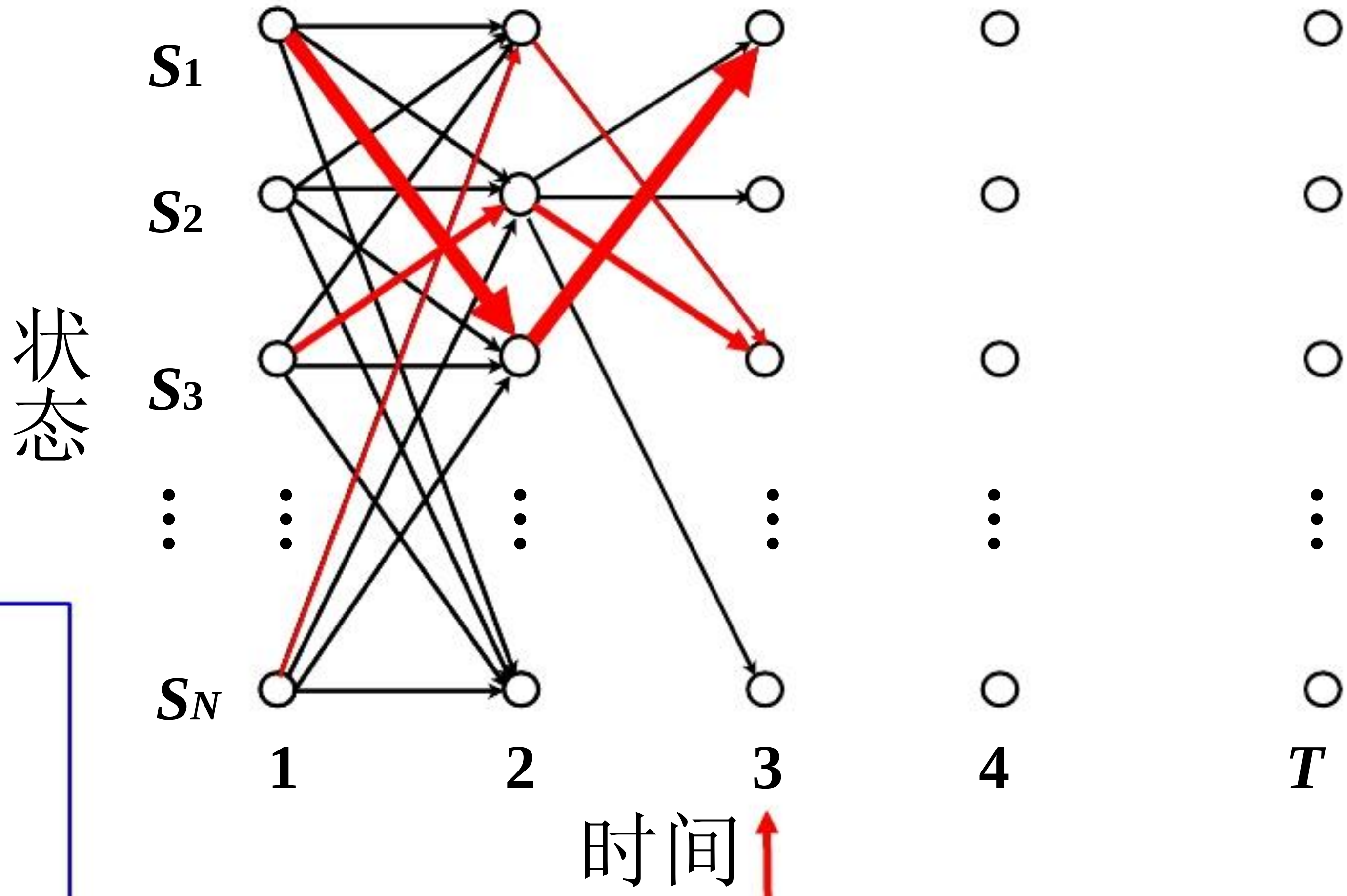
6.5 Viterbi 搜索算法

图解
Viterbi
搜索
过程



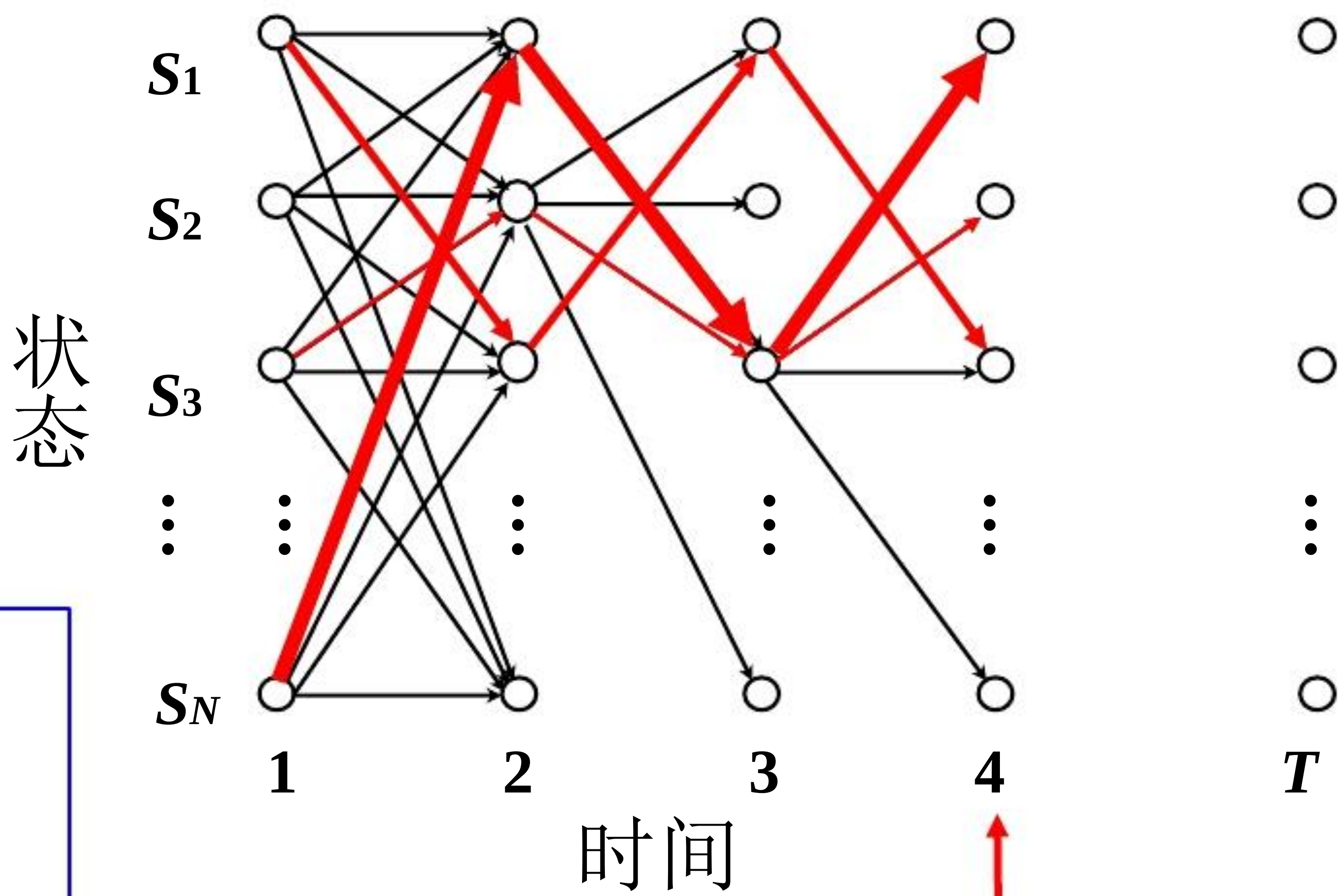
6.5 Viterbi 搜索算法

图解
Viterbi
搜索
过程



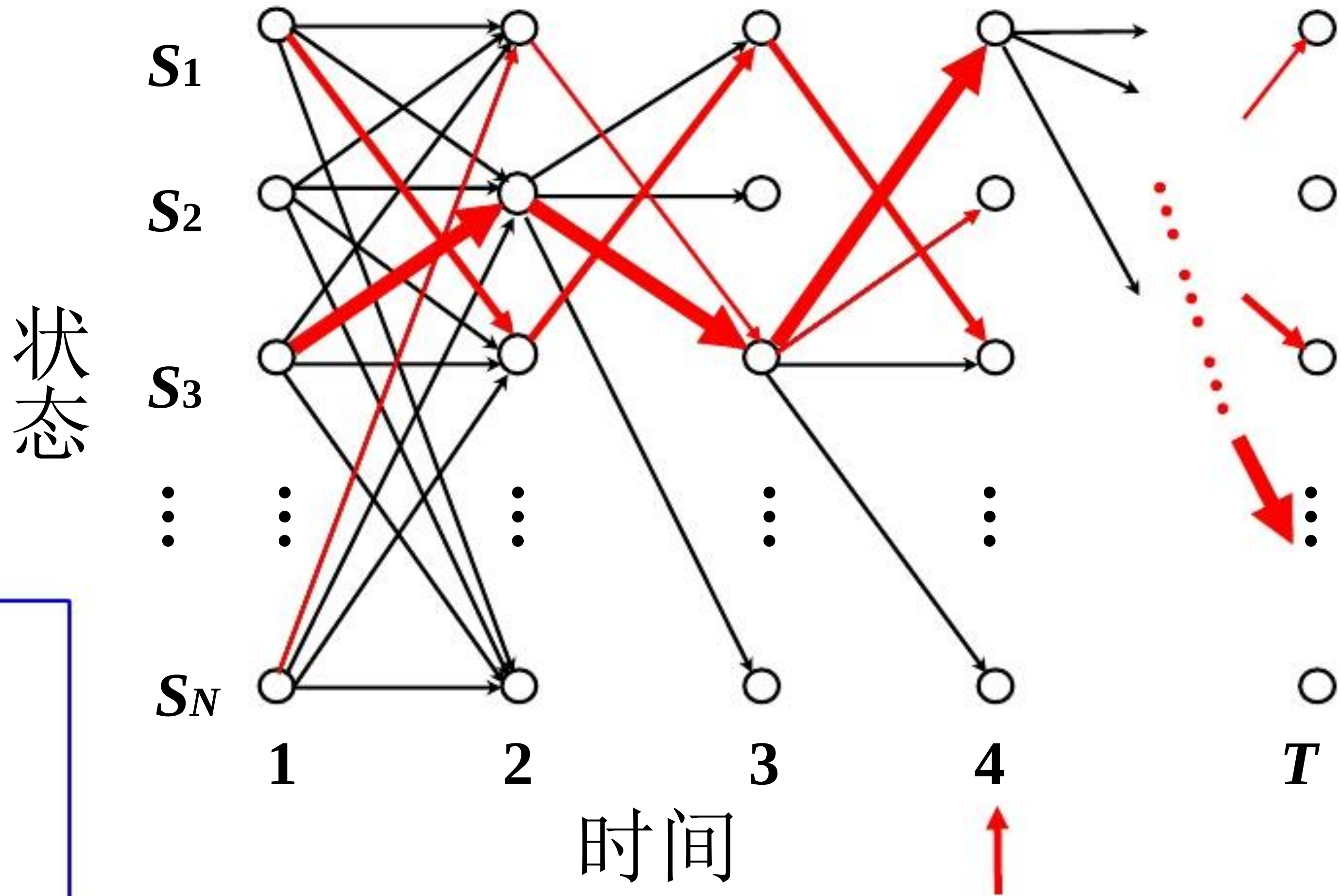
6.5 Viterbi 搜索算法

图解
Viterbi
搜索
过程



6.5 Viterbi 搜索算法

图解
Viterbi
搜索
过程



6.6 参数学习

6.6 参数学习

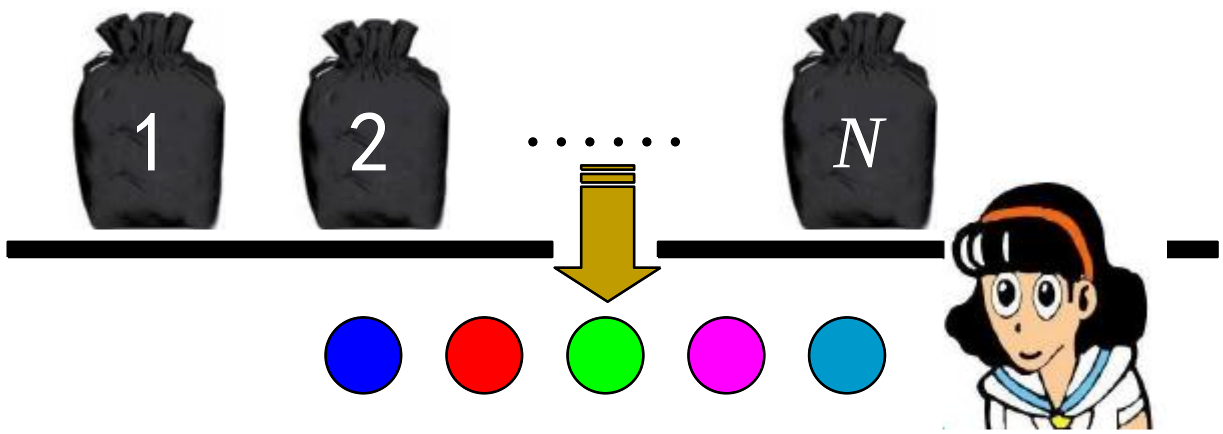
◆问题3－模型参数学习

给定一个观察序列 $O = O_1O_2 \ldots O_T$，如何根据极大似然估计来求模型的参数值？

即估计模型中的π_i, a_{ij}, $b_j(k)$使得观察序列 O 的概率 $p(O|\mu)$ 最大。

6.6 参数学习

如果产生观察序列 O 的状态 $Q = q_1q_2\ldots q_T$已知(即存在状态标注的样本), 可以用极大似然估计来计算 μ 的参数:



这时，实验员从袋子中取球的过程是透明的，可以知道整个过程经历了哪些内部状态改变。

6.6 参数学习

如果产生观察序列 O 的状态 $Q = q_1 q_2 \ldots q_T$ 已知(即存在状态标注的样本), 可以用极大似然估计来计算 μ 的参数:

$$\bar{\pi}_i = \delta(q_1, S_i)$$

$$\bar{a}_{ij} = \frac{Q\text{中从状态}q_i\text{转移到}q_j\text{的次数}}{Q\text{中所有从状态}q_i\text{转移到另一状态(包括}q_j\text{自身)的总数}}$$

$$= \frac{\sum_{t=1}^{T-1} \delta(q_t, S_i) \times \delta(q_{t+1}, S_j)}{\sum_{t=1}^{T-1} \delta(q_t, S_i)} \qquad \ldots (6.24)$$

其中，δ(*x*, *y*) 为<u>克罗奈克(Kronecker)函数</u>， 当 ***x*= *y*** 时, δ(*x*, *y*)=1, 否则δ(*x*, *y*) = 0。

6.6 参数学习

类似地，

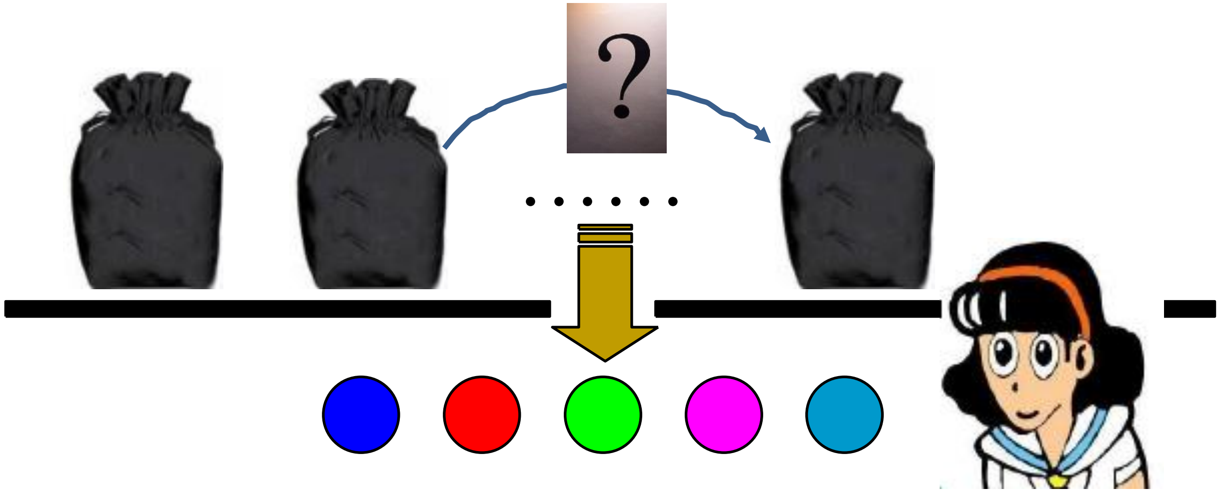
$$\bar{b}_j(k) = \frac{Q\text{中从状态}q_j\text{输出符号}v_k\text{的次数}}{Q\text{到达}q_j\text{的总次数}}$$

$$= \frac{\sum_{t=1}^{T} \delta(q_t, S_j) \times \delta(O_t, v_k)}{\sum_{t=1}^{T} \delta(q_t, S_j)} \quad \ldots (6.25)$$

其中，v_k 是模型输出符号集中的第 k 个符号。

6.6 参数学习

如果不存在标注（状态）的样本



这时，只能观察到取出球的序列，但整个过程经历了哪些内部状态改变就是未知的。

6.6 参数学习

如果不存在大量标注的样本。

- 期望值最大化算法 (Expectation-Maximization, EM)

基本思想：

（1）初始化时随机地给模型的参数赋值，得到模型μ_0

（2）根据μ_0求的模型中隐变量的期望值。

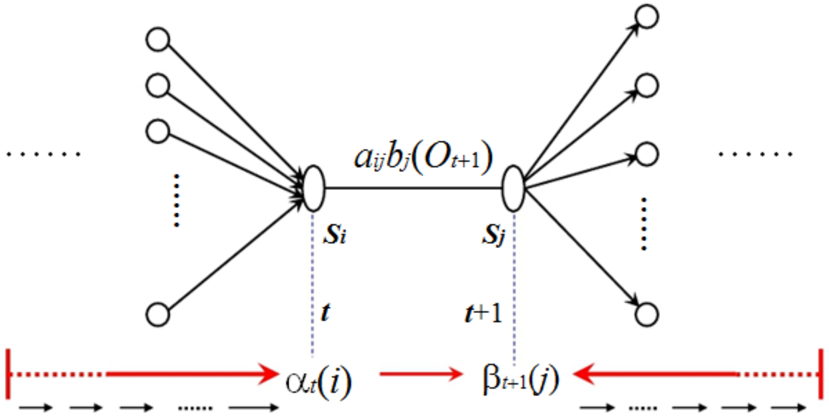
——如根据μ_0 求得到从某一状态转移到另一状态的期望次数

(3) 然后以期望次数代替公式中的实际次数，由此更新得到新的模型μ_1。

循环这一过程，直到参数收敛于最大似然估计值。

6.6 参数学习

给定模型 μ和观察序列 $O=O_1O_2\ldots O_T$，前一时间(t)位于状态 S_i，后一时间($t+1$) 位于状态 S_j 的概率：



$$\xi_t(i,j)=p(q_t=S_i,q_{t+1}=S_j\mid O,\mu)=\frac{p(q_t=S_i,q_{t+1}=S_j,O\mid\mu)}{p(O\mid\mu)}$$

$$=\frac{\alpha_t(i)\times a_{ij}b_j(O_{t+1})\times\beta_{t+1}(j)}{p(O\mid\mu)}$$

计算中要用到初始模型参数

$$=\frac{\alpha_t(i)\times a_{ij}b_j(O_{t+1})\times\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)\times a_{ij}b_j(O_{t+1})\times\beta_{t+1}(j)}\qquad\ldots(6.26)$$

6.6 参数学习

那么，给定模型 μ 和观察序列 $O = O_1 O_2 \ldots\ldots O_T$，在时间 t 位于状态 S_i 的概率为：

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j) \qquad \ldots (6.27)$$

由此，模型 μ 的参数可由下面的公式重新估计：

(1) q_1 为 S_i 的概率：

$$\pi_i = \gamma_1(i) \qquad \ldots (6.28)$$

6.6 参数学习

(2)

$$\bar{a}_{ij} = \frac{Q\text{中从状态}q_i\text{ 转移到 }q_j\text{的期望次数}}{Q\text{中所有从状态 }q_i\text{ 转移到下一状态(包括 }q_j\text{自身)的期望次数}}$$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \cdots (6.29)$$

假定状态序列中1....T时刻，从qi到qj的转移次数t=1+0+1+1+0+1+......

期望出现次数=$\sum 1*\xi_t(i,j)+0*\xi_t(i,j)+\cdots$

6.6 参数学习

(3) $$\bar{b}_j(k) = \frac{Q\text{中从状态}q_j\text{输出符号}v_k\text{的期望次数}}{Q\text{到达}q_j\text{的期望次数}}$$

$$= \frac{\sum_{t=1}^{T} \gamma_t(j) \times \delta(O_t, v_k)}{\sum_{t=1}^{T} \gamma_t(j)} \quad \ldots (6.30)$$

6.6 参数学习

- **<u>算法6.4</u>：Baum-Welch 算法(前向后向算法)描述：**

(1) 初始化：随机地给 $\pi_i, a_{ij}, b_j(k)$ 赋值，

使得
$$\begin{cases} \sum_{i=1}^{N} \pi_i = 1 & \\ \sum_{j=1}^{N} a_{ij} = 1 & 1 \le i \le N \\ \sum_{k=1}^{M} b_i(k) = 1 & 1 \le i \le N \end{cases} \qquad \ldots (6.31)$$

由此得到模型μ_0，令 $i = 0$。

6.6 参数学习

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

$$\gamma_t(i)=\sum_{j=1}^{N}\xi_t(i,j)$$

(2) 执行 EM 算法:

E-步: 由模型 μ_i 根据公式 (6.26) 和 (6.27) 计算期望值 $\xi_t(i,j)$和 $\gamma_t(i)$。

M-步: 用E-步中所得到的期望值，根据公式 (6.28-6.30) 重新估计π_i, a_{ij}, $b_j(k)$ 得到模型μ_{i+1}。

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

$$\bar{b}_j(k)=\frac{\sum_{t=1}^{T}\gamma_t(j)\times\delta(O_t,v_k)}{\sum_{t=1}^{T}\gamma_t(j)}$$

6.6 参数学习

(2) 执行 EM 算法:

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

$$\gamma_t(i)=\sum_{j=1}^{N}\xi_t(i,j)$$

E-步: 由模型 μ_i 根据公式 (6.26) 和 (6.27) 计算期望值 $\xi_t(i,j)$和 $\gamma_t(i)$。

M-步: 用E-步中所得到的期望值，根据公式 (6.28-6.30) 重新估计 π_i, a_{ij}, $b_j(k)$ 得到模型μ_{i+1}。

循环: $i = i+1$，重复执行 E-步和M-步，直至 π_i, a_{ij}, $b_j(k)$的值收敛：$|\log p(O|\mu_{i+1})-\log p(O|\mu_i)|<\varepsilon$ 。

(3) 结束算法，获得相应的参数。

6.6 参数学习

假设一个HMM的模型的状态集S={1,2,3},观测集V={1,2},π = (0,1,0),转移概率A,发射概率B如下:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0.0193 & 0 & 0.9807 \\ 0.0001 & 0.9999 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0.9858 & 0.0142 \\ 1 & 0 \\ 0.1505 & 0.8495 \end{bmatrix}.$$

使用1000个观测值O = (1,2,1,2,1,2,1,2,1,1......)训练后

$$\pi^{**} = (0.0000, 1.0000, 0.0000)$$

$$A^{**} = \begin{bmatrix} 0.0000 & 1.0000 & 0.0000 \\ 0.0565 & 0.0000 & 0.9435 \\ 0.0000 & 1.0000 & 0.0000 \end{bmatrix}$$

$$B^{**} = \begin{bmatrix} 0.9369 & 0.0631 \\ 1.0000 & 0.0000 \\ 0.1304 & 0.8696 \end{bmatrix}.$$

HMM应用

（1）将状态集Q设为{B,E,M,S}，表示词的开始、结束、中间（begin、end、middle）及字符独立成词（single）；
（2）观测序列即为中文句子。比如，“今天天气不错”；
（3）通过HMM求解得到状态序列“B E B E B E”，则分词结果为“今天/天气/不错”。

中文分词的任务对应于之前问题二（解码）：
对于字符串C{c1,···,cn}，求下列最大条件概率

$$\max P(q_1, q_2, \ldots, q_t \mid c_1, c_2 \ldots c_n)$$

北 N B
京 N E
欢 V B
迎 V M
你 N E

其中，qi表示字符ci对应的状态; 求解上述问题的方法便是Viterbi算法；

6.8 CRFs及其应用

6.8 CRFs及其应用

◆提出

条件随机场(conditional random fields, CRFs)于2001年由 **J. Lafferty** 等人提出，是用于标注和划分序列结构数据的概率化结构模型，在**NLP**和图像处理中得到了广泛应用。

基本思路：给定观察序列 $\boldsymbol{X}$，输出标识序列 $\boldsymbol{Y}$，通过计算 $\boldsymbol{P(Y|X)}$ 求解最优标注序列。

6.8 CRFs及其应用

◆定义

设 $G=(V, E)$ 为一个无向图，V为结点集合，E为无向边的集合，如果以观察序列X为条件，每个随机变量Y_v都满足以下马尔可夫特性：

$$p(Y_v \mid X, Y_w, w \neq v) = p(Y_v \mid X, Y_w, w \sim v)$$

其中，$w \sim v$ 表示两个结点在图中是邻近结点。那么，(X, Y) 为一个条件随机场。

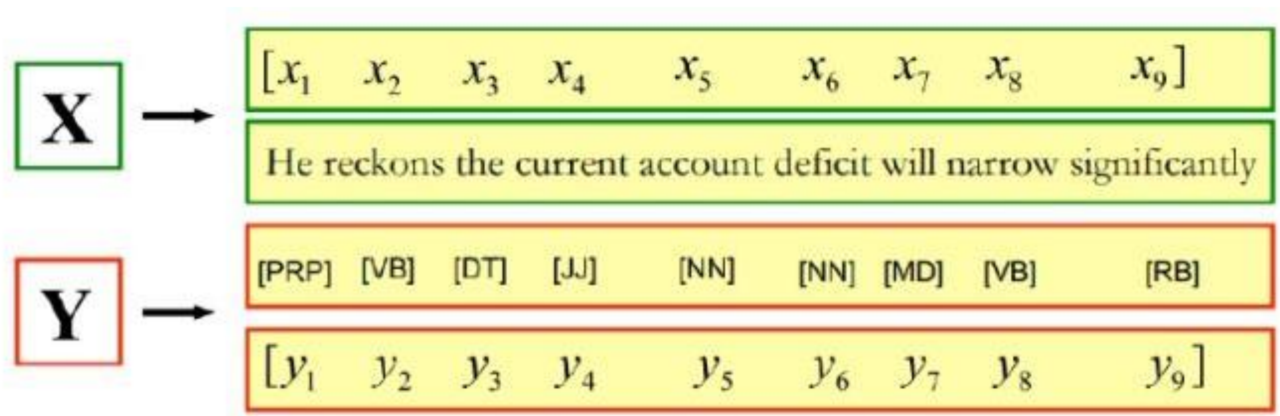
CRF 假设模型中只有 **X** (观测值) 和 **Y** (状态值)。在 CRF 中每一个状态值 yi 只和其相邻的状态值有关，而观测值 x 不具有马尔科夫性质。

6.8 CRFs及其应用

◆理解CRF

随机场是由若干个位置组成的整体，当给每一个位置中按照某种分布随机赋予一个值之后，其全体就叫做随机场。例如，词性标注。

条件随机场即为给定条件X后，由变量Y构成的随机场P(Y | X)。

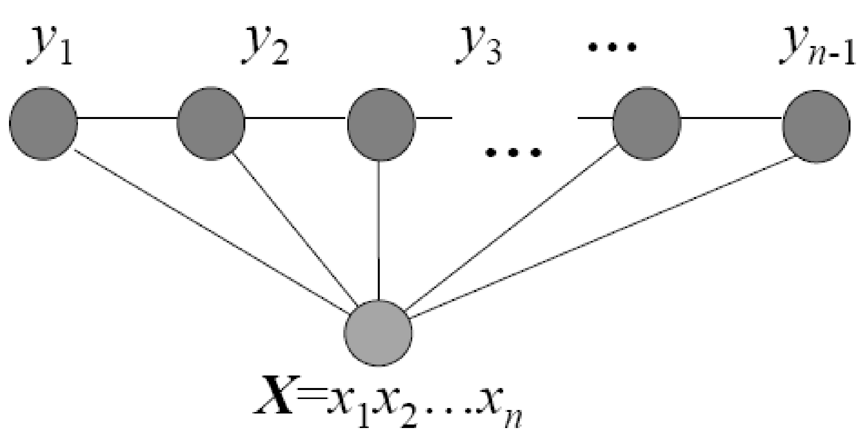


6.8 CRFs及其应用

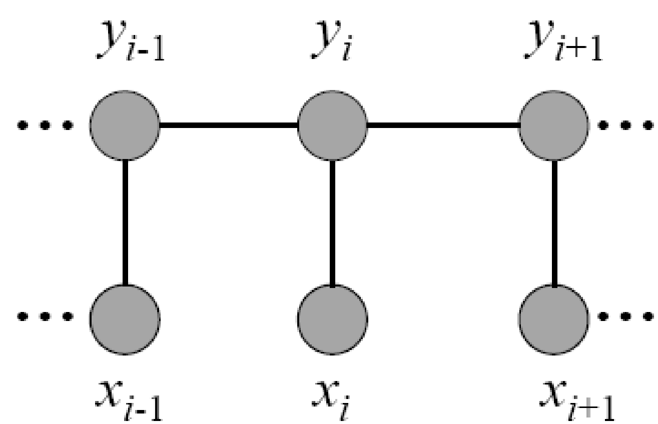
◆线性链条件随机场

假设X和Y有相同的图结构，且满足:

$$P(Y_i|X, Y_1, Y_2, \ldots Y_n) = P(Y_i|X, Y_{i-1}, Y_{i+1})$$

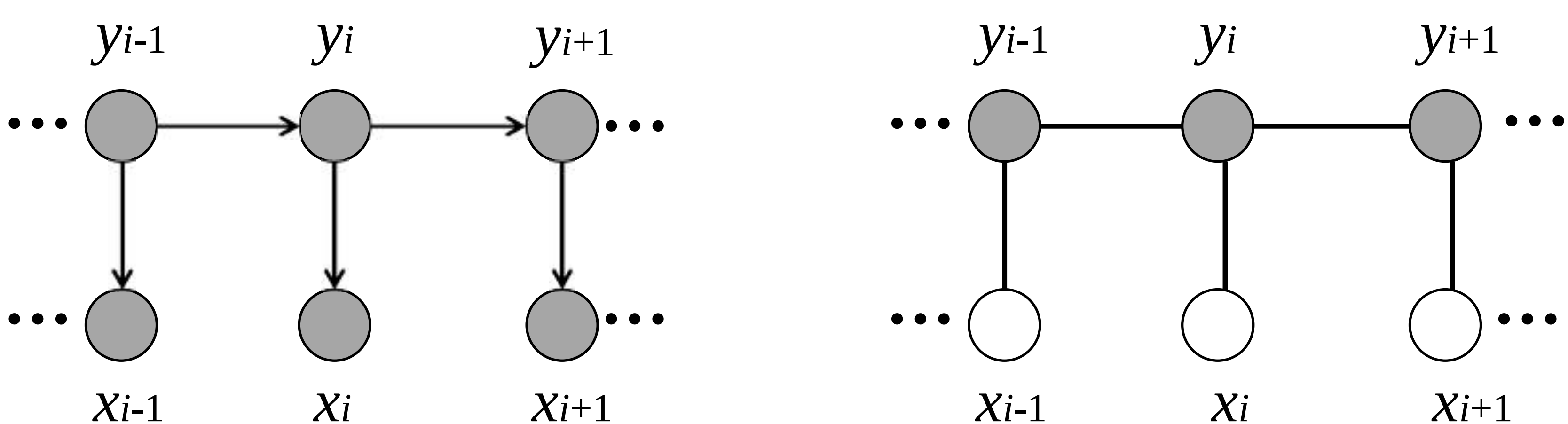


或者:



6.8 CRFs及其应用

HMMs vs. CRFs



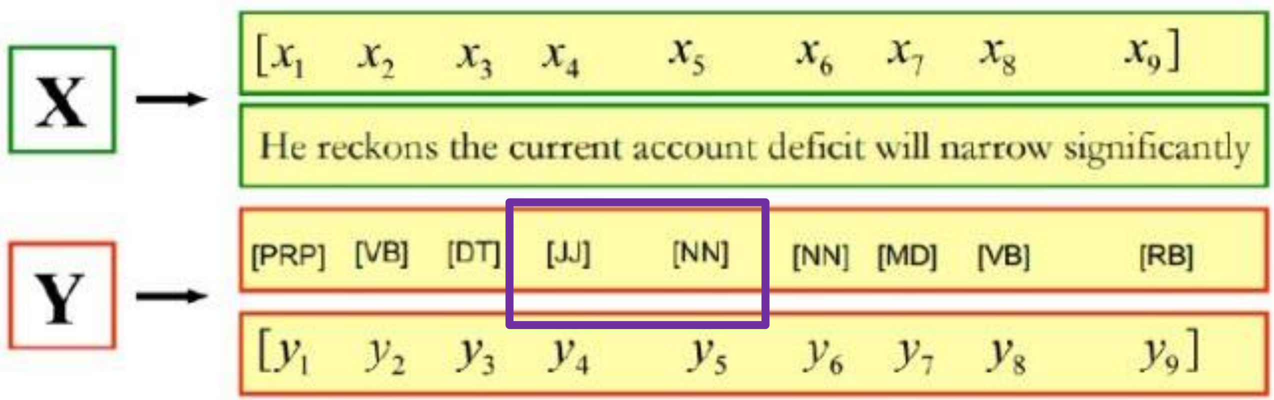
HMMs **CRFs**

一个是有向图，一个是无向图。

CRFs 中的空心节点 $\boldsymbol{x}$ 表示该节点并不是由模型生成的。

6.8 CRFs及其应用

例如，我们想对语句X进行词性标注， 输出词性序列Y：



观察结论：

（1）根据训练语料库，我们知道“He”是个代词，因此根据P(Y|X)可以直接预测x1的词性为prp。（X对Y有影响）

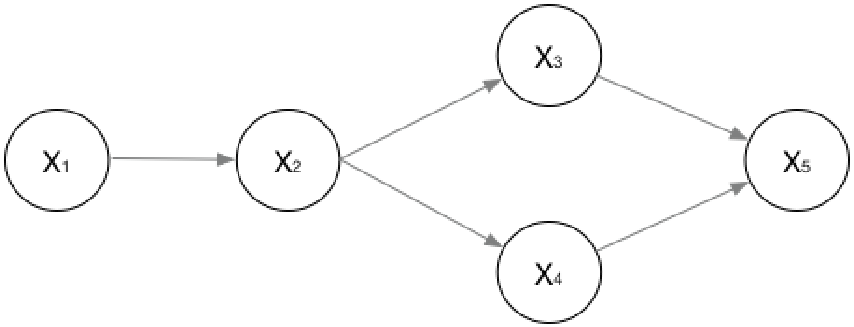
（2）但预测“account”更困难（可能是名词（账户），也可能是动词（导致））。注意yi之间是有顺序性的，预测时，除考虑X与Y之间的关系，以及yi-1与yi之间的关系，则预测准确性将极大增强。

6.8 CRFs及其应用

对于有向图模型，这么求联合概率：

$$P(x_1, \cdots, x_n) = \prod_{i=0} P(x_i | \pi(x_i))$$

如下图的联合概率为：



$$P(x_1, \cdots, x_n) = P(x_1) \cdot P(x_2|x_1) \cdot P(x_3|x_2) \cdot P(x_4|x_2) \cdot P(x_5|x_3, x_4)$$

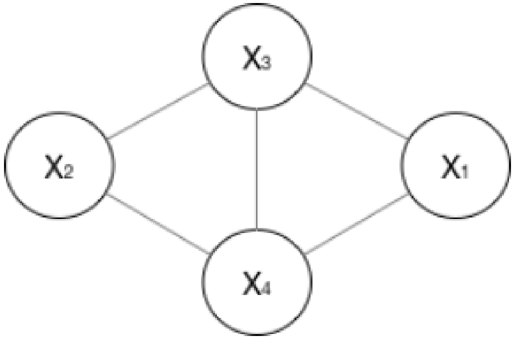
6.8 CRFs及其应用

对于无向图，不能用上式，而要采用因子分解的方式，将其表示为若干个团的联合概率乘积

注意：每个团必须是“最大团”，即最大连通子图

$$P(Y)=\frac{1}{Z(x)}\prod_{c}\psi_c(Y_c)$$

，其中 $$Z(x)=\sum_{Y}\prod_{c}\psi_c(Y_c)$$



$$P(Y)=\frac{1}{Z(x)}(\psi_1(X_1,X_3,X_4)\cdot\psi_2(X_2,X_3,X_4))$$

6.8 CRFs及其应用

$\psi_c(Y_c)$ 叫势函数，是一个最大团上随机变量的联合概率，一般用指数函数：$\psi_c(Y_c) = e^{-E(Y_c)}$

$$P(Y) = \frac{1}{Z(x)} \prod_c \psi_c(Y_c) = \frac{1}{Z(x)} \prod_c e^{\sum_k \lambda_k f_k(c,y|c,x)}$$

其中，f()为团上的特征函数，主要由它决定团的能量大小。

特征函数f定义： 特征函数f(x,y)是一个二值函数，描述x与y之间的某个事实，当x与y满足事实时取值为 1 ，否则取值为 0 。

6.8 CRFs及其应用

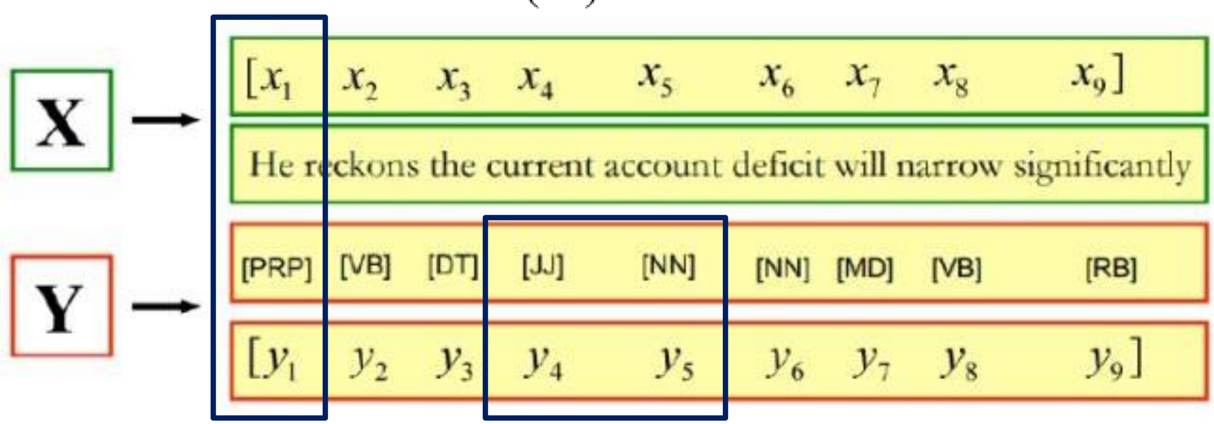
特征函数理解

特征函数定义了团中变量间的约束关系，满足为1，否则为0。

func1=if(output=PRP and feature='U00:He')   return 1 else return 0
func2=if(output=NN and feature='U00:He')   return 1 else return 0

每个特征函数Fj都会为x_j打分，特征函数的评分值越高，势函数越大。

$$p(Y|X,\lambda)=\frac{1}{Z(X)}\exp\left(\sum_j \lambda_j \cdot F_j(Y,X)\right)$$



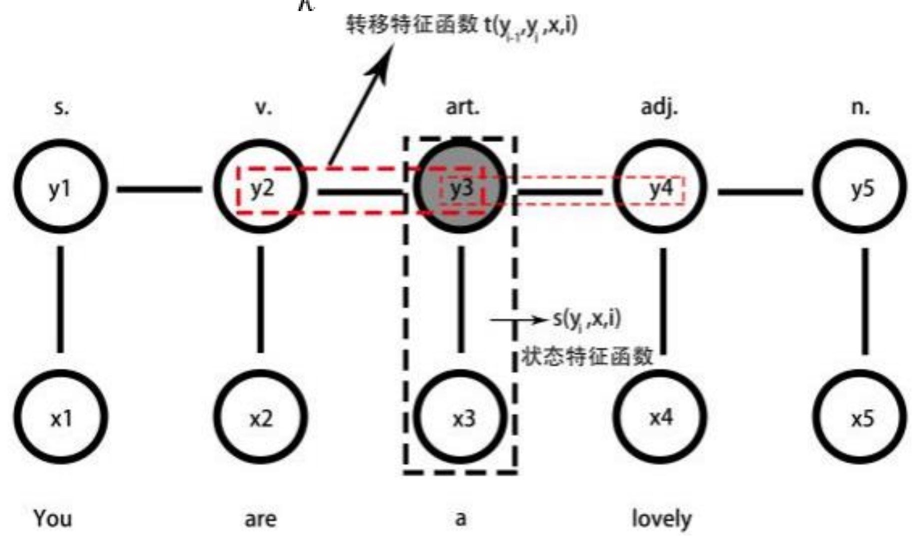
6.8 CRFs及其应用

根据上述分析，可将特征函数分为两类：

- 状态特征函数$s_k(y_i, X, i)$，表示观察序列X在i位置的标记概率；
- 转移特征函数$t_j(y_{i-1}, y_i, X, i)$，表示标注序列Y在i及i-1位置上标记的转移概率；

P(Y|X)= $\exp(\sum_j \lambda_j t_j(y_{i-1}, y_i, X, i) + \sum_k \mu_k s_k(y_i, X, i))$

λ_j和μ_k分别是 t_j和s_k的权重，需要从训练样本中估计。



6.8 CRFs及其应用

可以将两个特征函数统一表示为：

$$F_j(Y,X)=\sum_{i=1}^{n} f_j(y_{i-1},y_i,X,i) \qquad \ldots (6\text{-}34)$$

其中，每个局部特征函数 $f_j(y_{i-1},y_i,X,i)$ 表示状态特征 $s(y_{i-1},y_i,X,i)$ 或转移数 $t(y_{i-1},y_i,X,i)$。

条件随机场定义的条件概率可以由下式给出：

$$\boxed{p(Y\mid X,\lambda)=\frac{1}{Z(X)}\exp(\sum_j \lambda_j \cdot F_j(Y,X))} \qquad \ldots (6\text{-}35)$$

其中，$Z(X)$为归一化因：$Z(X)=\sum_Y \exp(\sum_j \lambda_j \cdot F_j(Y,X))$

6.8 CRFs及其应用

实现 **CRFs** 需要解决如下三个问题：

①特征函数选取

②解码

③参数训练

6.8 CRFs及其应用

①特征函数选取 $f_j(y_{i-1}, y_i, X, i)$

实际应用中，特征函数可能非常多，直接设计比较麻烦。可以采取先创建特征模版，再根据模版自动创建特征函数的方法。

下面，我们以CRF++包为例，讲解下模板构建方法。模板分为两类，一类是Unigram，一类是Bigram。

Unigram模板是比较常用的模板，这类模板提取的信息较为全面，模板数量也比较多，对应之前的状态函数。

Bigram模板，即考虑前一个状态y_{i-1}转移到当前状态y_i以及和x组合构成的特征，对应之前的转移函数。

6.8 CRFs及其应用

①特征函数选取

Unigram模板结构

特征模板格式：%x[row,col]

首字母可取U或B，对应两种类型。

方括号里的编号用于标定特征来源，

row表示相对当前位置的行，0即是当前行；

col对应训练文件中的列。

U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-2,0]/%x[-1,0]/%x[0,0]
U06:%x[-1,0]/%x[0,0]/%x[1,0]
U07:%x[0,0]/%x[1,0]/%x[2,0]
U08:%x[-1,0]/%x[0,0]
U09:%x[0,0]/%x[1,0]

6.8 CRFs及其应用

①特征函数选取

训练样本

小	B
明	I
今	B
天	I
穿	S
了	S
一	B
件	I
红	B
色	I
上	B
衣	I

以语句‘小明今天穿了一件红色上衣’分词为例，符合CRF++处理格式的训练语料如下所示。

北	N	B
京	N	E
欢	V	B
迎	V	M
你	N	E

最后列是标注（输出），前边为特征列，可以有多列多个特征

6.8 CRFs及其应用

①特征函数选取

Unigram模板结构

小	B
明	I
今	B
天	I
穿	S
了	S
一	B
件	I
红	B
色	I
上	B
衣	I

如果当前词是‘今’，那-2位置对应的字就是‘小’，每个特征对应的字如下：

U00:%x[-2,0]=====>小
U01:%x[-1,0]=====>明
U02:%x[0,0]=====>今
U05:%x[-2,0]/%x[-1,0]/%x[0,0]=====>小/明/今
U06:%x[-1,0]/%x[0,0]/%x[1,0]=====>明/今/天

如果新定义模版U11:%U[-2,1]/%U[-2,0] 什么意思

北	N	B
京	N	E
欢	V	B
迎	V	M
你	N	E

6.8 CRFs及其应用

①特征函数选取

Unigram模板结构

小	B
明	I
今	B
天	I
穿	S
了	S
一	B
件	I
红	B
色	I
上	B
衣	I

根据第一个模板U00:%x[-2,0]，系统自动创建的特征函数如下：

func1=if(output=B and feature='U00:小')　return 1 else return 0

其中output=B 指的是当前字（即'今'）的预测标记，feature表示满足的特征条件。

以上特征函数的意义：如果当前位置输出为B，且前两个位置的字为‘小’，则返回1

对每个模版中当前字，系统会重复L次（L表示标记个数，如BIS）

func2=if(output=I and feature='U00:小’) return 1 else return 0

func3=if(output=S and feature='U00:小) return 1 else return 0

注意：3个特征函数只是代表3种可能性，实际每个特征函数的值要通过训练样本来赋予。如本例func1得分加1。

6.8 CRFs及其应用

①特征函数选取

Unigram模板结构

小	B
明	I
今	B
天	I
穿	S
了	S
一	B
件	I
红	B
色	I
上	B
衣	I

对第一个模板U00:%x[-2,0]，然后下移，扫描下一个字'天'，同样会得到三个特征函数：

func4=if(output=B and feature='U00:明') return 1 else return 0

func5=if(output=I and feature='U00:明') return 1 else return 0

func6=if(output=S and feature='U00:明') return 1 else return 0

最终会生成N*T*M个特征函数，N代表分词中字的个数，T代表分词标注的tag标签（B,I,S等），M代表模板个数。

合理的“标记与特征”在样本中出现的次数多，对应的权重就高，不合理的标记在训练样本中出现的少，对应的权重就小。

6.8 CRFs及其应用

①特征函数选取

Bigram模板结构

如对模版U01:%x[0,0]，样本将产生特征函数：

北	N	B
京	N	E
欢	V	B
迎	V	M
你	N	E

func1 = if (prev_output = B and output = B and feature=B01: “北”) return 1 else return 0
func1 = if (prev_output = B and output = M and feature=B01: “北”) return 1 else return 0
func1 = if (prev_output = B and output =E and feature=B01: “北”) return 1 else return 0
func1 = if (prev_output = M and output = B and feature=B01: “北”) return 1 else return 0
func1 = if (prev_output = M and output = M and feature=B01: “北”) return 1 else return 0
func1 = if (prev_output = M and output = E and feature=B01: “北”) return 1 else return 0
func1 = if (prev_output = E and output = B and feature=B01: “北”) return 1 else return 0
func1 = if (prev_output = E and output = M and feature=B01: “北”) return 1 else return 0
func1 = if (prev_output = E and output = E and feature=B01: “北”) return 1 else return 0
...

CRFs及其应用

② 解码

条件随机场解码的过程就是给定条件随机场P(Y|X)和输入序列x，求条件概率最大的标记序列y*，即对观测序列进行标注。

可以由维特比 (**Viterbi**)算法完成。

CRFs及其应用

② 解码

$$p(Y \mid X,\lambda)=\frac{1}{Z(X)}\exp\left(\sum_j \lambda_j \cdot F_j(Y,X)\right)$$

j为位置下标

以中文分词为例：乒 乓 球 拍 卖 完 了

维特比算法就是在下面由标记组成的矩阵中搜索一条最优的路径。

乒 **B**
乓 **M**
球 **M**

乒	乓	球	拍	卖	完	了
B	B	B	B	B	B	B
M	M	M	M	M	M	M
E	E	E	E	E	E	E
S	S	S	S	S	S	S

分词结果：乒/**B** 乓/**M** 球/**M** 拍/**E** 卖/**S** 完/**S** 了/**S**

6.8 CRFs及其应用

③　参数训练

为了训练特征权重λ_j，需要计算模型的损失和梯度。由梯度更新λ_j，直到λ_j收敛。

- 损失函数定义为负对数似然函数：

$$L(\boldsymbol{\lambda}) = -\log p(Y \mid X, \boldsymbol{\lambda}) + \frac{\varepsilon}{2}\boldsymbol{\lambda}^2$$

(ε取值范围：10^{-6}~10^{-3})

- 损失函数的梯度为：$$\frac{\partial L(\boldsymbol{\lambda})}{\partial \lambda_j} = \frac{\partial \log Z(X)}{\partial \lambda_j} - F_j(Y, X) + \varepsilon \boldsymbol{\lambda}$$

6.8 CRFs及其应用

关于条件随机场模型的实现工具：

- **CRF++**（**C++**版）：
  http://crfpp.googlecode.com/svn/trunk/doc/index.html
- **CRFSuite**（**C**语言版）：
  http://www.chokkan.org/software/crfsuite/
- **MALLET**（**Java**版，通用的自然语言处理工具包，包括分类、序列标注等机器学习算法）：
  http://mallet.cs.umass.edu/
- **NLTK**（**Python**版，通用的自然语言处理工具包，很多工具是从**MALLET**中包装转成的**Python**接口）：
  http://nltk.org/

谢谢！