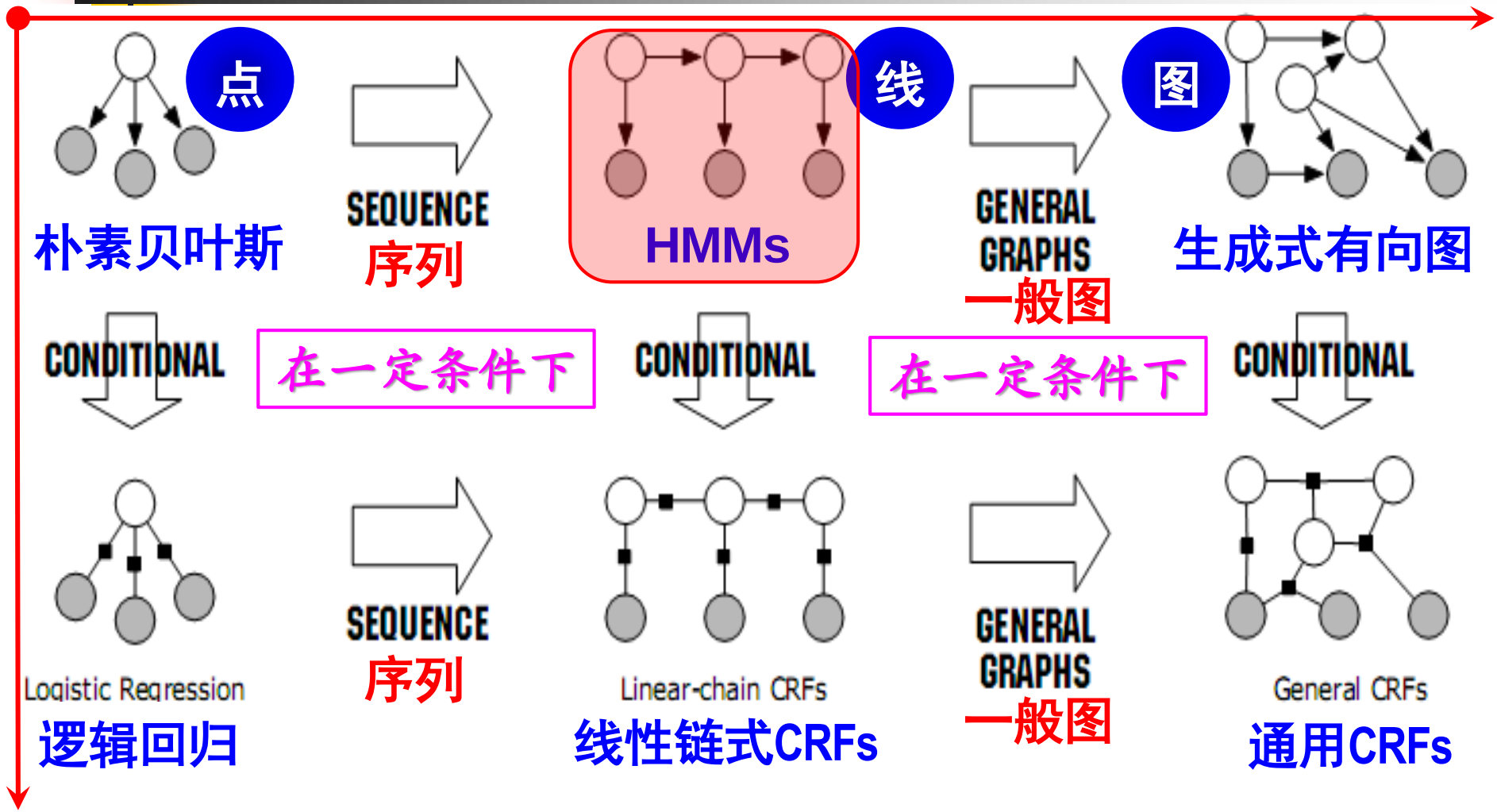
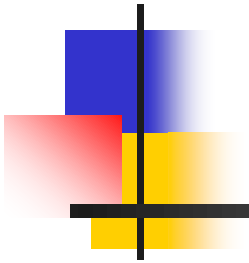
第6章 隐马尔可夫模型与条件随机场

NLP中概率图模型的演变

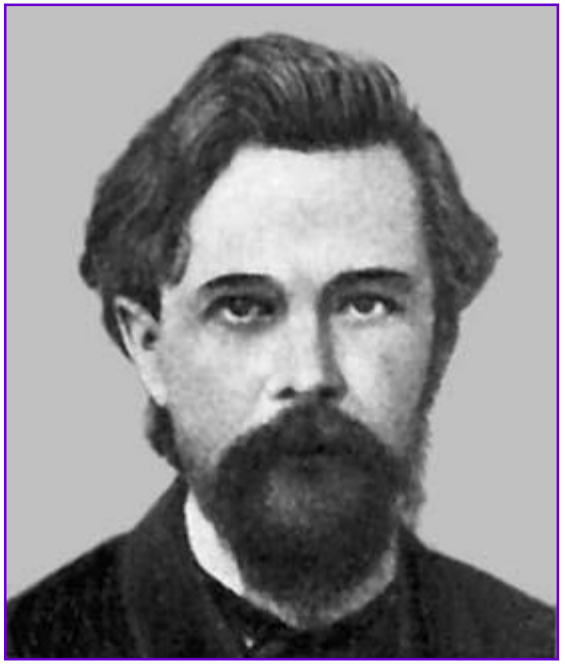




6.1 马尔可夫模型

6.1 马尔可夫模型

◆马尔可夫(Andrei Andreyevich Markov)
(1856.6.14～1922.7.20)



前苏联数学家。**切比雪夫**(1821年5月16日～1894年12月8日)的学生。在概率论、数论、函数逼近论和微分方程等方面卓有成就。他提出了用数学分析方法研究自然过程的一般图式—马尔可夫链，并开创了随机过程(马尔可夫过程)的研究。

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 $s_1, s_2, \ldots, s_N$, 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 s_j $(1 \le j \le N)$ 的概率取决于前 t-1 个时刻 $(1, 2, \ldots, t\text{-}1)$ 的状态，该概率为：

$$p(q_t = s_j \mid q_{t-1} = s_i, q_{t-2} = s_k, \cdots)$$

6.1 马尔可夫模型

●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二阶马尔可夫链：

$$p(q_t = s_j \mid q_{t-1} = s_i, q_{t-2} = s_k, \cdots) = p(q_t = s_j \mid q_{t-1} = s_i)$$

… (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：

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

该随机过程称为**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：

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

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

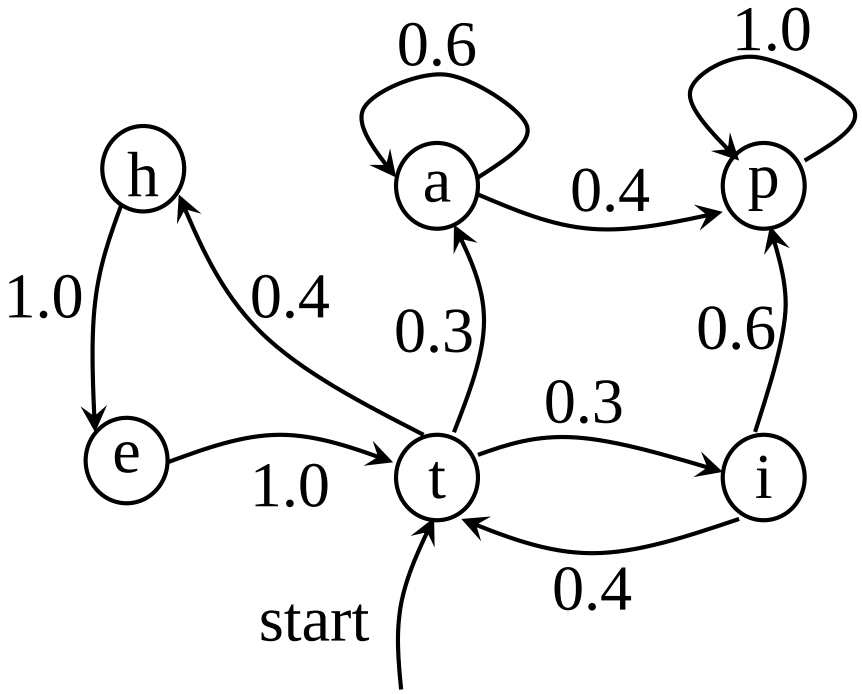
马尔可夫模型又可视为随机的有限状态自动机，该有限状态自动机的每一个状态转换过程都有一个相应的概率，该概率表示自动机采用这一状态转换的可能性。

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限状态自动机）

— 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。



6.1 马尔可夫模型

状态序列 $S_1, \ldots, S_T$ 的概率：

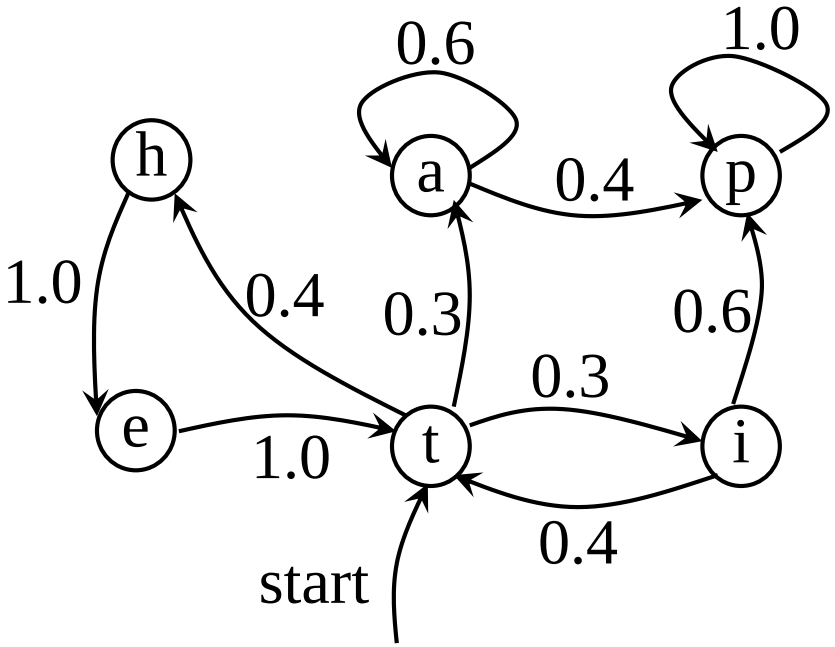
$$p(s_1,\cdots,s_T) = p(s_1)\times p(s_2\mid s_1)\times p(s_3\mid s_1,s_2)\times\cdots\times p(s_T\mid s_1,\cdots,s_{T-1})$$

$$= p(s_1)\times p(s_2\mid s_1)\times p(s_3\mid s_2)\times\cdots\times p(s_T\mid s_{T-1})$$

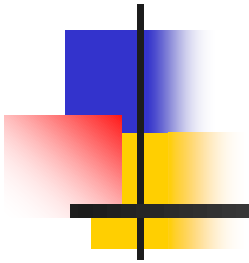
$$= \pi_{S_1}\prod_{t=1}^{T-1} a_{S_tS_{t+1}} \qquad \ldots (6.5)$$

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



$$p(\mathrm{t},\mathrm{i},\mathrm{p}) = p(s_1=\mathrm{t}) \times p(s_2=\mathrm{i} \mid s_1=\mathrm{t}) \times p(s_3=\mathrm{p} \mid s_2=\mathrm{i})$$
$$= 1.0 \times 0.3 \times 0.6$$
$$= 0.18$$



6.2 隐马尔可夫模型

6.2 隐马尔可夫模型

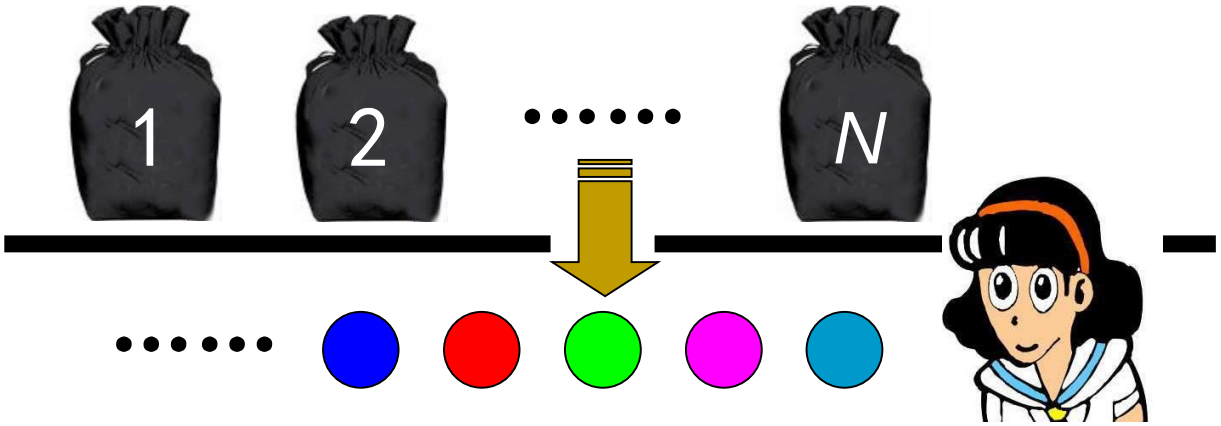
◆隐马尔可夫模型(Hidden Markov Model, HMM)

创建于20世纪70年代，美国数学家鲍姆(Leonard E. Baum)等人提出。

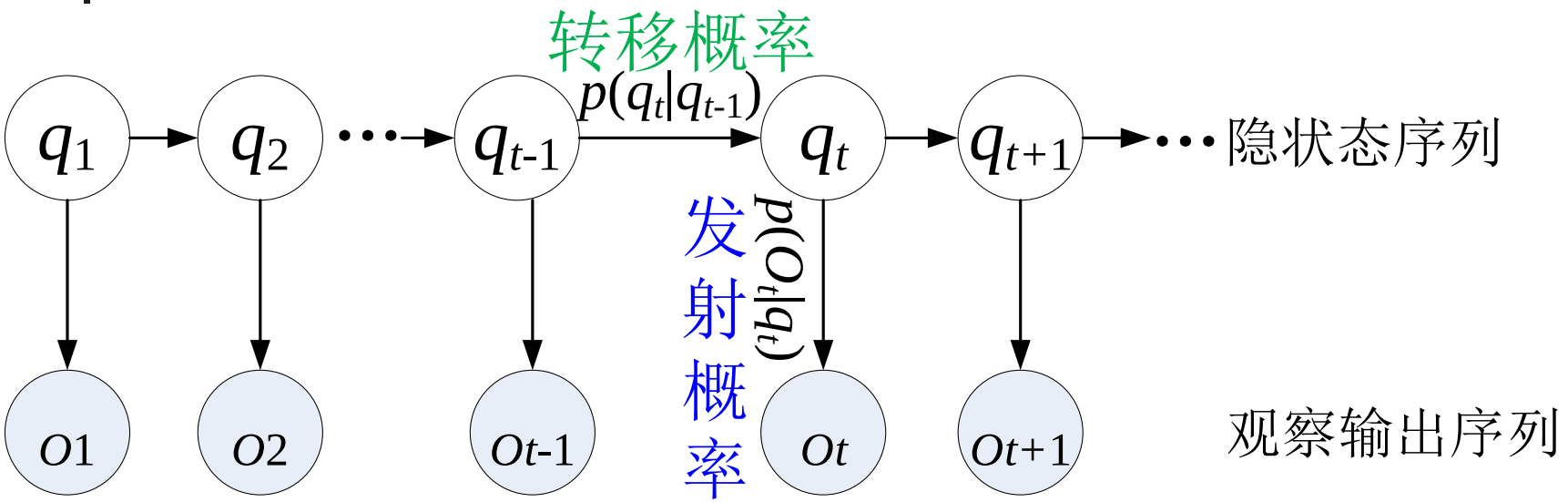
描写：该模型是一个双重随机过程，我们不知道具体的状态序列，只知道状态转移的概率，即模型的状态转换过程是不可观察的（隐蔽的），而可观察事件的随机过程是隐蔽状态转换过程的随机函数。

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，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。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。每只袋子对应HMM中的一个状态；球的颜色对应于 HMM 中状态的输出。



6.2 隐马尔可夫模型



HMM 图解

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 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) 的概率)。其中,

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

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。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，

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

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。

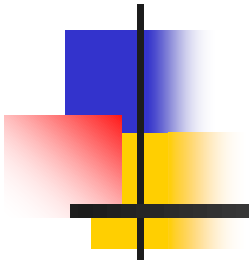
6.2 隐马尔可夫模型

◆三个问题：

(1) 在给定模型$\mu=(A, B, \pi)$ 和观察序列 $O=o_1 o_2 \ldots o_T$ 的情况下，怎样快速计算概率 $p(O|\mu)$?

(2) 在给定模型 $\mu=(A, B, \pi)$ 和观察序列 $O=o_1 o_2 \ldots o_T$ 的情况下，如何选择在一定意义下“最优”的状态序列 $Q = q_1 q_2 \ldots q_T$，使得该状态序列“最好地解释”观察序列?

(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?



6.3 前向算法

6.3 前向算法

◆问题1：快速计算观察序列概率$p(O|\mu)$

给定模型 $\mu=(A, B, \pi)$ 和观察序列$O=o_1 o_2 \ldots o_T$，
快速计算 $p(O|\mu)$：

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

$$p(O|\mu) = \sum_Q p(O,Q|\mu) = \sum_Q p(Q|\mu) \times p(O|Q,\mu) \quad \ldots (6.9)$$

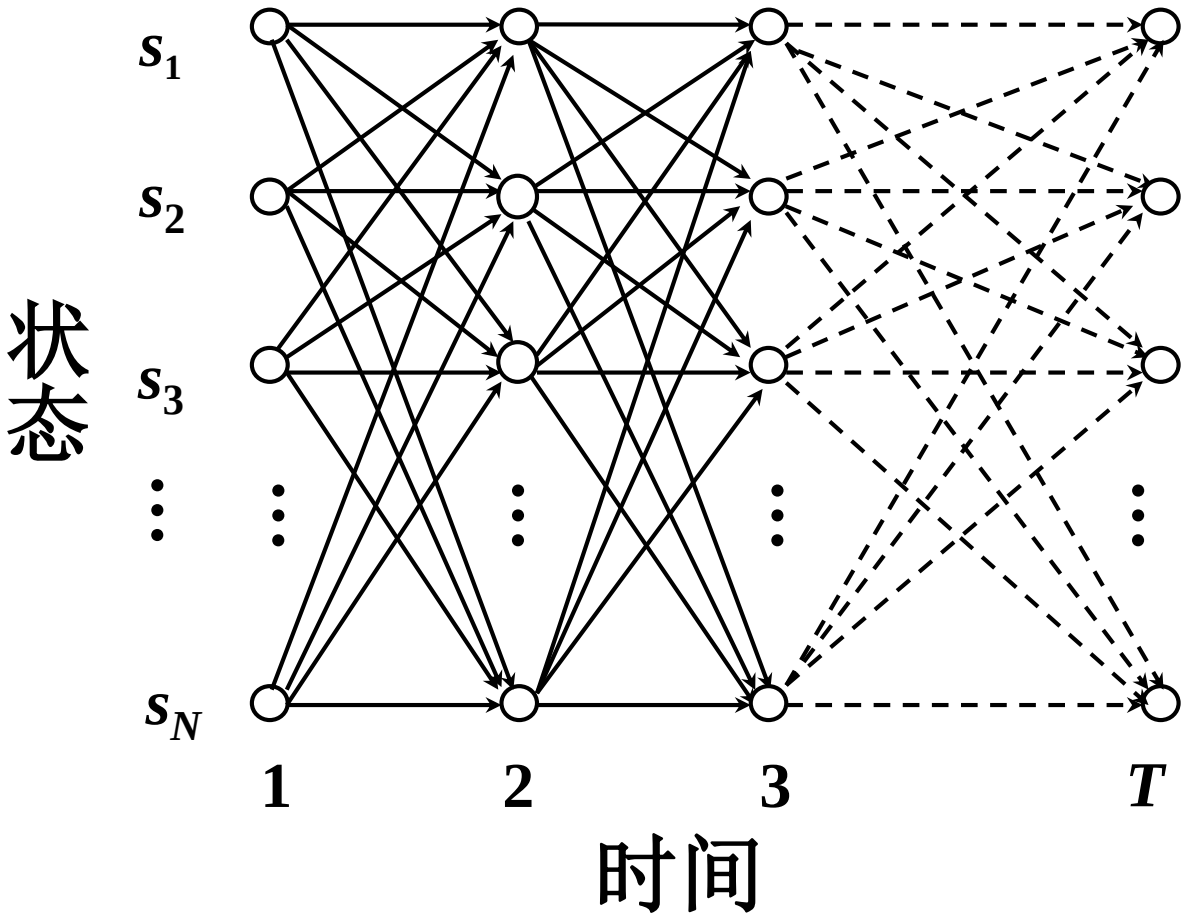
$$p(Q|\mu) = \pi_{q_1} \times a_{q_1 q_2} \times a_{q_2 q_3} \times \cdots \times a_{q_{T-1} q_T} \quad \ldots (6.10)$$

$$p(O|Q,\mu) = b_{q_1}(o_1) \times b_{q_2}(o_2) \times \cdots \times b_{q_T}(o_T) \quad \ldots (6.11)$$

状态q_1生成观察值o_1的发射概率

状态q_{T-1}转换为状态q_T的转移概率

6.3 前向算法



● **困难：**

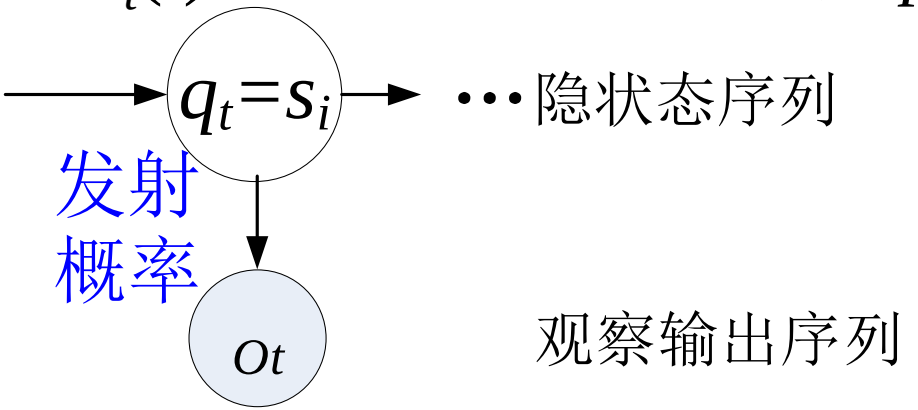
如果模型μ 有 N 个不同的状态，时间长度为 T，那么有 N^T 个可能的状态序列，搜索路径成指数级组合爆炸。

6.3 前向算法

- **解决办法：动态规划**
  前向算法(The forward procedure)
- **基本思想：定义前向变量 $\alpha_t(i)$：在时间t，输出序列$o_1o_2\ldots o_t$并且位于状态s_i的概率**

$$\alpha_t(i) = p(o_1 o_2 \cdots o_t, q_t = s_i \mid \mu) \qquad \ldots(6.12)$$

如果可以高效地计算$\alpha_t(i)$，就可以高效地求得 $p(O|\mu)$。



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(所有可能状态的概率之和)：

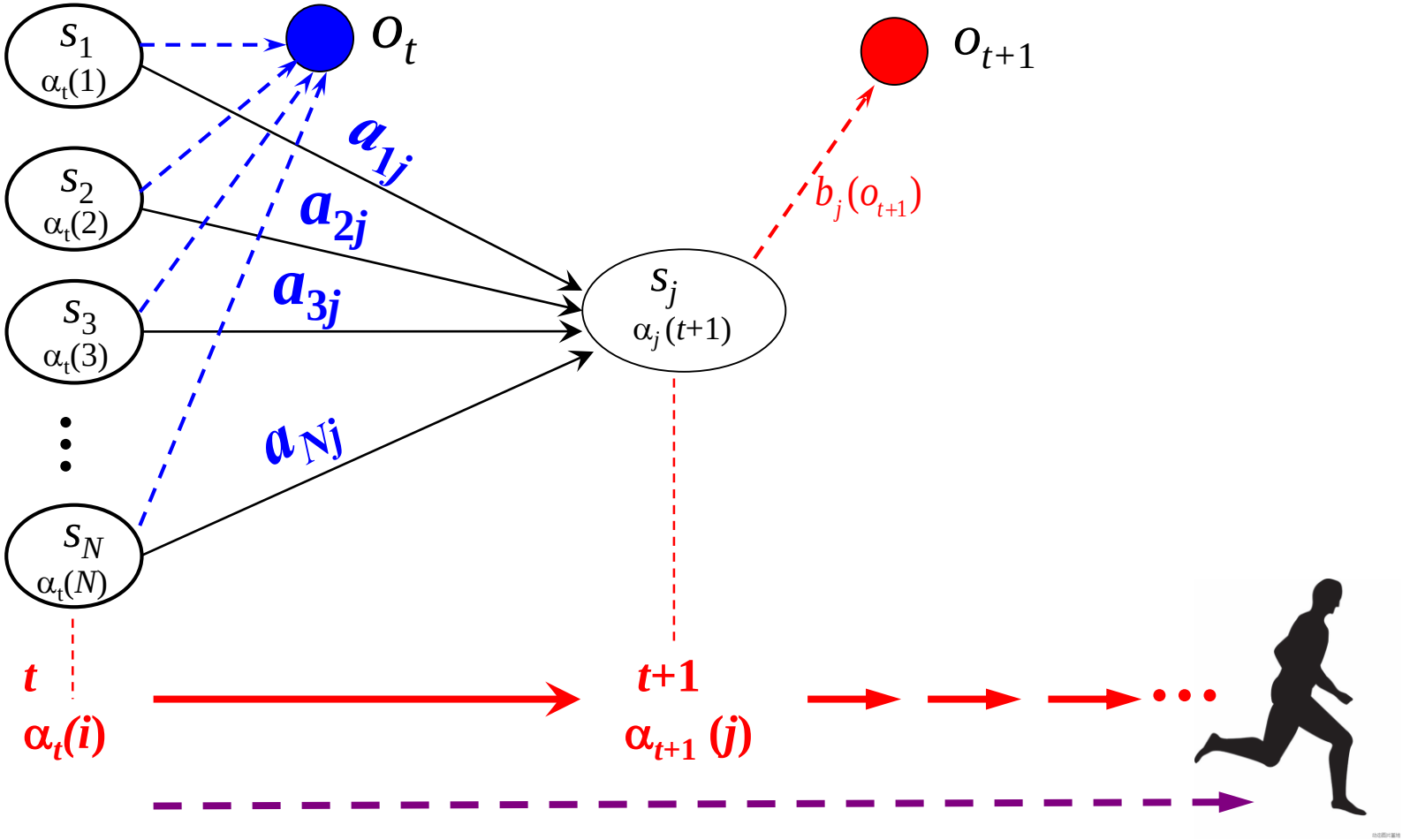
$$p(O\,|\,\mu) = \sum_{s_i} p(o_1 o_2 \cdots o_T, q_T = s_i\,|\,\mu) = \sum_{i=1}^{N} \alpha_T(i) \quad \ldots (6.13)$$

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：

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

6.3 前向算法

$$\alpha_{t+1}(j)=[\sum_{i=1}^{N}\alpha_t(i)a_{ij}]\times b_j(o_{t+1})$$



6.3 前向算法

- **算法6.1：前向算法描述**

(1) 初始化： $\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \le i \le N$

(2) 循环计算：

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

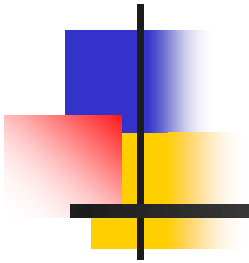
(3) 结束，输出：

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

6.3 前向算法

- **算法的时间复杂性：**

每计算一个 $\alpha_t(i)$ 必须考虑从 t-1 时的所有 N 个状态转移到状态 s_i 的可能性，时间复杂性为 $O(N)$，对应每个时刻 t，要计算 N 个前向变量：$\alpha_t(1), \alpha_t(2), \ldots, \alpha_t(N)$，所以，时间复杂性为：$O(N)\times N = O(N^2)$。 又因 $t = 1, 2, \ldots, T$，所以前向算法总的复杂性为：$O(N^2T)$。



6.4 后向算法

6.4 后向算法

- **后向算法** (The backward procedure)

定义后向变量 $\beta_t(i)$ 是在给定了模型 $\mu=(A,B,\pi)$ 和假定在时间 t 状态为 s_i 的条件下，模型输出观察序列 $o_{t+1}o_{t+2}\cdots o_T$ 的概率：

$$\beta_t(i)=p(o_{t+1}o_{t+2}\cdots o_T \mid q_t=s_i,\mu) \qquad \cdots (6.15)$$

6.4 后向算法

与前向变量一样，运用动态规划计算后向量：

(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 $\boldsymbol{o}_{t+1}$；

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

6.4 后向算法

第一步的概率： $a_{ij} \times b_j(o_{t+1})$

第二步的概率按后向变量的定义为：$\beta_{t+1}(j)$

于是，有归纳关系：

$$\beta_t(i) = \sum_{j=1}^{N} [a_{ij} b_j(o_{t+1}) \times \beta_{t+1}(j)] \qquad \ldots (6.16)$$

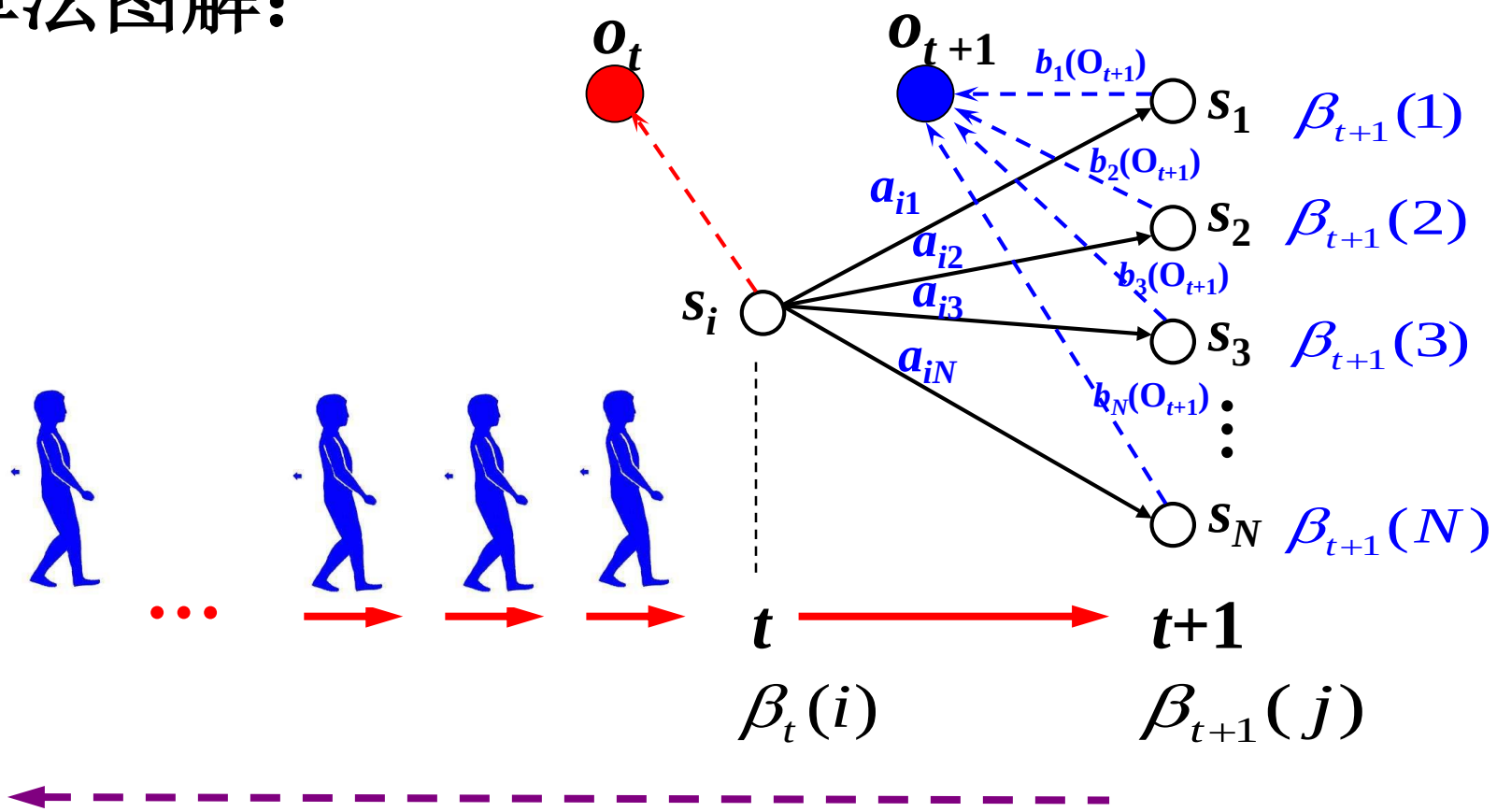
归纳顺序： $\beta_T(x), \beta_{T-1}(x), \cdots, \beta_1(x)$

(x 为模型的状态)

6.4 后向算法

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

算法图解：



6.4 后向算法

- **算法6.2：后向算法描述**

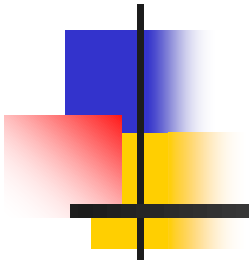
(1)初始化： $\beta_T(i)=1,\quad 1\le i\le N$

(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： $p(O\mid\mu)=\sum_{i=1}^{N}\beta_1(i)\times\pi_i\times b_i(o_1)$

算法的时间复杂性： $O(N^2T)$



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

解释不是唯一的，关键在于如何理解“最优”的状态序列？一种解释是：状态序列中的每个状态都单独地具有概率，对于每个时刻 $t\ (1 \le t \le T)$，寻找 q_t 使得 $\gamma_t(i) = p(q_t = s_i \mid O, \mu)$ 最大。

6.5 Viterbi 搜索算法

$$\gamma_t(i) = p(q_t = s_i \mid O, \mu) = \frac{p(q_t = s_i, O \mid \mu)}{p(O \mid \mu)} \quad \ldots (6.17)$$

模型的输出序列 O，并且在时间 t 到达状态 s_i 的概率。

6.5 Viterbi 搜索算法

- **分解过程：**

(1) 模型在时间 t 到达状态 s_i, 并且输出$O=o_1o_2\ldots o_T$。

(2) 根据前向变量的定义（在时间t，输出序列$o_1o_2\ldots o_t$并且位于状态s_i的概率**），实现这一步的概率为**$\alpha_t(i)$。

(3) 根据后向变量的定义（在时间t 状态为s_i 的条件下，模型输出观察序列$o_{t+1}\ldots o_T$的概率**），实现这一步的概率为**$\beta_t(i)$。

于是：

$$p(q_t = s_i, O \mid \mu) = \alpha_t(i) \times \beta_t(i) \qquad \ldots (6.18)$$

6.5 Viterbi 搜索算法

而 $p(O|\mu)$与时间 t 的状态无关，因此：

$$p(O|\mu)=\sum_{i=1}^{N}\alpha_t(i)\times\beta_t(i) \quad \ldots (6.19)$$

将公式(6.18)和(6.19)带入(6.17)式得：

$$\gamma_t(i)=\frac{\alpha_t(i)\times\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\times\beta_t(i)} \quad \ldots (6.20)$$

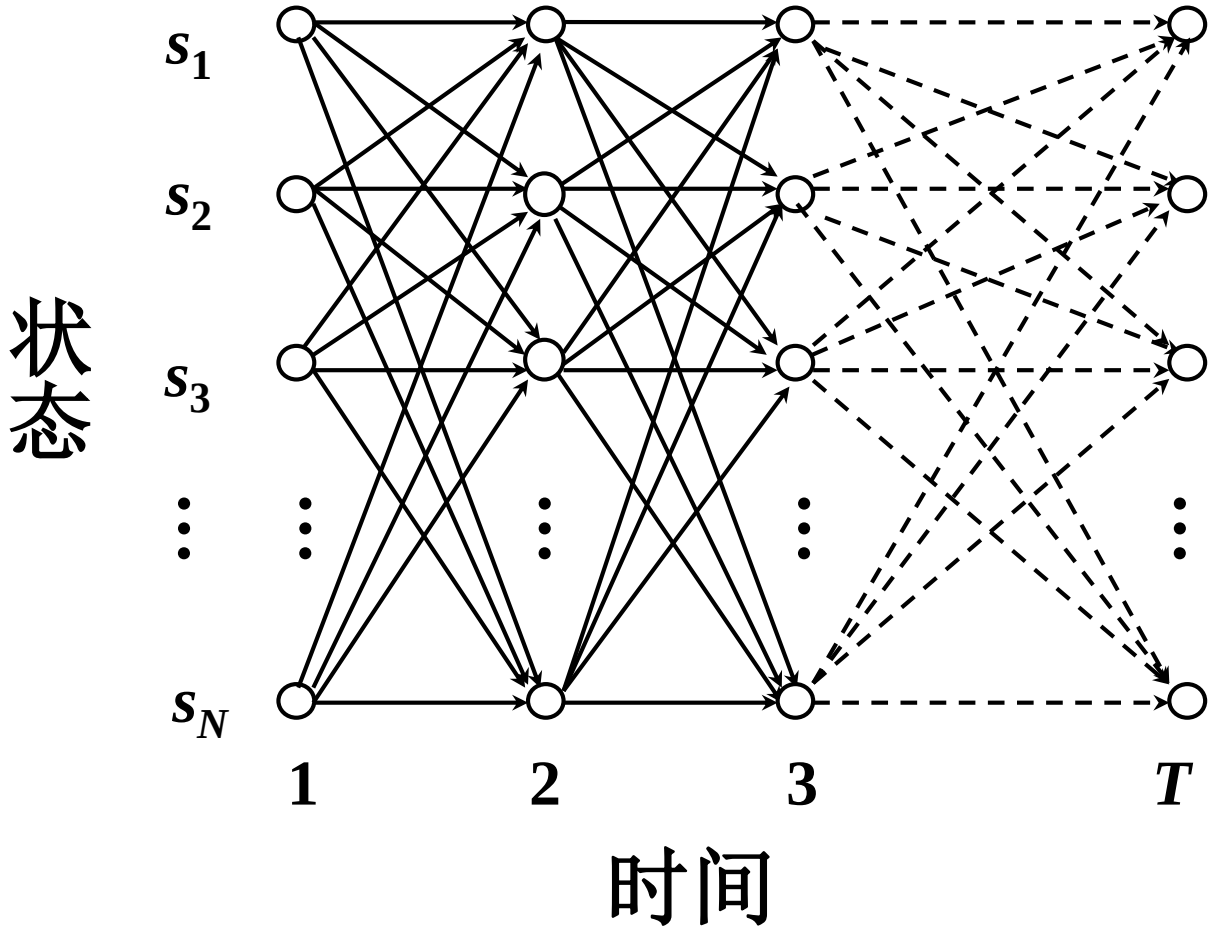
t 时刻的最优状态为：$\hat{q}_t=\arg\max_{1\le i\le N}(\gamma_t(i))$

6.5 Viterbi 搜索算法

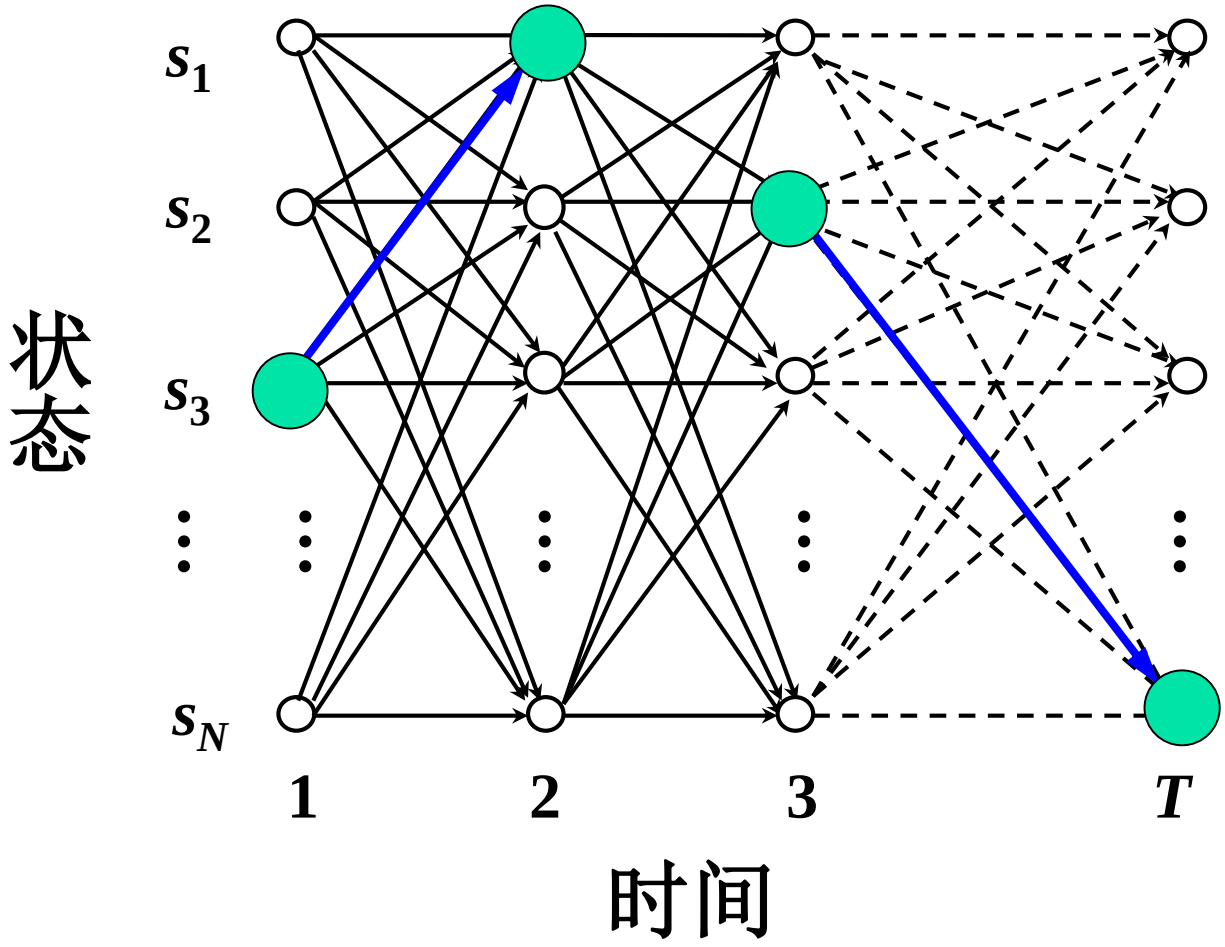
- **问题：**

每一个状态单独最优不一定使整体的状态序列最优，可能两个最优的状态 $\hat{q}_t$ 和 $\hat{q}_{t+1}$ 之间的转移概率为0，即 $a_{\hat{q}_t\hat{q}_{t+1}} = 0$ 。

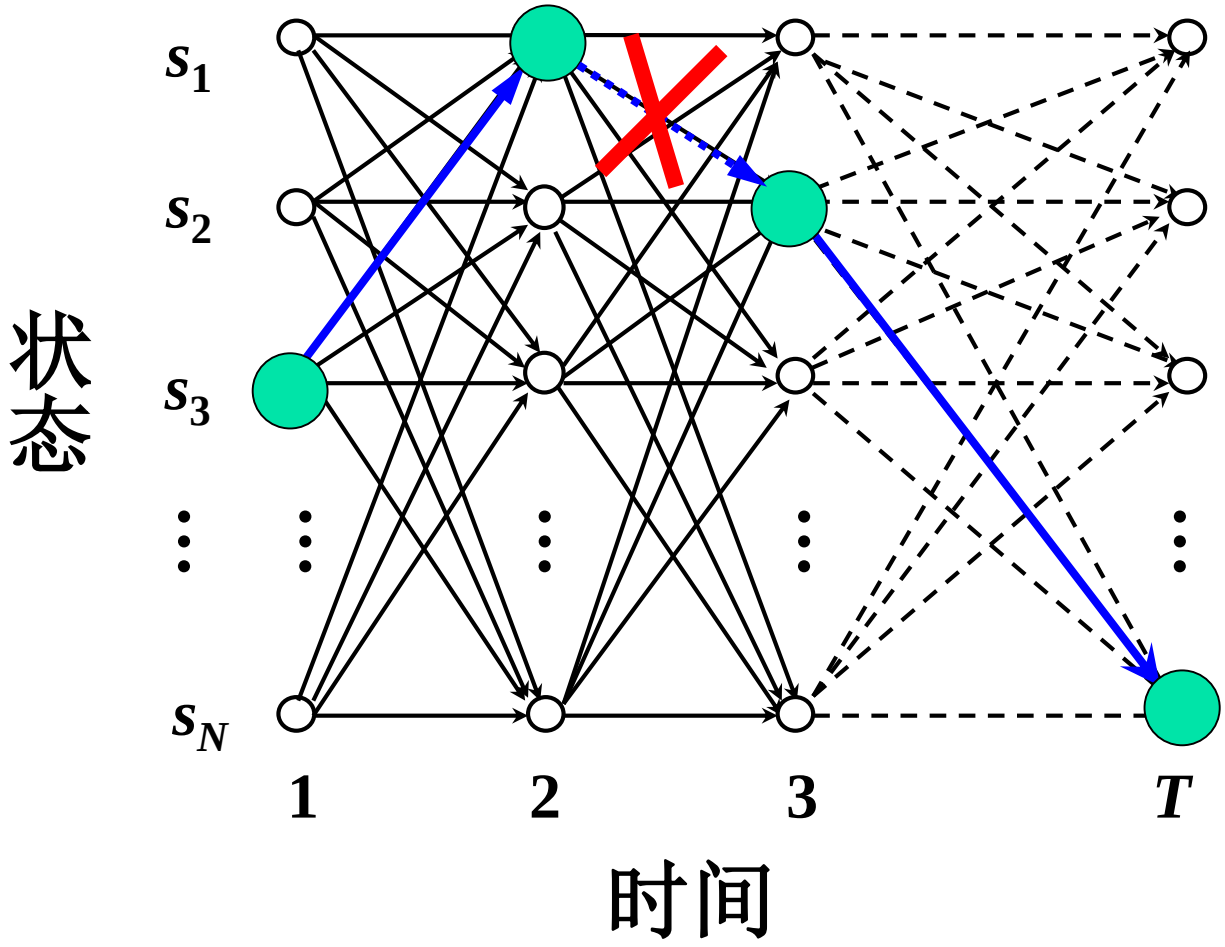
6.5 Viterbi 搜索算法



6.5 Viterbi 搜索算法



6.5 Viterbi 搜索算法



6.5 Viterbi 搜索算法

另一种解释：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：

$$\hat{Q} = \arg\max_{Q} p(Q \mid O, \mu) \qquad \ldots (6.21)$$

Viterbi 算法: 动态搜索最优状态序列。

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，并输出观察序列 $O=o_1o_2\ldots o_t$ 的最大概率：

$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} p(q_1,q_2,\cdots,q_t = s_i, o_1o_2\cdots o_t \mid \mu) \qquad \ldots (6.22)$$

6.5 Viterbi 搜索算法

从状态s_j转移到状态s_i的概率

状态s_i发射观察值o_{t+1}的概率

递归计算：$\delta_{t+1}(i)=\max_j[\delta_t(j)\cdot a_{ji}]\cdot b_i(o_{t+1})$ … (6.23)

- **算法6.3：Viterbi 算法描述**

(1)初始化： $\delta_1(i)=\pi_i b_i(o_1),\quad 1\le i\le N$

概率最大对应的路径变量： $\psi_1(i)=0$

(2)递推计算：

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

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

6.5 Viterbi 搜索算法

(3)结束：

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

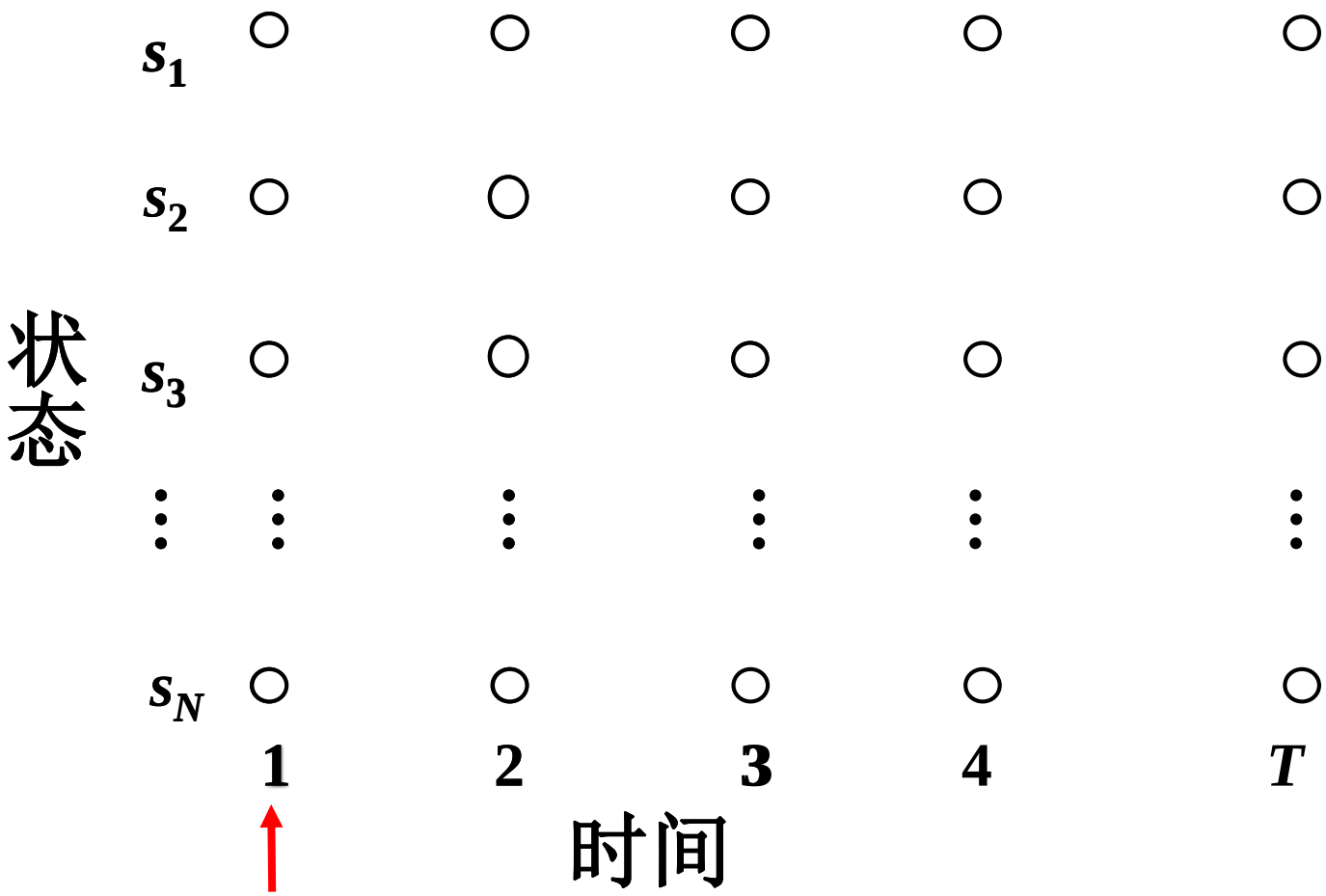
(4)通过回溯得到路径（状态序列）：

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

算法的时间复杂度： $O(N^2T)$

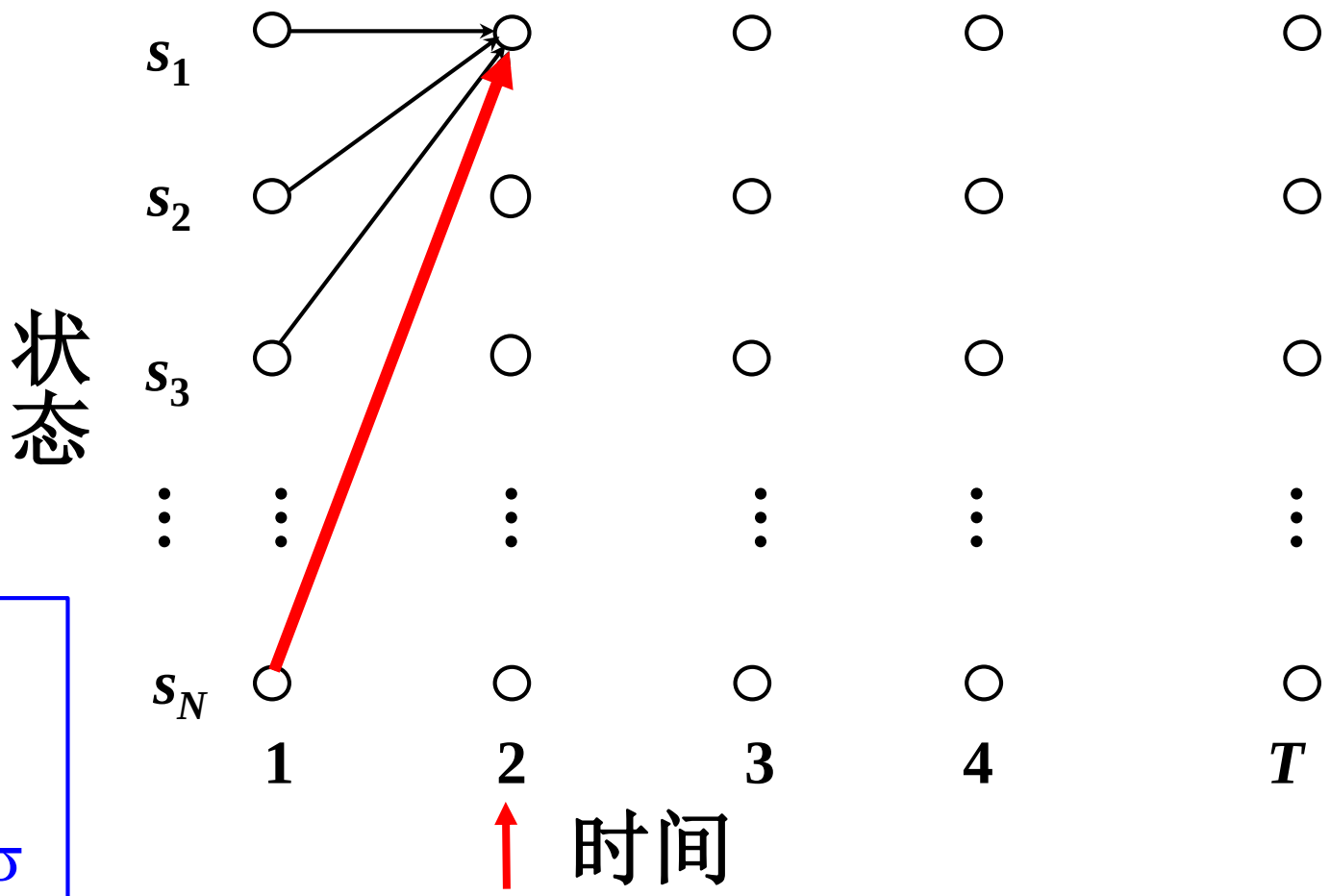
6.5 Viterbi 搜索算法

图解Viterbi搜索过程



6.5 Viterbi 搜索算法

图解 Viterbi 搜索过程



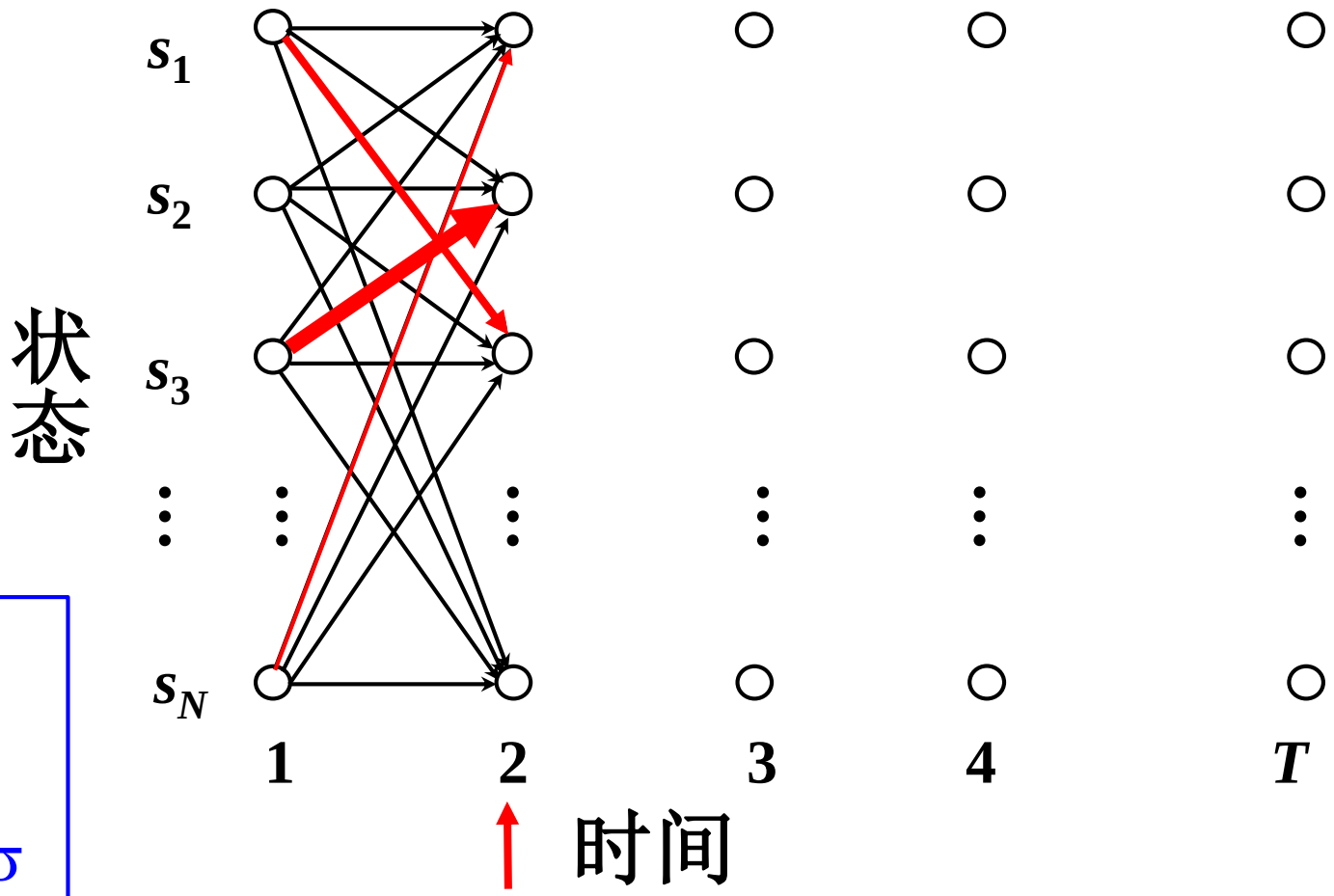
剪枝策略：

❶ $\delta_t(j) \geq \Delta$

❷ NPath $\leq \sigma$

6.5 Viterbi 搜索算法

图解Viterbi搜索过程



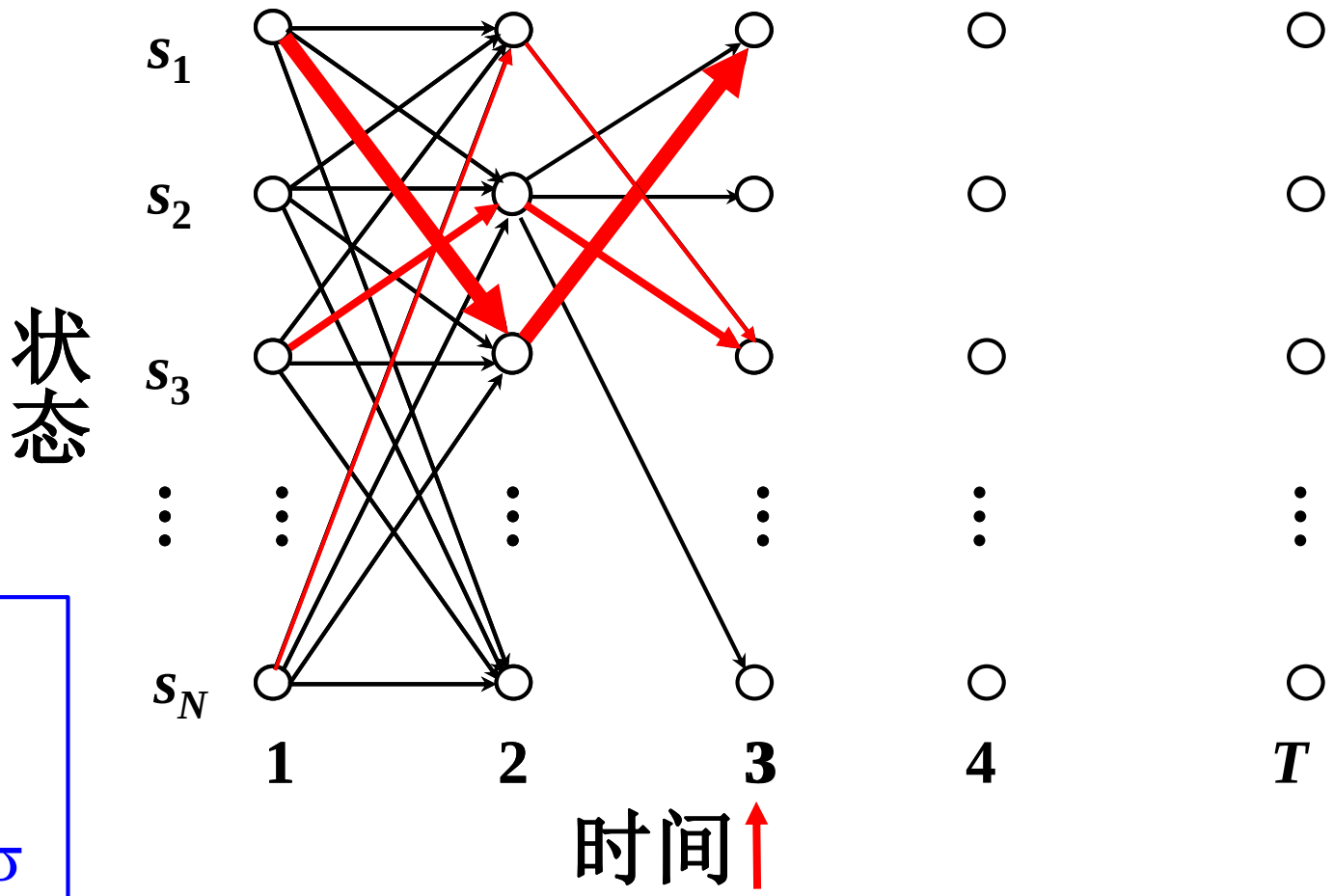
剪枝策略：

❶ $\delta_t(j) \geq \Delta$

❷ NPath $\leq \sigma$

6.5 Viterbi 搜索算法

图解 Viterbi 搜索过程



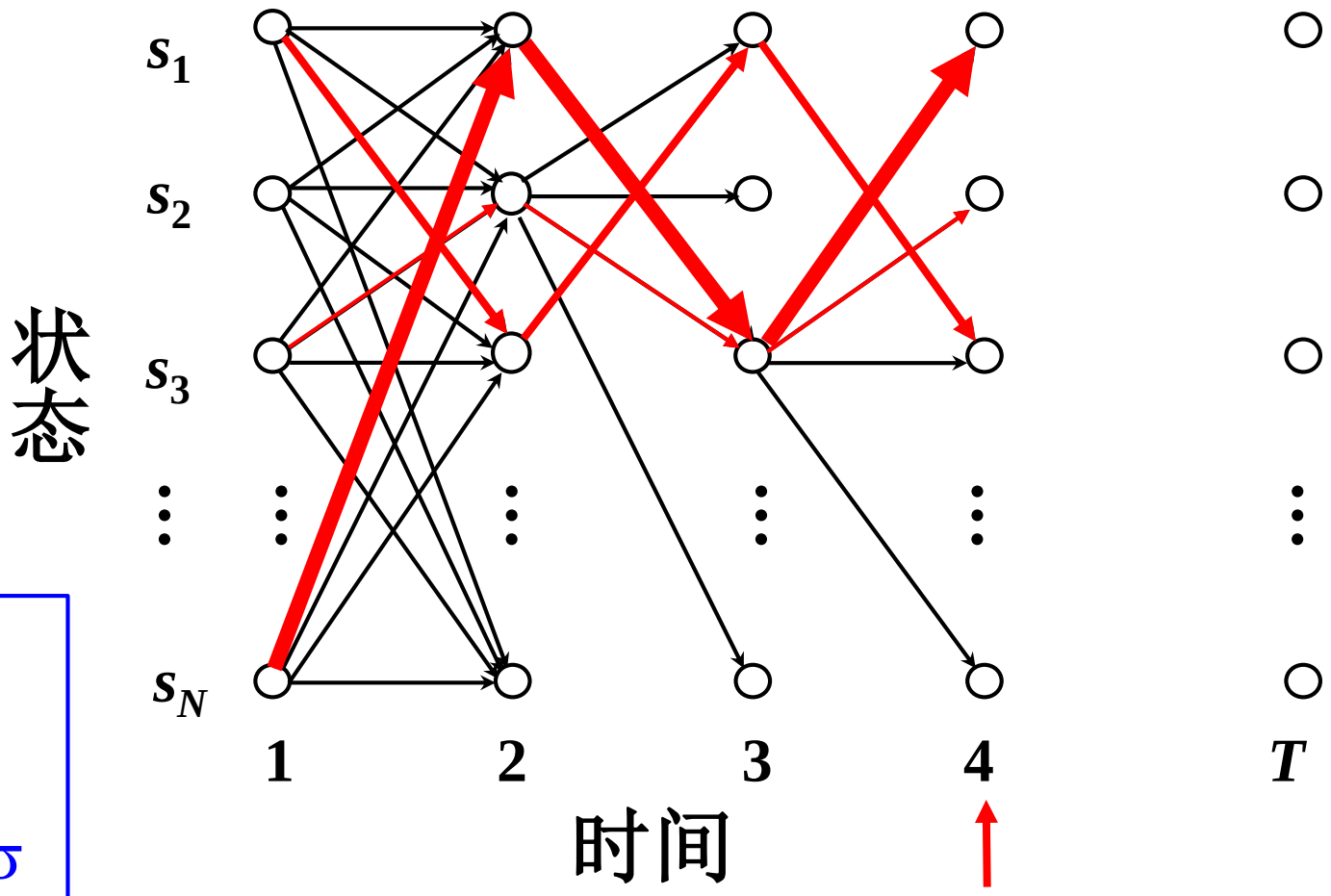
剪枝策略：

❶ $\delta_t(j) \geq \Delta$

❷ NPath $\leq \sigma$

6.5 Viterbi 搜索算法

图解 Viterbi 搜索过程



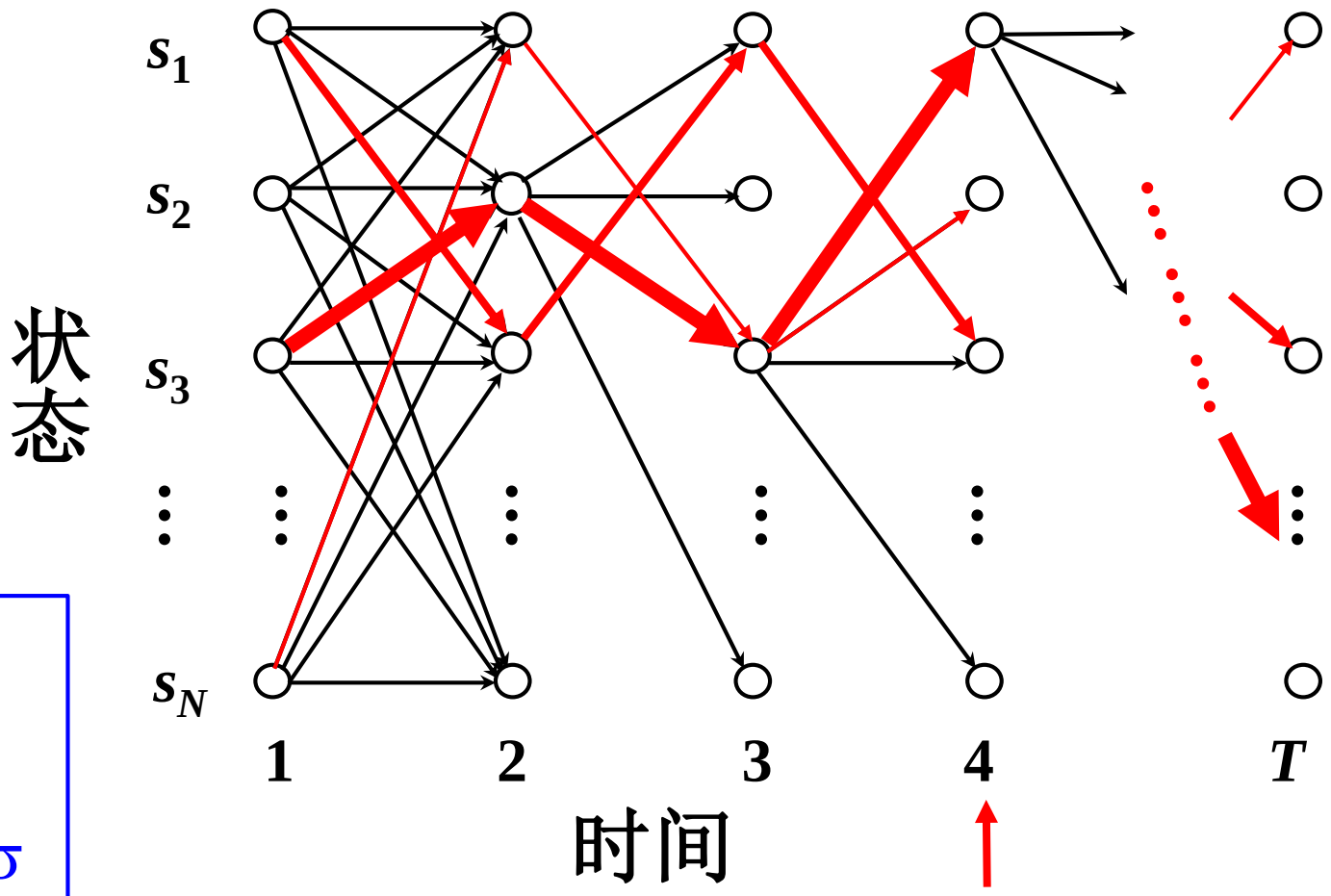
剪枝策略：

❶ $\delta_t(j) \geq \Delta$

❷ NPath $\leq \sigma$

6.5 Viterbi 搜索算法

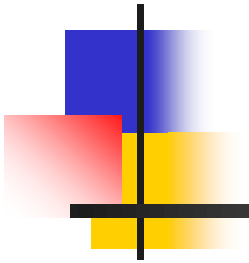
图解Viterbi搜索过程



剪枝策略：

❶ $\delta_t(j) \geq \Delta$

❷ NPath $\leq \sigma$



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 $O = o_1 o_2 \ldots o_T$，如何根据最大似然估计来求模型的参数值？或者说如何调节模型 μ 的参数，使得 $p(O|\mu)$ 最大？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。

- **前向后向算法**

  (Baum-Welch or forward-backward procedure)

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(即存在大量标注的样本), 可以用最大似然估计来计算 μ 的参数:

$$\overline{\pi}_i = \delta(q_1, s_i)$$

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

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

其中，$\delta(x, y)$ 为克罗奈克(Kronecker)函数，当 $x = y$ 时, $\delta(x, y)=1$, 否则$\delta(x, y) = 0$。

6.6 参数学习

类似地，

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

$$= \frac{\sum_{t=1}^{T} \delta(q_t, s_j) \times \delta(o_t, v_k)}{\sum_{t=1}^{T} \delta(q_t, s_j)} \qquad \ldots (6.25)$$

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

如果不存在大量标注的样本 –

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

基本思想：初始化时随机地给模型的参数赋值(遵循限制规则，如：从某一状态出发的转移概率总和为1，得到模型μ_0,

然后可以从μ_0 得到从某一状态转移到另一状态的期望次数，然后以期望次数代替公式中的次数，得到模型参数的新估计，由此得到新的模型μ_1,

从 μ_1 又可得到模型中隐变量的期望值，由此重新估计模型参数。循环这一过程，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 $O=o_1 o_2 \ldots o_T$，那么，在时间 t 位于状态 s_i，时间 $t+1$ 位于状态 s_j 的概率：

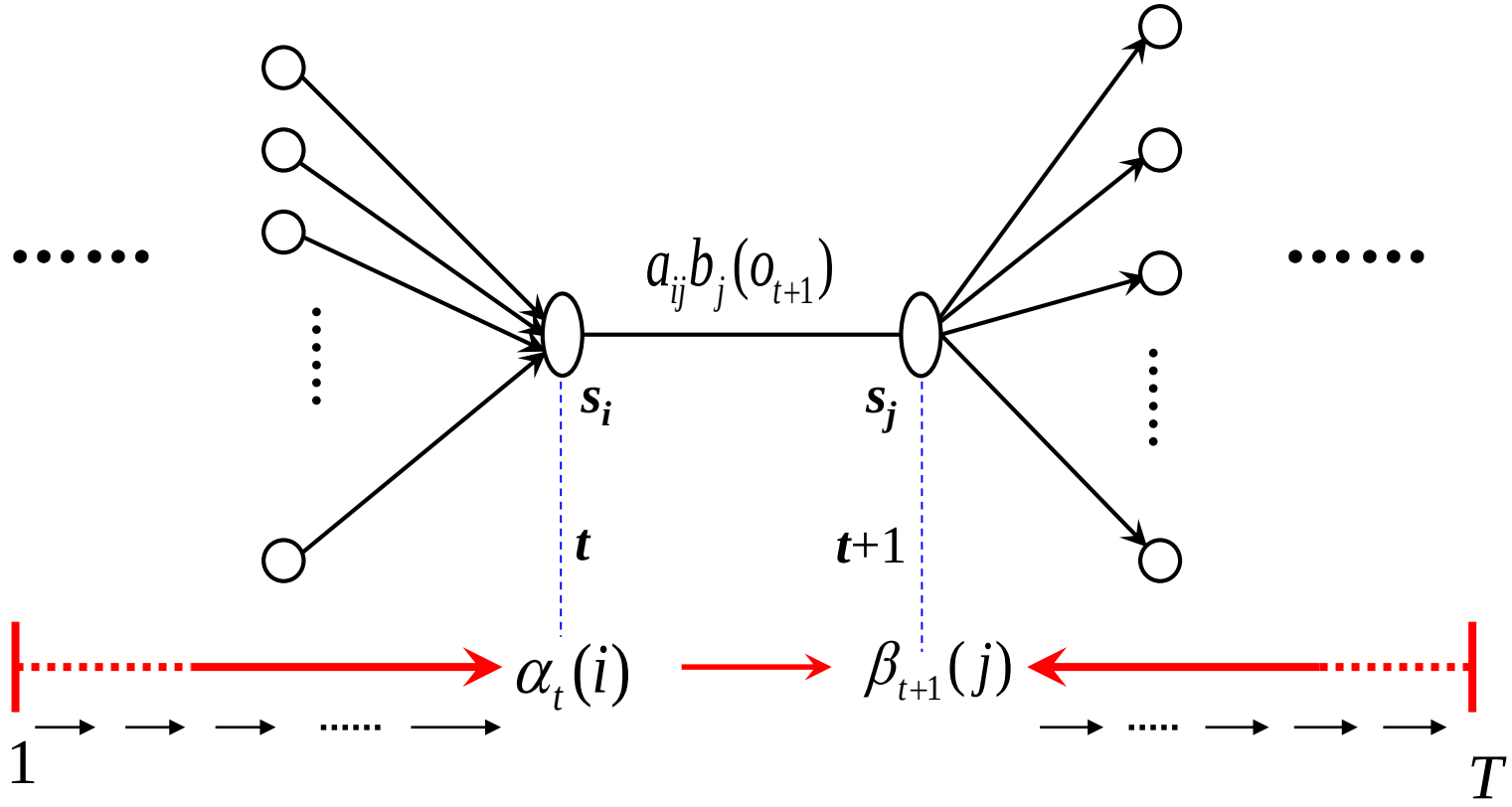
$$\xi_t(i,j) = p(q_t = s_i, q_{t+1} = s_j \mid O, \mu) = \frac{p(q_t = s_i, q_{t+1} = s_j, O \mid \mu)}{p(O \mid \mu)}$$

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

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

6.6 参数学习

图解搜索过程：



6.6 参数学习

那么，给定模型 μ 和观察序列 $O = o_1 o_2 \cdots o_T$ ，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)

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

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

(3)

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

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

6.6 参数学习

- **算法6.4：Baum-Welch 算法(前向后向算法)描述：**

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

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

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 $\gamma_t(i)$。

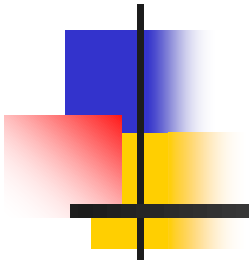
M-步：用E-步中所得到的期望值，根据公式 (6.28-6.30) 重新估计π_i, a_{ij}, $b_j(k)$ 得到模型μ_{i+1}。

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

- HMM使用中注意的问题
- Viterbi 算法运算中的小数连乘，出现溢出
  - 取对数
- Baum-Welch 算法的小数溢出
  - 放大系数
  - 参阅[Rabiner and Juang, 1993: pp. 365-368]
  - 参阅 http://htk.eng.cam.ac.uk/



6.7 HMM应用举例

6.7 应用举例

汉语的自动分词与词性标注问题。举例：

武汉市长江大桥于1957年9月6日竣工。

列出所有可能的切分：

① 武汉市/N 长江/N 大桥/N 于/P 1957年9月6日/Time 竣工/V。 /Pun

② 武汉/N 市长/N 江大桥/N 于/P 1957年9月6日/Time 竣工/V。 /Pun

6.7 应用举例

用 HMM 解决问题必须考虑的几个问题：

(1) 如何确定状态、观察及其各自的数目？

(2) 参数估计：初始状态概率、状态转移概率、输出概率如何确定？

思路：

如果把汉语自动分词结果作为观察序列 $O=O_1O_2\ldots O_T$，那么，要求解的是：$\hat{O}=\arg\max_{O} p(O|\mu)$ 。

对于词性标注而言，则需求解：$\hat{Q}=\arg\max_{Q} p(Q|O,\mu)$ 。

6.7 应用举例

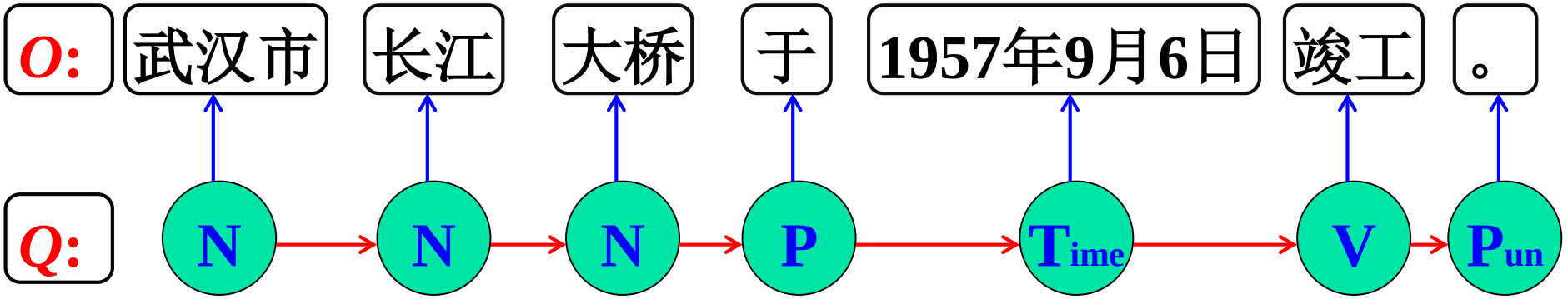
进一步解释：

(1)估计HMM模型$\mu=(A, B, \pi)$ 的参数；(Train)

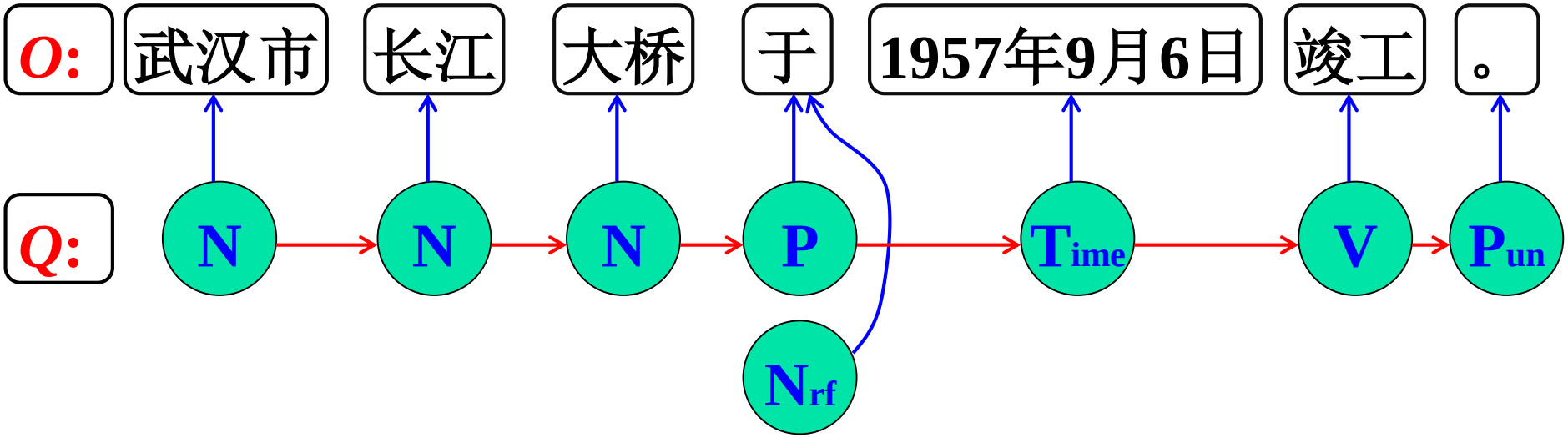
(2)对于任意给定的一个输入句子及其可能的输出序列O，求找所有可能的O中使概率$p(O|\mu)$最大的解；(分词)

(3)快速地选择“最优”的状态序列Q(词性序列)，使其最好地解释观察序列$p(Q|O,\mu)$。(词性标注)

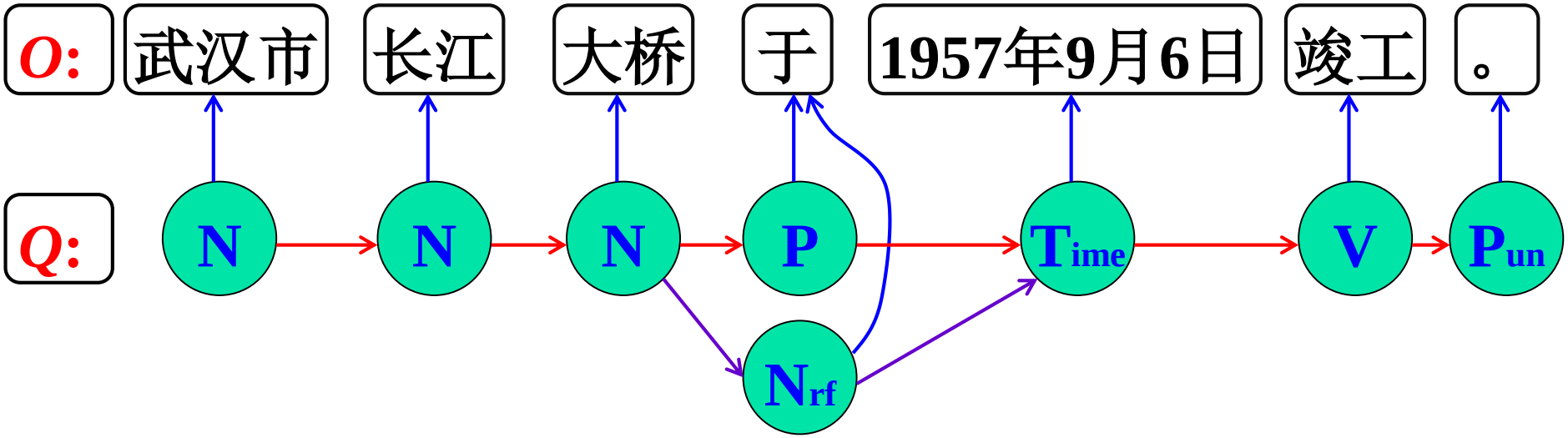
6.7 应用举例



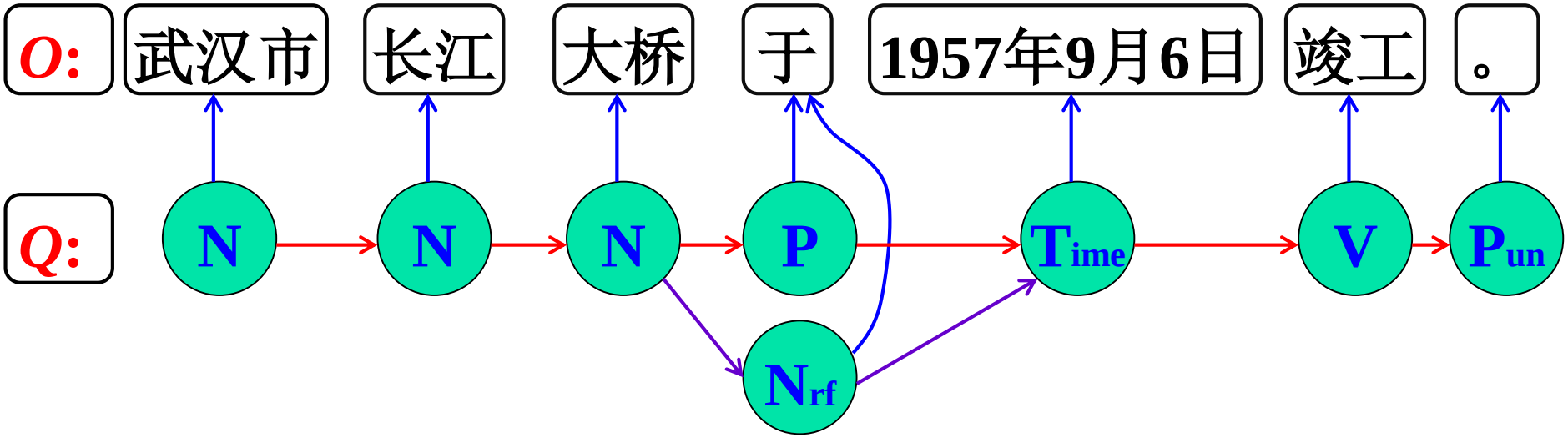
6.7 应用举例



6.7 应用举例



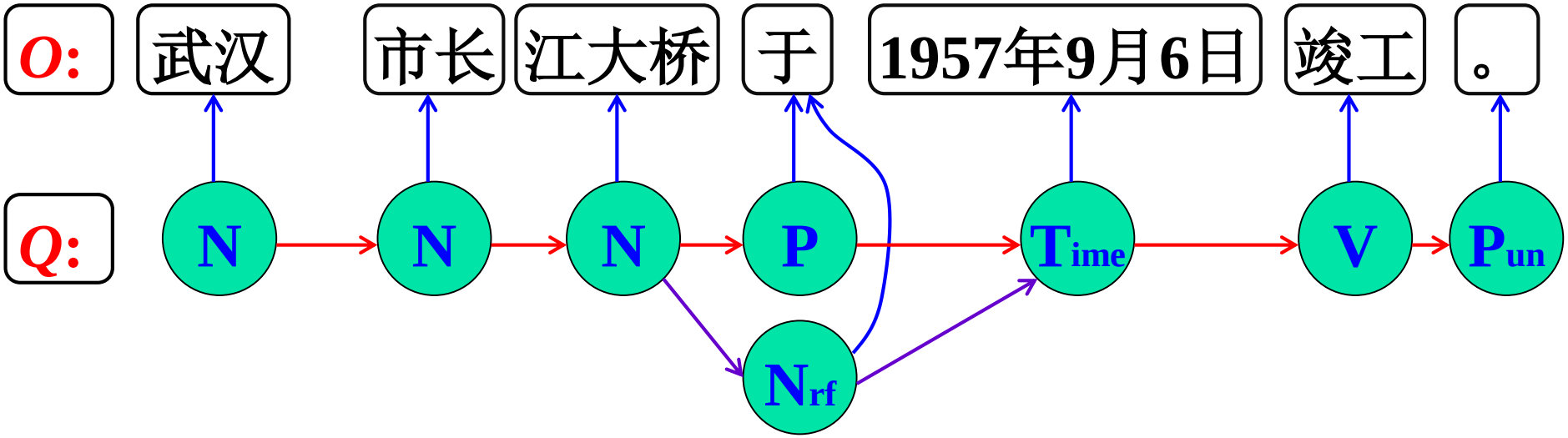
6.7 应用举例



a) 武汉市/N 长江/N 大桥/N 于/P 1957年9月6日/Time
竣工/V 。/Pun

b) 武汉市/N 长江/N 大桥/N 于/Nrf 1957年9月6日/Time
竣工/V 。/Pun

6.7 应用举例



c) 武汉/N 市长/N 江大 桥/N 于/P 1957年9月6日/Time
竣工/V 。/Pun

d) 武汉/N 市长/N 江大桥/N 于/Nrf 1957年9月6日/Time
竣工/V 。/Pun

6.7 应用举例

◆**问题1：模型参数**

(1) 观察序列：单词序列

(2) 状态序列：词类标记序列

(3) 状态数目*N*：为词类标记符号的个数，如 Upenn LDC 汉语树库中有33个词类，北大语料库词类标记符号106个等；

(4) 输出符号数*M*：每个状态可输出的不同词汇个数(和为总的词汇数)，如汉语介词 P 约有60个，连词C约有110个，即状态 P 和 C分别对应的输出符号数为60、110。

6.7 应用举例

◆参数估计

(1) 如果无任何标注语料：需要一部有词性标注的词典，采用无指导学习方法：

(*a*)获取词类个数*N*(状态数)；

(*b*)获取对应每种词类的词汇数(输出符号数)；

(*c*)利用 EM 迭代算法获取初始状态概率、状态转移概率和输出符号概率。

6.7 应用举例

(2) 若有大规模分词和词性标注语料：有指导学习方法

咱们/rr 中国/ns 这么/rz 大{da4}/a 的{de5}/ud 一个/mq 多/a 民族/n 的{de5}/ud 国家/n 如果/c 不/df 团结/a ，/wd 就/d 不/df 可能/vu 发展/v 经济/n ，/wd 人民/n 生活/n 水平/n 也/d 就/d 不/df 可能/vu 得到/v 改善/vn 和{he2}/c 提高/vn 。/wj

可以从这些标注语料中抽取出所有的词汇和词类标记，并用最大似然估计方法计算各种概率。

6.7 应用举例

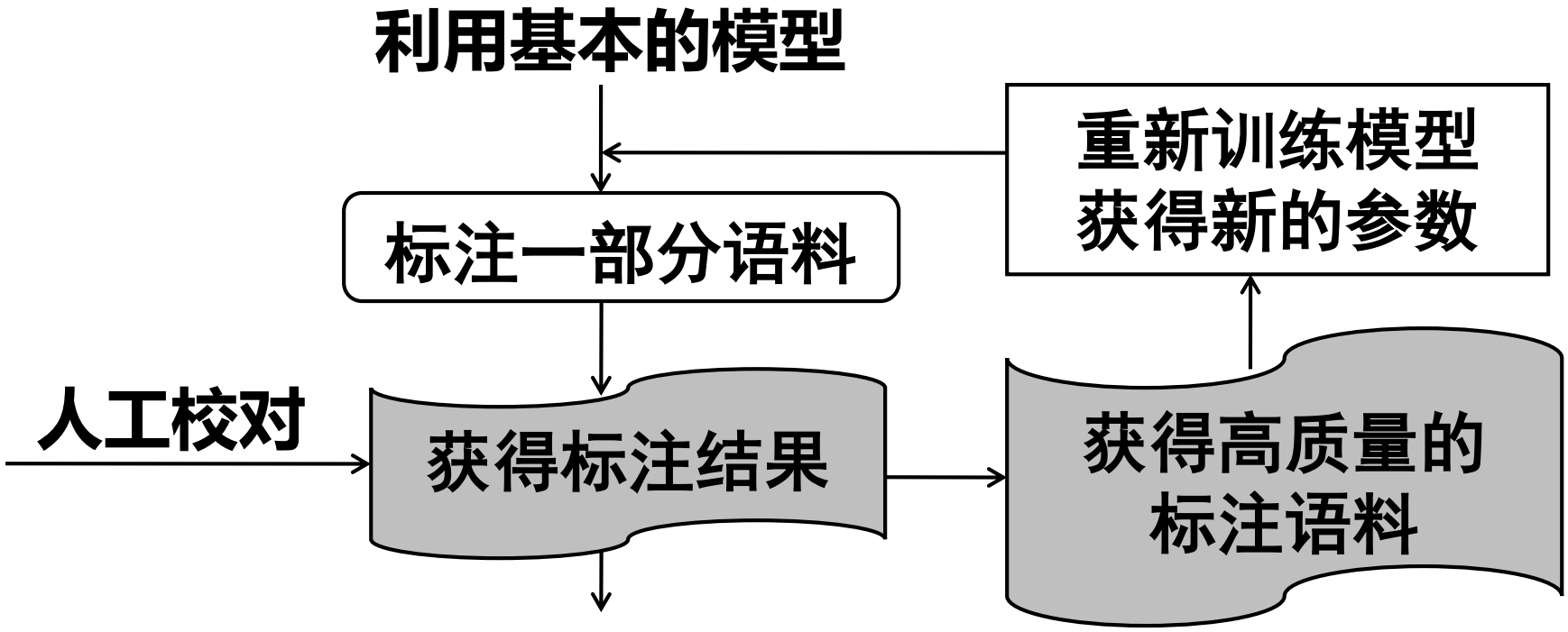
$$\bar{\pi}_{\text{pos}_i} = \frac{\text{POS}_i\text{出现在句首的次数}}{\text{所有句首的个数}}$$

$$\bar{a}_{ij} = \frac{\text{从词类POS}_i\text{转移到POS}_j\text{的次数}}{\text{所有从状态POS}_i\text{转移到另一POS(包括POS}_j\text{)的总数}}$$

$$\bar{b}_j(k) = \frac{\text{从状态POS}_j\text{输出词汇}w_k\text{的次数}}{\text{状态POS}_j\text{出现的总次数}}$$

6.7 应用举例

一般来说，需要通过错误驱动的机器学习方法修正模型的参数：



6.7 应用举例

◆问题2：如何获取观察序列？

—借助于其他工具，获得 *n*-best 的粗切分。

本地主叫通话时长1400分钟。

⟶ 本地/ 主叫/ 通话/ 时长/ 1400/ 分钟/ 。

本/ 地主/ 叫/ 通话/ 时/ 长/ 1400/ 分钟/ 。

本/ 地主/ 叫/ 通话/ 时长/ 1400/ 分钟/ 。

负责任 ⟶ 负/ 责任

负责/ 任

负/ 责/ 任

6.7 应用举例

◆**分词实验**：以“负责任”为例

利用部分《人民日报》语料。

词类 / 词	A	C	Q	NF	NG	NL	V	VN	总计
负责	4	0	0	0	0	0	177	50	231
任	0	4	11	59	2	4	98	0	178
其他	34469	25475	24232	11453	4550	25670	184488	42674	
总计	34473	25479	24243	11512	4552	25674	184763	42724	

6.7 应用举例

O_1= w_1 w_2 = 负责/ 任　　$p(O_1|\mu)=5.4\times10^{-6}$

O_2= w_1 w_2 = 负/ 责任　　$p(O_2|\mu)=9.3\times10^{-4}$

O_3= w_1 w_2 w_3= 负/ 责/ 任　　$p(O_3|\mu)=4.3\times10^{-6}$

$$p(O_2|\mu) > p(O_1|\mu) > p(O_3|\mu)$$

第二种切分结果可能性较大：负/ 责任

6.7 应用举例

◆**分词性能测试**：

(1) 封闭测试：《人民日报》1998年1月份的部分切分和标注语料，约占训练语料的1/10，计78396个词，含中国人名1273个。(人名识别前)准确率：90.34%。

(2) 开放测试：《人民日报》1998年2月份的部分切分和标注语料，也占训练语料的1/10，共82347个词，含中国人名2316个。(人名识别前)准确率：86.32%。

熊冬明，汉语自动分词和中文人名识别技术研究[硕士学位论文]，浙江大学，2006

6.7 应用举例

◆ **词性标注**：$\hat{Q} = \arg\max_{Q} p(Q \mid O, \mu)$

(1) 采用有指导的参数估计方法；

(2) 训练语料：北京大学标注的《人民日报》2000年1、2、4月份的语料；

(3) 封闭测试：2000年2月20-29日的标注语料，词性标注的精确率为：**95.16%**；

(4) 开放测试：2000年3月1-7日的语料，词性标注的精确率为：**88.45%**。

6.7 应用举例

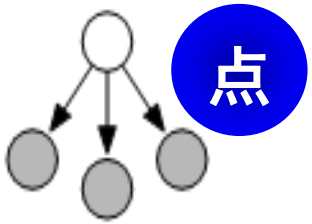
◆ **训练语料规模对模型参数的影响：**

选用北大标注的2000年《人民日报》语料作为训练数据。5个训练语料集大小不同: C1为2月份的；C2为1月及2月份的；C3为1、2和4月份的；C4为1、2、4和9月份的；C5为1、2、4、9和10月份五个月的。采用相同的测试集(2000年3月份前7天的语料)，观察词性标注的精确率变化：

语料	C1	C2	C3	C4	C5
精确率(%)	86.16	90.85	88.45	88.82	89.04

刘伟强，应用于词性标注的隐马尔可夫模型参数估计[硕士学位论文]，大连理工大学，2006

NLP中概率图模型的演变

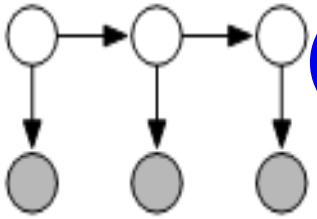


点

朴素贝叶斯

SEQUENCE

序列



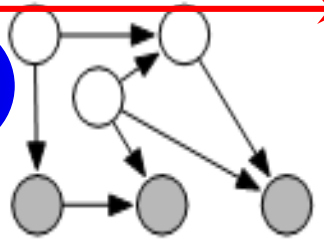
线

HMMs

GENERAL GRAPHS

一般图

图



生成式有向图

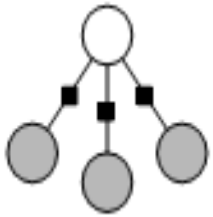
CONDITIONAL

在一定条件下

CONDITIONAL

在一定条件下

CONDITIONAL

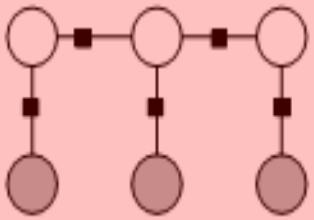


Logistic Regression

逻辑回归

SEQUENCE

序列

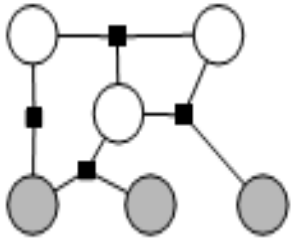


Linear-chain CRFs

线性链式CRFs

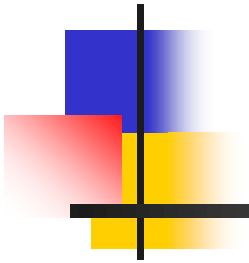
GENERAL GRAPHS

一般图



General CRFs

通用CRFs



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 ***X***，输出标识序列 ***Y***，通过计算 ***P*(*Y*|*X*)** 求解最优标注序列。

6.8 CRFs及其应用

◆ 定义

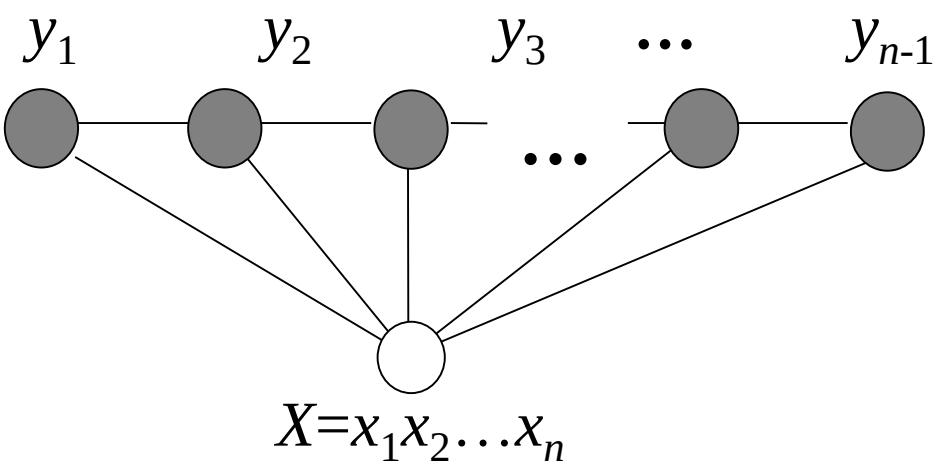
设 $G=(V, E)$ 为一个无向图，V为结点集合，E为无向边的集合，$Y = \{ Y_v \mid v \in V \}$，即V中每个结点对应于一个随机变量$Y_v$，其取值范围为可能的标记集合{y}。如果以观察序列X为条件，每个随机变量Y_v都满足以下马尔可夫特性：

$$p(Y_v \mid X, Y_w, w \neq v) = p(Y_v \mid X, Y_w, w \sim v) \quad \ldots (6\text{-}32)$$

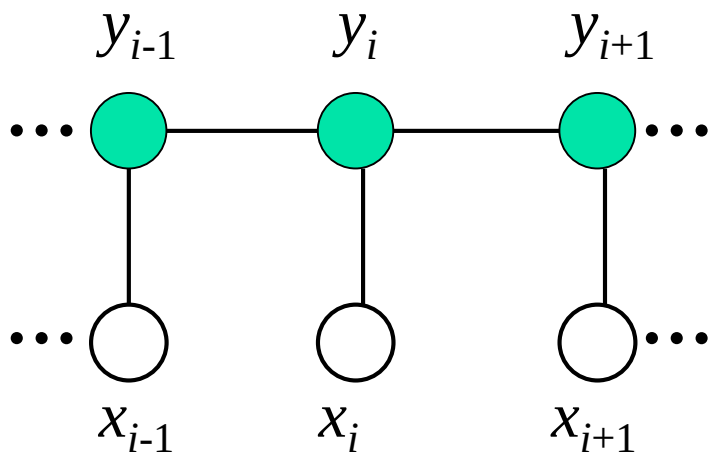
其中，$w \sim v$ 表示两个结点在图中是邻近结点。那么，(X, Y) 为一个条件随机场。

6.8 CRFs及其应用

理论上，只要在标记序列这描述了一定的条件独立性，G的图结构可以任意的。序列标注问题可以建模为简单的链式结构图，结点对应标记序列Y中的元素。如下图所示：



或者：



6.8 CRFs及其应用

在给定观察序列X 时，某个特定标记序列Y的概率可以定义为：

$$\exp\left(\sum_j \lambda_j t_j(y_{i-1}, y_i, X, i) + \sum_k \mu_k s_k(y_i, X, i)\right) \quad \ldots (6\text{-}33)$$

其中，$t_j(y_{i-1}, y_i, X, i)$ 是转移函数，表示对于观察序列X 的标注序列在 i 及 i-1 位置上标记的转移概率；

$s_k(y_i, X, i)$是状态函数，表示观察序列X 在 i 位置的标记概率；

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出。

6.8 CRFs及其应用

可以定义一组关于观察序列的 {0, 1} 二值特征 $b(X, i)$，表示训练样本中某些特征的分布，如

$$b(X,i)=\begin{cases}1 & \text{如果}X\text{的}i\text{位置为某个特定的词}\\ 0 & \text{否则}\end{cases}$$

转移函数可以定义为如下形式：

$$t_j(y_{i-1},y_i,X,i)=\begin{cases}b(X,i) & \text{如果}y_{i-1}\text{和}y_i\text{满足某种搭配条件}\\ 0 & \text{否则}\end{cases}$$

也可以把状态函数写成如下形式：

$$s(y_i,X,i)=s(y_{i-1},y_i,X,i)$$

6.8 CRFs及其应用

由此，特征函数可以统一表示为：

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

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 $s(y_{i-1},y_i,X,i)$ 或转移函数 $t(y_{i-1},y_i,X,i)$。

条件随机场定义的条件概率可以由下式给出：

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

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

6.8 CRFs及其应用

实现 CRFs 也需要解决如下三个问题：

①特征选取

②参数训练

③解码

定义和选取特征函数，利用GIS迭代算法选取λ权重。

请参阅前面第2章的最大熵模型

参考文献：

[1]J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proc.ICML'2001,* pages 282-289

[2]H. M. Wallach. Conditional Random Fields: An Introduction. *CIS Technical Report MS-CIS-04-21*, Univ. of Penn., 2004

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/

6.8 CRFs及其应用

◆ 应用举例

由字构词（基于字标注）的分词方法(Character-based tagging)

该方法由**N. Xue(薛念文)** 和 **S. Converse** 提出，首篇论文发表在2002年第一届国际计算语言学学会(ACL)汉语特别兴趣小组 SIGHAN (http://www.sighan.org/) 组织的汉语分词评测研讨会上[Xue and Converse, 2002]。

基本思想：将分词过程看作是字的分类问题：每个字在构造一个特定的词语时都占据着一个确定的构词位置(即词位)。一般情况下，每个字只有4个词位：词首(B)、词中(M)、词尾(E)和单独成词(S) 。

6.8 CRFs及其应用

(1) 上海/ 计划/ 到/ 本/ 世纪/ 末/ 实现/ 人均/ 国内/ 生产/ 总值/ 五千美元/ 。/

(2) 上/B 海/E 计/B 划/E 到/S 本/S 世/B 纪/E 末/S 实/B 现/E 人/B 均/E 国/B 内/E 生/B 产/E 总/B 值/E 五/B 千/M 美/M 元/E 。/S

在字标注过程中，对所有的字根据预定义的特征进行词位特征学习，获得一个概率模型，然后在待切分字串上，根据字与字之间的结合紧密程度，得到一个词位的分类结果，最后根据词位定义直接获得最终的分词结果。

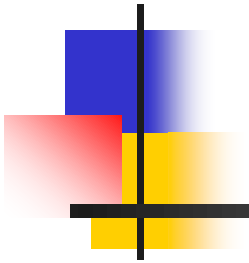
6.8 CRFs及其应用

上/B **海**/E **计**/B **划**/E **到 本 世 纪 ……**

B, E, M, S ?

- 当前字的前后 n 个字 (如 $n=\pm 2$)
- 当前字左边字的标记
- 当前字在词中的位置

……



- HMM和CRF用于分词的进一步示例

基于HMM的中文分词方法

- 给定一个观察序列(句子)$X=x_1x_2\ldots x_t\ldots x_T$, 其中$x_t$是一个字、词等文字单元
- 假设X的状态序列(如词的开始符、词的结束符等)为Y
  - $y_t(i)$有M个状态　$Y = y_1(i)y_2(i)...y_t(i)...y_T(i),\ 1 \le i \le M$

$$Y^* = \arg\max_Y P(Y \mid X) = \arg\max_Y \frac{P(Y,X)}{P(X)} \propto \arg\max_Y P(X \mid Y)P(Y)$$

1) 独立性假设 $\Rightarrow P(X \mid Y) = \prod_{t=1}^{T} P(x_t \mid y_t)$

2) 马尔可夫(一阶)假设：$P(y_t \mid y_{t-1}y_{t-2}\ldots y_1) = P(y_t \mid y_{t-1}) \Rightarrow P(Y)=P(y_1)\prod_{t=2}^{T} P(y_t \mid y_{t-1})$

$$Y^* = \arg\max_Y P(y_1)\prod_{t=1}^{T} P(x_t \mid y_t)\prod_{t=2}^{T} P(y_t \mid y_{t-1})$$

$$= \arg\max_Y P(y_1)P(x_1 \mid y_1)\prod_{t=2}^{T}[P(y_t \mid y_{t-1})P(x_t \mid y_t)]$$

$$= \arg\max_Y [P(y_1)P(x_1 \mid y_1)][P(y_2 \mid y_1)P(x_2 \mid y_2)][P(y_3 \mid y_2)P(x_3 \mid y_3)]\ldots$$

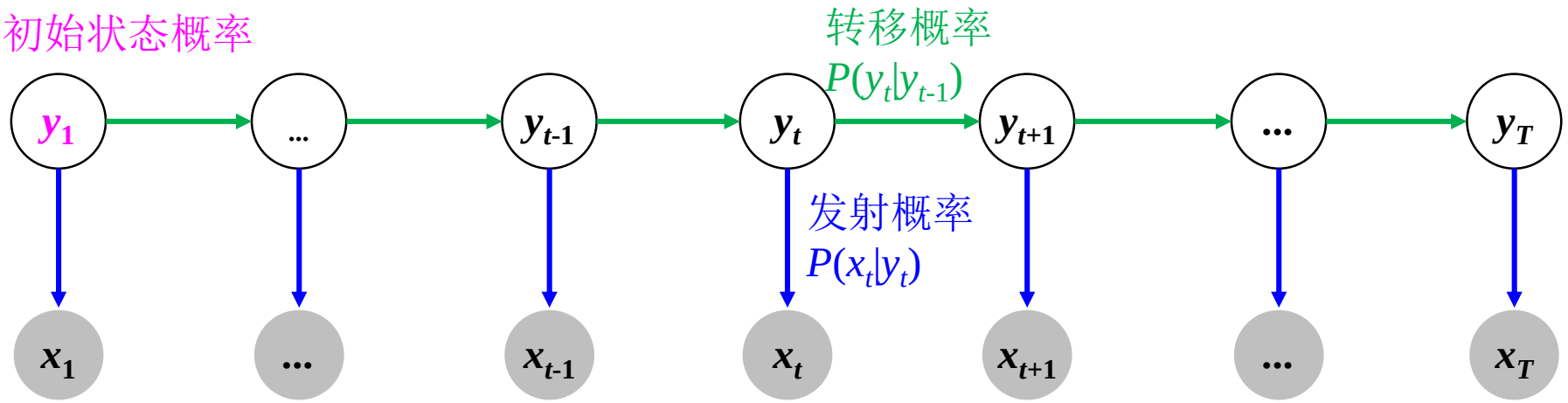
$$Y^* = \arg\max_Y P(y_1)\prod_{t=1}^{T} P(x_t \mid y_t)\prod_{t=2}^{T} P(y_t \mid y_{t-1})$$

$$= \arg\max_Y P(y_1)P(x_1 \mid y_1)\prod_{t=2}^{T}[P(y_t \mid y_{t-1})P(x_t \mid y_t)]$$

$$= \arg\max_Y [P(y_1)P(x_1 \mid y_1)][P(y_2 \mid y_1)P(x_2 \mid y_2)][P(y_3 \mid y_2)P(x_3 \mid y_3)]...$$

- 发射概率 $P(x_t|y_t)$
- 转移概率 $P(y_t|y_{t-1})$
  - Y共有M个状态 $\sum_{i=1}^{M} P(y_t(i) \mid y_{t-1}) = 1$
- 初始状态概率 $P(y_1(i))$ (i=1...M)
- HMM包括隐层状态Y，观察序列X，状态转移概率A，符号发射概率B，和初始状态概率分布π。
- HMM表示为μ={A,B, π}，参数通过训练集来学习获得。

HMM可以用有向图模型来表示，因为states (Y)与observations (X)之间存在着明显的依赖关系。



中文分词

- 输入观察序列X：南京市长江大桥
- 状态集合$Y(i)$ ={B, M, E, S}：

状态Y	Begin	Middle	End	Single
解释	词的开始字	词的中间字	词的结束字	单字成词
示例	南京的“南”	乒乓球的“乓”	南京的“京”	你

- 输出状态序列Y：BMEBMME

Viterbi 搜索算法

给定模型μ和观察序列$X=x_1x_2\ldots x_t\ldots x_T$的条件下求概率最大的状态序列$Y=y_1y_2\ldots y_t\ldots y_T$：

$$Y^* = \arg\max_{Y} P(Y \mid X, \mu)$$

Viterbi 算法: 动态搜索最优状态序列。

定义：Viterbi 变量$\delta_t(y(i))$**是在时间** t **时，模型沿着某一条路径到达状态** $y(i)$**，并输出观察序列** $X=x_1x_2\ldots x_t$**的最大概率：**

$$\delta_t(y(i)) = \max_{y_1y_2\cdots y_{t-1}} P(y_1y_2\cdots y_{t-1}y_t(i), x_1x_2\cdots x_{t-1}x_t \mid \mu)$$

从状态$y_t(j)$转移到状态$y_{t+1}(i)$的概率

t+1步状态$y_{t+1}(i)$发射观察值x_{t+1}的概率

$$\delta_{t+1}(y(i)) = \max_{y(j)}\{\delta_t(y(j)) \cdot P(y_{t+1}(i) \mid y_t(j))\} \cdot P(x_{t+1} \mid y_{t+1}(i))$$

where $y(i), y(j) \in \{B, M, E, S\}$

递归计算：

算法描述

(1)初始化：$\delta_1(i) = y_1(i)P(x_1 \mid y_1(i)), \quad 1 \le i \le M$

概率最大的**路径**变量：$\psi_1(i) = 0$

(2)递推计算：

$$\delta_t(i) = \max_{1 \le j \le M}\{\delta_{t-1}(j) \cdot P(y_t(i) \mid y_{t-1}(j))\} \cdot P(x_t \mid y_t(i)), 2 \le t \le T, 1 \le j, i \le M$$

第t步从所有可能的$j \rightarrow i$的M条路径中取最大值

$$\psi_t(i) = \arg\max_{1 \le j \le M}[\delta_{t-1}(j) \cdot P(y_t(i) \mid y_{t-1}(j))] \cdot P(x_t \mid y_t(i)), \quad 2 \le t \le T, \quad 1 \le i, j \le M$$

arg是从1...t累积概率最大的路径

(3)结束：

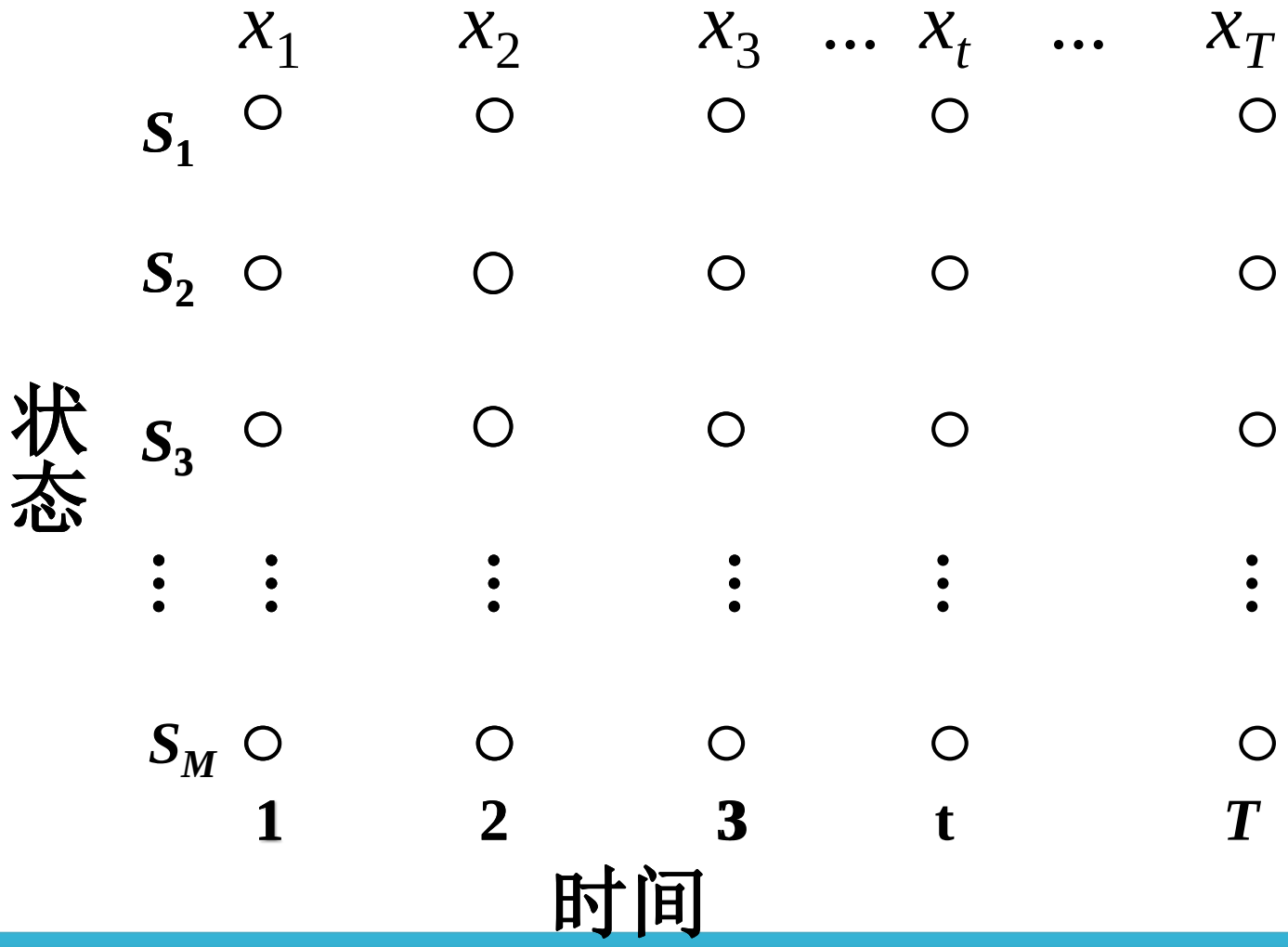
$$Y = \arg\max_{1 \le i \le M}[\delta_T(i)], \qquad P(Y) = \max_{1 \le i \le M} \delta_T(i)$$

(4)通过回溯得到路径（状态序列）：

$$y_t = \psi_{t+1}(y_{t+1}), \quad t = T-1,\ T-2,\ \cdots,\ 1$$

算法的时间复杂度：$O(M^2T)$

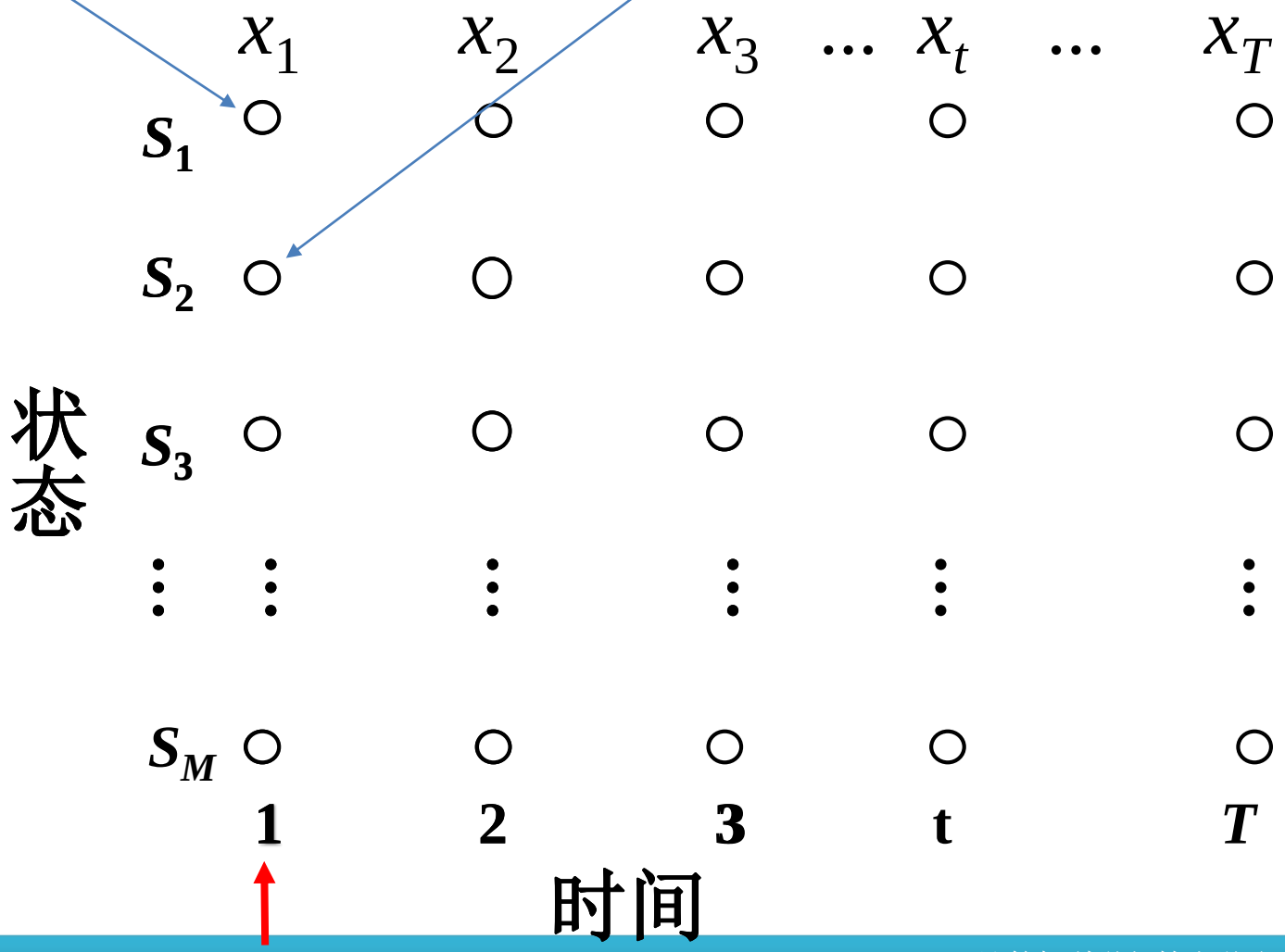
图解Viterbi搜索过程



$$\delta_1(i) = y_1(i)P(x_1 \mid y_1(i)), \quad 1 \le i \le M$$

$$\delta_1(1) = y_1(1)P(x_1 \mid y_1(1)) \quad \delta_1(2) = y_1(2)P(x_1 \mid y_1(2))$$

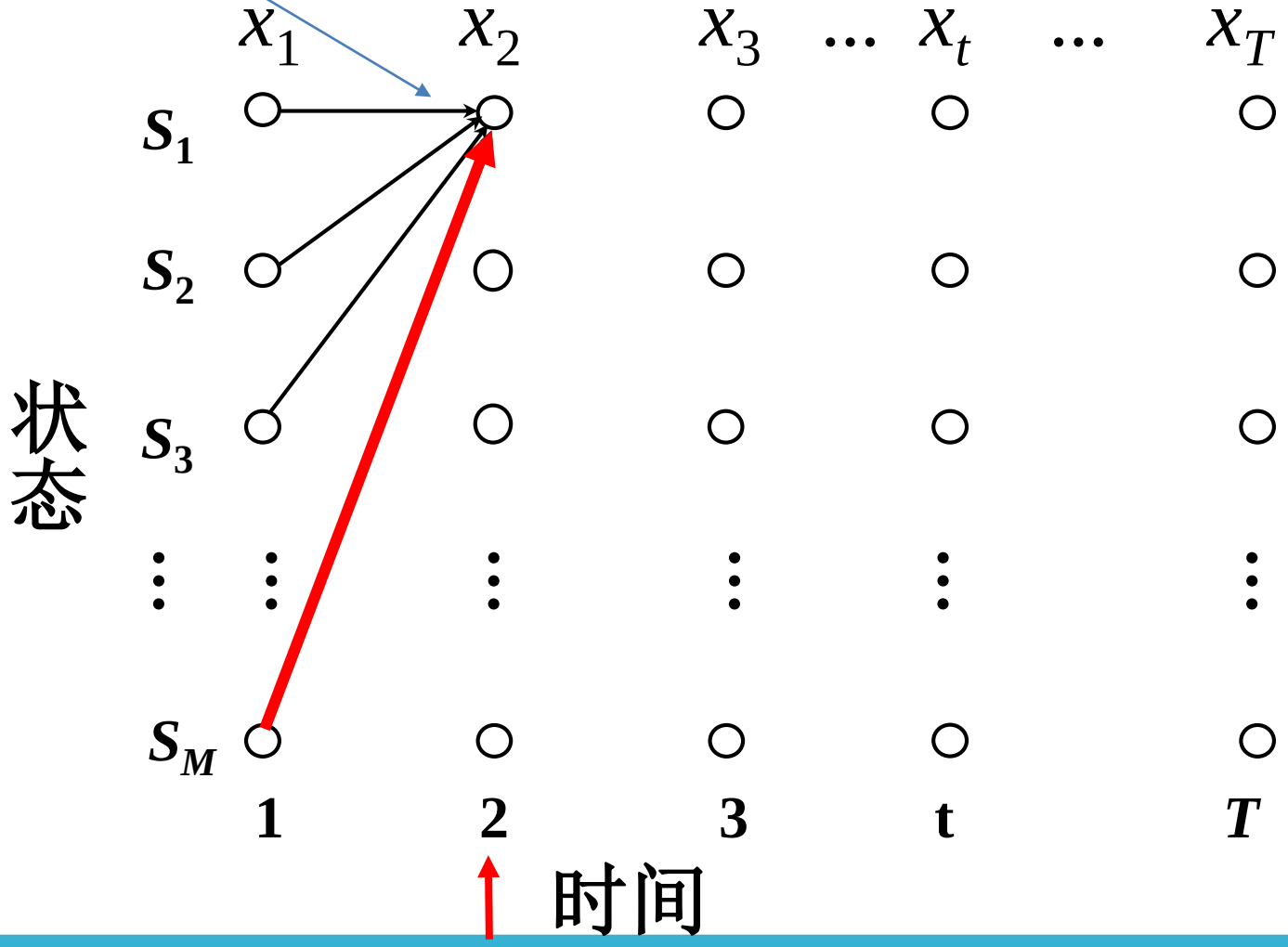
图解 Viterbi 搜索过程



$$\delta_t(i) = \max_{1\le j\le M}\{\delta_{t-1}(j)\cdot P(y_t(i)\mid y_{t-1}(j))\}\cdot P(x_t\mid y_t(i)), 2\le t\le T, 1\le j,i\le M$$

$$\delta_2(1) = \max_{1\le j\le M}\{\delta_1(j)\cdot P(y_2(1)\mid y_1(j))\}\cdot P(x_2\mid y_2(1))$$

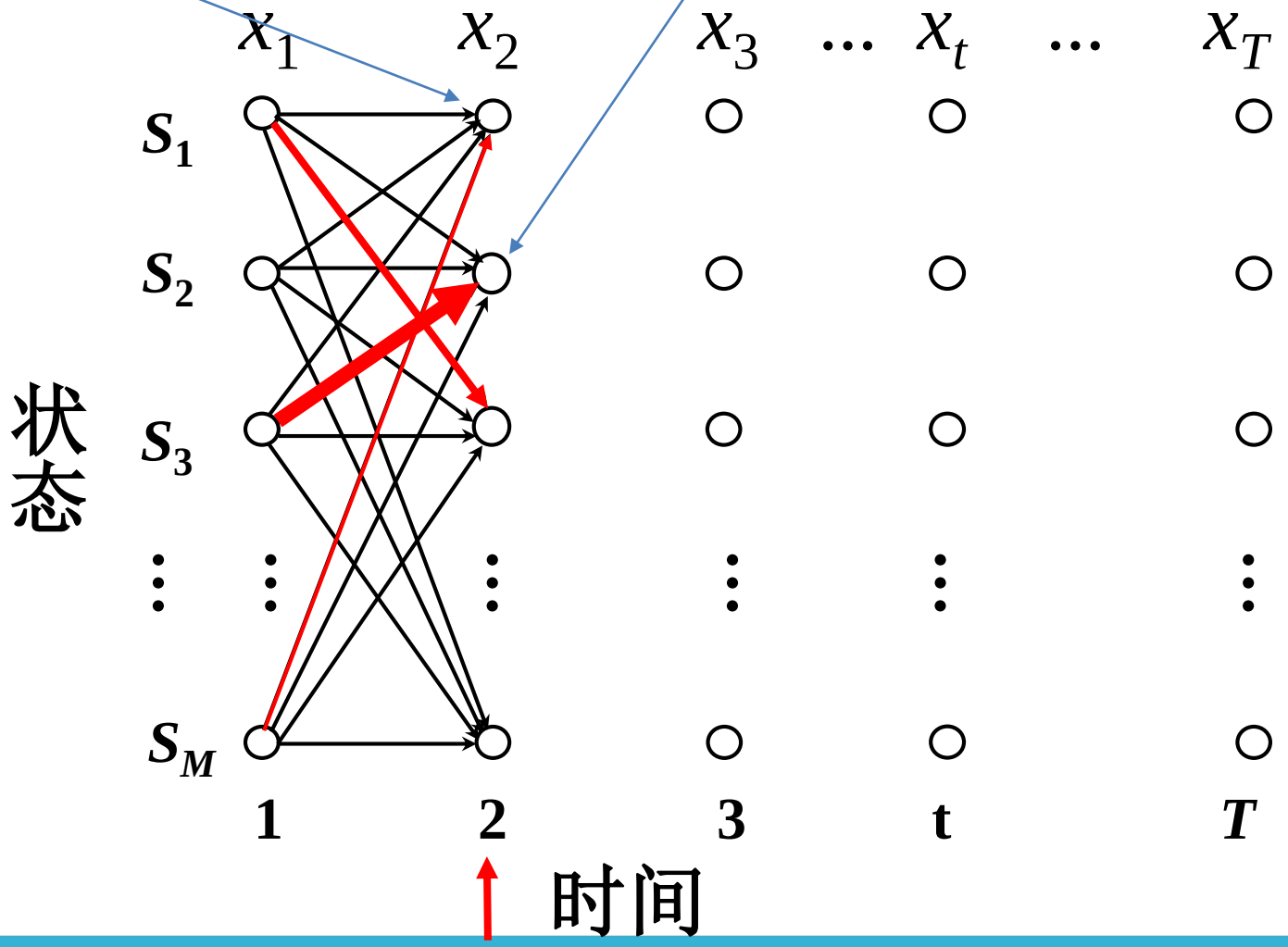
图解Viterbi搜索过程



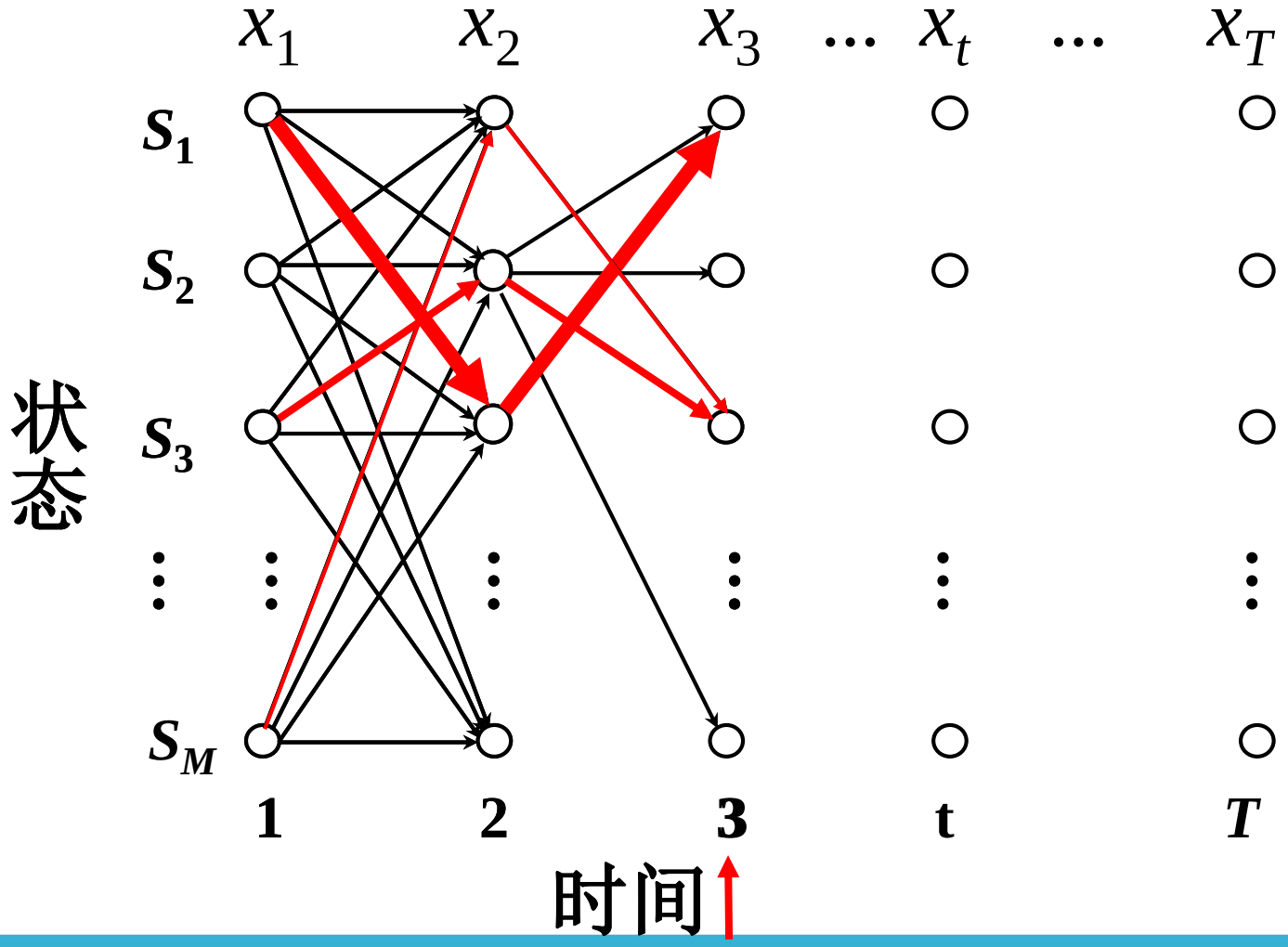
$$\delta_2(1) = \max_{1 \le j \le M} \{\delta_1(j) \cdot P(y_2(1) \mid y_1(j))\} \cdot P(x_2 \mid y_2(1))$$

$$\delta_2(2) = \max_{1 \le j \le M} \{\delta_1(j) \cdot P(y_2(2) \mid y_1(j))\} \cdot P(x_2 \mid y_2(2))$$

图解 Viterbi 搜索过程



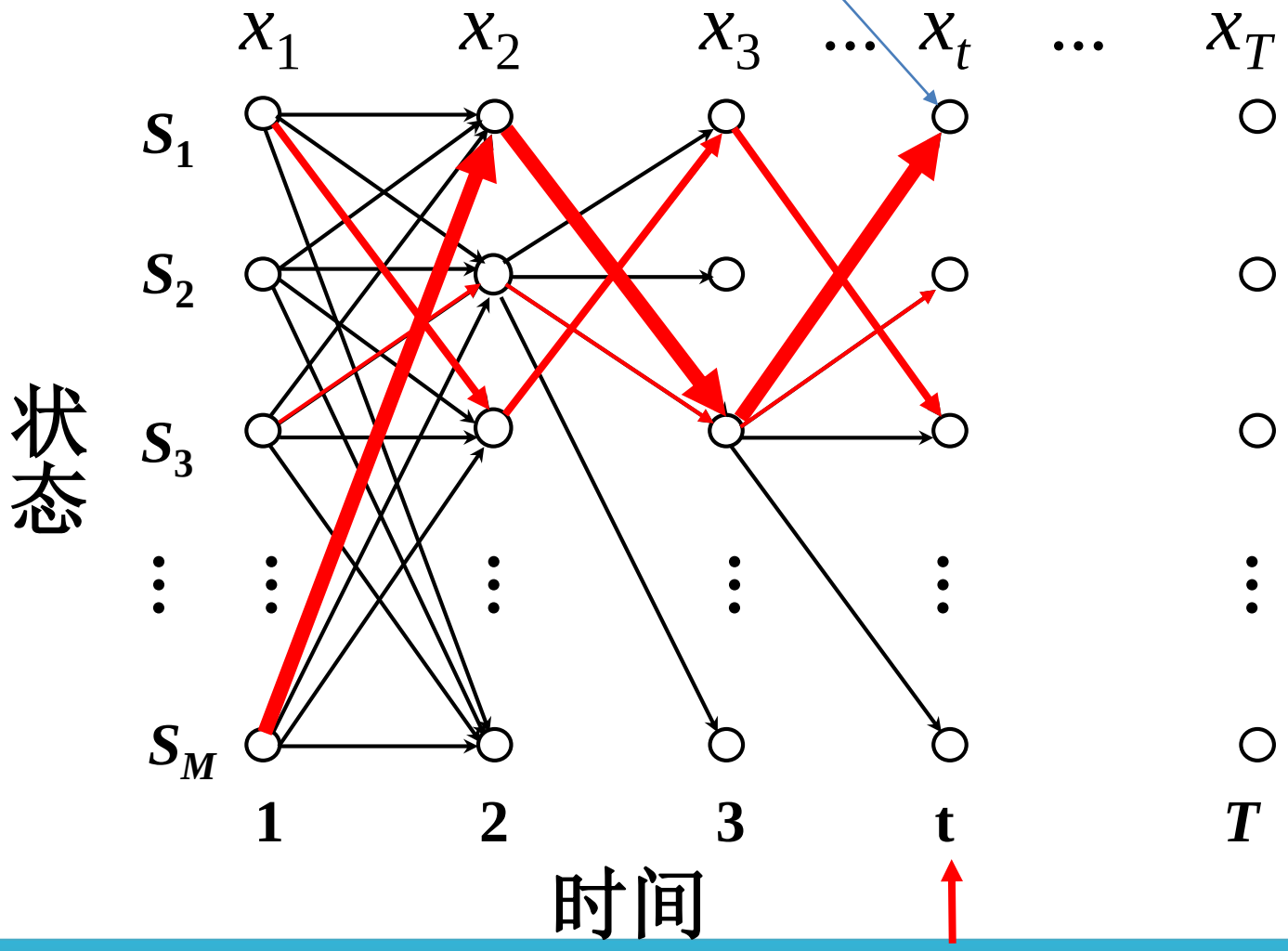
图解Viterbi搜索过程



$$\delta_t(i) = \max_{1\le j\le M}\{\delta_{t-1}(j)\cdot P(y_t(i)\mid y_{t-1}(j))\}\cdot P(x_t\mid y_t(i)), 2\le t\le T, 1\le j,i\le M$$

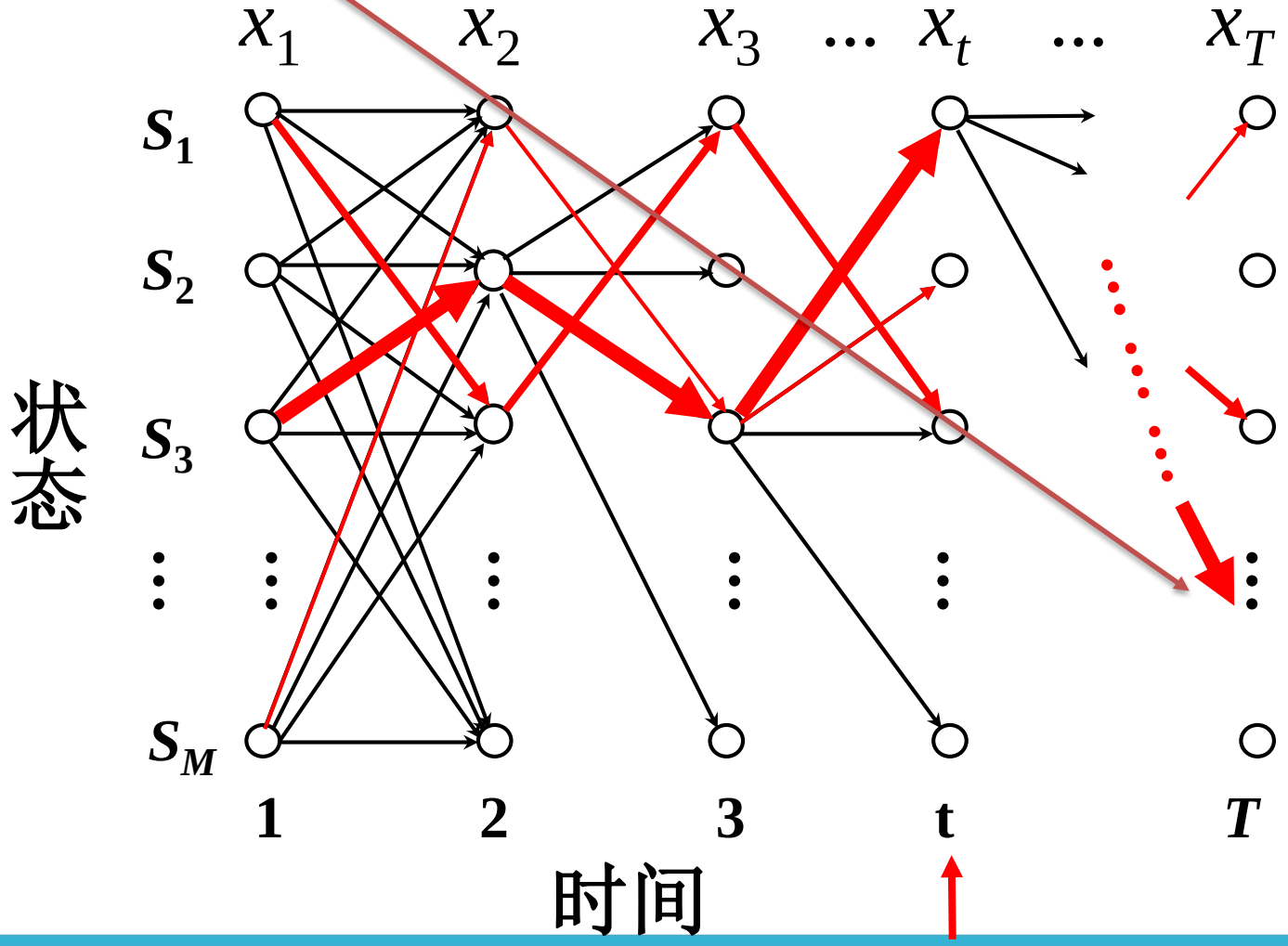
$$\delta_t(1) = \max_{1\le j\le M}\{\delta_{t-1}(j)\cdot P(y_t(1)\mid y_{t-1}(j))\}\cdot P(x_t\mid y_t(1))$$

图解Viterbi搜索过程



找到概率最大值 $\max_{1\le i\le M}\delta_T(i)$，然后通过回溯得到路径 $Y=\arg\max_{1\le i\le M}[\delta_T(i)]$

图解 Viterbi 搜索过程



- 输入观察序列X：南京市长江大桥
- 状态集合：

状态Y	Begin	Middle	End	Single
解释	词的开始字	词的中间字	词的结束字	单字成词
示例	南京的“南”	乒乓球的“乓”	南京的“京”	你

- 输出状态序列Y
- 给定HMM模型：{A,B, π}

字	南	京	市	长	江	大	桥
状态Begin	0.3	0.3	0.1	0.2	0.1	0.1	0.2
状态Middle	0.2	0.3	0.2	0.3	0.3	0.2	0.1
状态End	0.1	0.1	0.3	0.2	0.1	0.2	0.6
状态Single	0.2	0.2	0.2	0.1	0.4	0.2	0.1

南B 京M 市E 长B 江M 大M 桥E ⟹ 南京市， 长江大桥

条件随机场模型(Conditional Random Fields，CRF)

- 对汉字进行标注训练，不仅考虑了词语出现的频率，还考虑了上下文，具备较好的学习能力，对歧义词和未登录词的识别都具有良好的效果。

$$Y^* = \arg\max_Y P(Y \mid X;\lambda) = \frac{1}{Z(X)} \exp[\sum_j \lambda_j F_j(Y,X)]$$

$$Z(X) = \sum_Y \exp[\sum_j \lambda_j F_j(Y,X)]$$

$$F_j(Y,X) = \sum_{t=1}^{T} f_j(y_{t-1}, y_t, X, t)$$

- $F_j()$第j个特征函数，可以表示状态特征函数，或者状态转移函数
- λ_j第j个特征特征函数的权重
- $Z(X)$归一化因子

状态Y	Begin	Middle	End	Single
解释	词的开始字	词的中间字	词的结束字	单字成词
示例	南京的“南”	乒乓球的“乓”	南京的“京”	你

特征函数	释义
$f_1(x_t, y_t)$	状态特征函数：字x_t作为状态y_t出现的概率
$f_2(y_{t-1}, y_t)$	状态转移函数：4个状态，那么就是4*4的状态转移概率矩阵
$f_3(x_{t-1}, x_t, y_t)$ or $f_4(x_t, x_{t+1}, y_t)$	在状态y_t下上下文字转移概率(前后字)

$$P(y_t \mid x_t, \lambda) = \lambda_1 f_1(x_t, y_t) + \lambda_3 f_3(x_{t-1}, x_t, y_t) + \lambda_4 f_4(x_t, x_{t+1}, y_t) + \lambda_2 \max_{y'_{t-1} \in \{B,M,E,S\}} \{ f_2(y'_{t-1}, y_t) P(y'_{t-1} \mid x_{t-1}, \lambda) \}$$

字	南	京	市	长	江	大	桥
状态Begin							
状态Middle							
状态End							
状态Single							

$$P(y_t \mid x_t, \lambda) = \lambda_1 f_1(x_t, y_t) + \lambda_3 f_3(x_{t-1}, x_t, y_t) + \lambda_4 f_4(x_t, x_{t+1}, y_t) + \lambda_2 \max_{y'_{t-1} \in \{B,M,E,S\}} \{ f_2(y'_{t-1}, y_t) P(y'_{t-1} \mid x_{t-1}, \lambda) \}$$

字	南	京	市	长	江	大	桥
状态Begin	0.5						
状态Middle	0.2						
状态End	0.2						
状态Single	0.3						

$$P(B \mid 南, \lambda)$$
$$= \lambda_1 f_1(南, B) + \lambda_3 f_3(\text{null}, 南, B) + \lambda_4 f_4(南, 京, B) + \lambda_2 * 0$$
$$= \lambda_1 f_1(南, B) + \lambda_3 * 0 + \lambda_4 f_4(南, 京, B) + \lambda_2 * 0$$

$$P(M \mid 南, \lambda)$$
$$= \lambda_1 f_1(南, M) + \lambda_3 f_3(\text{null}, 南, M) + \lambda_4 f_4(南, 京, M) + \lambda_2 * 0$$
$$= \lambda_1 f_1(南, M) + \lambda_3 * 0 + \lambda_4 f_4(南, 京, M) + \lambda_2 * 0$$

$$P(y_t \mid x_t, \lambda) = \lambda_1 f_1(x_t, y_t) + \lambda_3 f_3(x_{t-1}, x_t, y_t) + \lambda_4 f_4(x_t, x_{t+1}, y_t) + \lambda_2 \max_{y'_{t-1} \in \{B,M,E,S\}} \{ f_2(y'_{t-1}, y_t) P(y'_{t-1} \mid x_{t-1}, \lambda) \}$$

字	南	京	市	长	江	大	桥
状态Begin	0.5	0.2					
状态Middle	0.2	0.4					
状态End	0.2	0.1					
状态Single	0.3	0.2					

$$P(B \mid 京, \lambda)$$

$$= \lambda_1 f_1(京, B) + \lambda_3 f_3(南, 京, B) + \lambda_4 f_4(京, 市, B)$$

$$+\lambda_2 \max_{y' \in \{B,M,E,S\}} \{ f_2(B,B)P(B \mid 南),\ f_2(M,B)P(M \mid 南),\ f_2(E,B)P(E \mid 南),\ f_2(S,B)P(S \mid 南) \}$$

$$= \lambda_1 f_1(京, B) + \lambda_3 f_3(南, 京, B) + \lambda_4 f_4(京, 市, B) +$$

$$\lambda_2 * \max_{y' \in \{B,M,E,S\}} \{ f_2(B,B)0.5,\ f_2(M,B)0.2,\ f_2(E,B)0.2,\ f_2(S,B)0.3 \}$$

$$P(M \mid 京, \lambda)$$

$$= \lambda_1 f_1(京, M) + \lambda_3 f_3(南, 京, M) + \lambda_4 f_4(京, 市, \mathrm{M})$$

$$+\lambda_2 \max_{y' \in \{B,M,E,S\}} \{ f_2(B,M)P(B \mid 南),\ f_2(M,M)P(M \mid 南),\ f_2(E,M)P(E \mid 南),\ f_2(S,M)P(S \mid 南) \}$$

$$P(y_t \mid x_t, \lambda) = \lambda_1 f_1(x_t, y_t) + \lambda_3 f_3(x_{t-1}, x_t, y_t) + \lambda_4 f_4(x_t, x_{t+1}, y_t) + \lambda_2 \max_{y'_{t-1} \in \{B,M,E,S\}} \{ f_2(y'_{t-1}, y_t) P(y'_{t-1} \mid x_{t-1}, \lambda) \}$$

字	南	京	市	长	江	大	桥
状态Begin	0.5	0.3	0.1	0.1	0.1	0.1	0.2
状态Middle	0.2	0.4	0.2	0.3	0.2	0.3	0.1
状态End	0.2	0.1	0.4	0.2	0.1	0.4	0.6
状态Single	0.3	0.2	0.3	0.1	0.2	0.2	0.1

南B 京M 市E 长B 江M 大M 桥E ⇒ 南京市， 长江大桥

本章小结

◆HMM 的构成：

①状态数 ②输出符号数 ③初始状态的概率分布 ④状态转移的概率 ⑤输出概率

◆HMM 的三个基本问题：

(1)快速计算给定模型的观察序列概率: 前/后向算法

(2)求最优状态序列：Viterbi 算法

(3) 参数估计：Baum-Welch 算法

◆模型实现中需要注意的问题：小数溢出

◆条件随机场(CRFs)

Thanks