自然语言处理

Natural Language Processing

权小军 教授

中山大学数据科学与计算机学院

quanxj3@mail.sysu.edu.cn

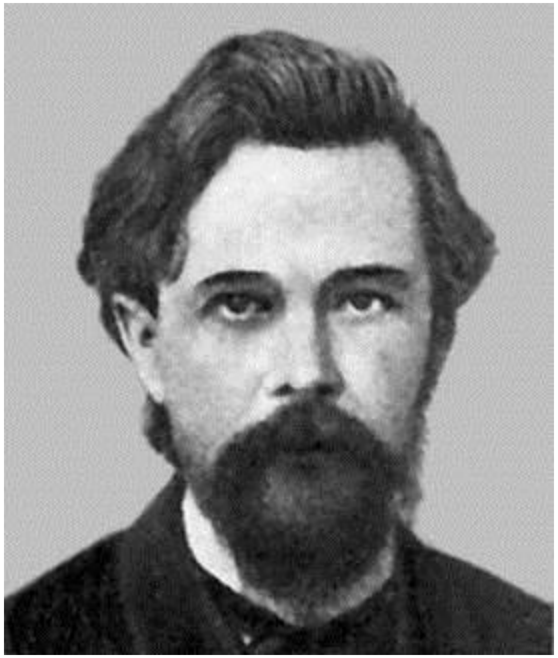
Lecture 5: 马尔可夫模型

Lecture 5.1 概述

马尔可夫模型

- 马尔可夫(Markov)(1856.6.14 ～ 1922.7.20)

前苏联数学家。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。



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, …, t-1) 的状态，该概率为：

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

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一阶马尔可夫链：

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

马尔可夫模型

●假设2:

如果只考虑公式(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：

$$p(q_t = S_j \mid q_{t-1} = S_i) = a_{ij}, \quad 1 \le i, j \le N \quad \ldots (2)$$

该随机过程称为**马尔可夫模型(Markov Model)**。

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：

$$a_{ij} \ge 0 \qquad \ldots (3)$$

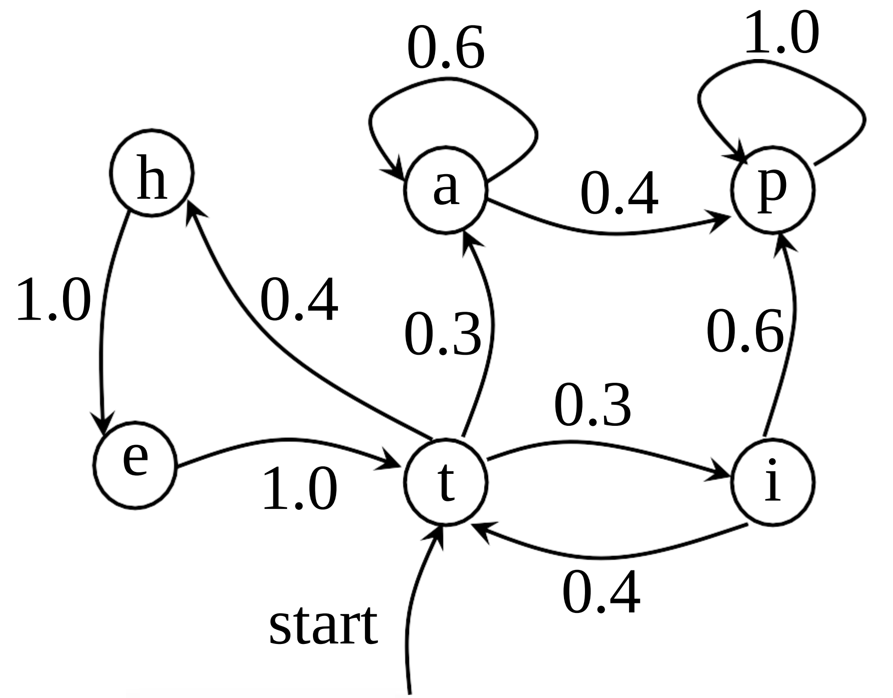
$$\sum_{j=1}^{N} a_{ij} = 1 \qquad \ldots (4)$$

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。



马尔可夫模型

状态序列 $S_1, \ldots, S_T$ 的概率：

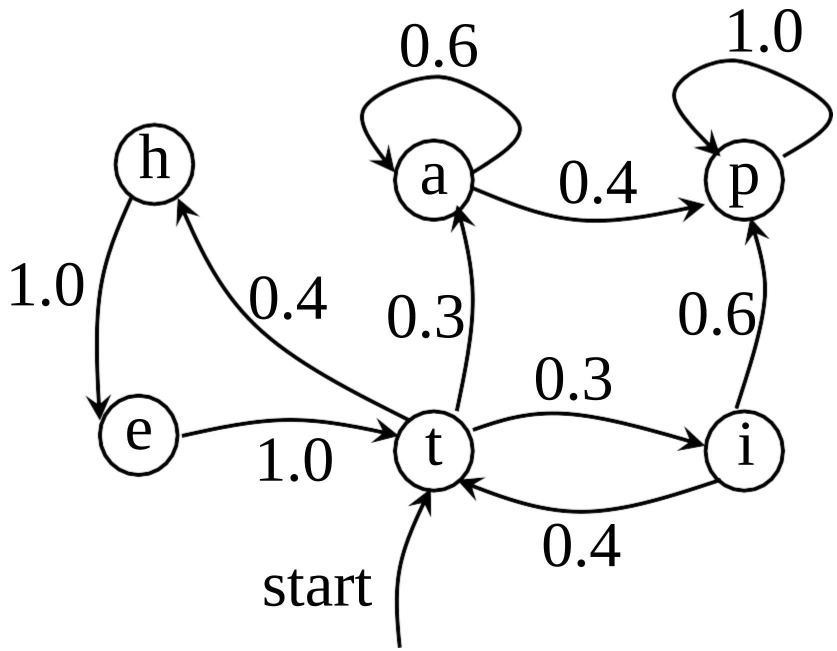
$$p(S_1,\cdots,S_T) = p(S_1)\times p(S_2 \mid S_1)\times p(S_3 \mid S_1,S_2)\times\cdots\times p(S_T \mid S_1,\cdots,S_{T-1})$$

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

$$= \pi_{S_1}\prod_{t=1}^{T-1} a_{S_t S_{t+1}} \qquad \ldots (5)$$

其中， $\pi_i = p(q_1 = S_i)$为初始状态的概率。

马尔可夫模型



$$p(\mathrm{t},\mathrm{i},\mathrm{p}) = p(S_1=\mathrm{t}) \times p(S_2=\mathrm{i} \mid S_1=\mathrm{t}) \times p(S_3=\mathrm{p} \mid S_2=\mathrm{i})$$
$$= 1.0 \times 0.3 \times 0.6$$
$$= 0.18$$

Lecture 5.2 隐马尔可夫模型

隐马尔可夫模型

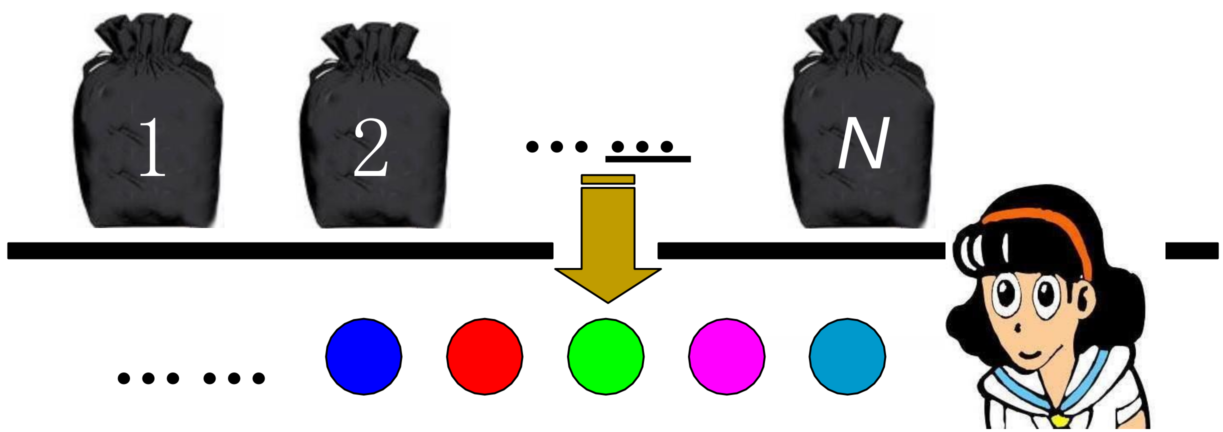
◆隐马尔可夫模型(Hidden Markov Model,HMM)

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，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。

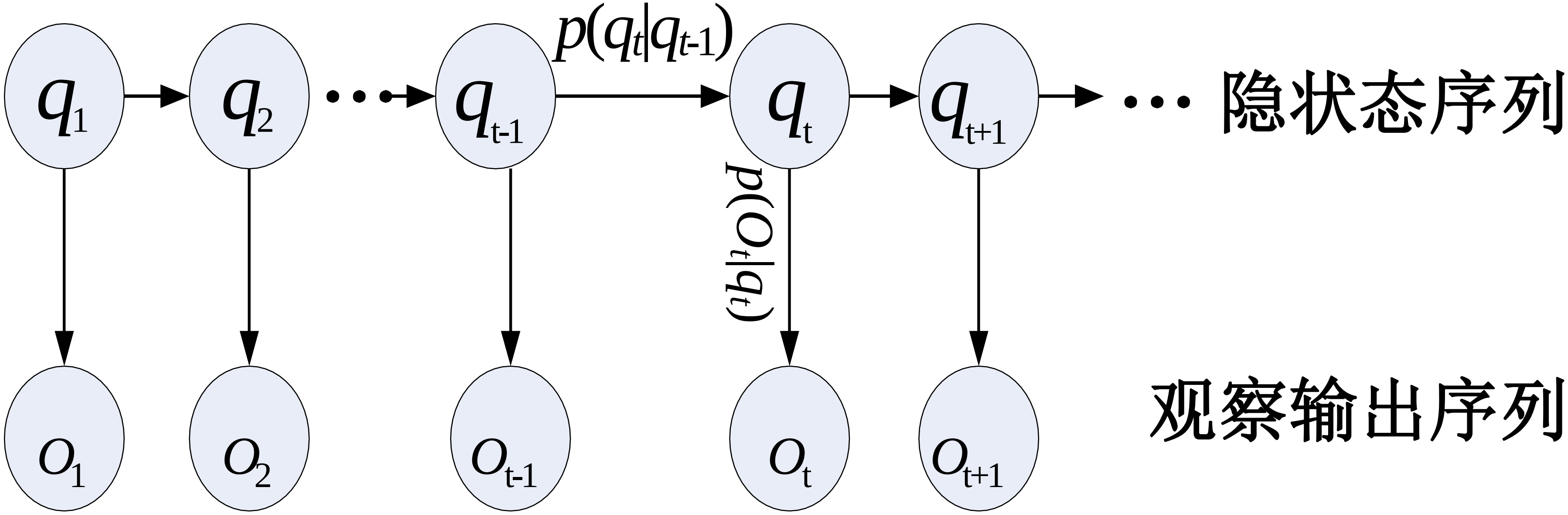
隐马尔可夫模型

<u>例如</u>：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。



隐马尔可夫模型



HMM 图解

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)

隐马尔可夫模型

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，

$$
\begin{cases}
a_{ij} = p(q_{t+1} = S_j \mid q_t = S_i), \quad 1 \leq i, j \leq N \\
a_{ij} \geq 0 \\
\sum_{j=1}^{N} a_{ij} = 1
\end{cases}
\qquad \ldots (6)
$$

隐马尔可夫模型

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，

$$
\begin{cases}
b_j(k) = p(O_t = v_k \mid q_t = S_j), \quad 1 \le j \le N, \; 1 \le k \le M \\
b_j(k) \ge 0 \\
\sum_{k=1}^{M} b_j(k) = 1
\end{cases}
\qquad \ldots (7)
$$

隐马尔可夫模型

初始状态的概率分布为：$\pi = \pi_i$，其中，

$$\begin{cases} \pi_i = p(q_1 = S_i), \quad 1 \le i \le N \\ \pi_i \ge 0 \\ \sum_{i=1}^{N} \pi_i = 1 \end{cases} \qquad \ldots (8)$$

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。

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 $O=O_1O_2\ldots O_T$：

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。

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 $O=O_1O_2 \ldots O_T$ 的情况下，怎样快速计算概率$p(O|\mu)$？

隐马尔可夫模型

◆三个问题：

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

(2) 在给定模型 μ=(A, B, π) 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 $Q = q_1 q_2 \ldots q_T$，使得该状态序列“最好地解释”观察序列?

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 $O=O_1O_2 \ldots O_T$ 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 $O=O_1O_2 \ldots O_T$ 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?

(3) 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? 即如何调节模型的参数，使得$p(O|\mu)$ 最大？

Lecture 5.3 前向算法

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$O = O_1O_2 \ldots O_T$，快速计算 $p(O|\mu)$：

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$O = O_1O_2 \ldots O_T$，快速计算 $p(O|\mu)$：

$$p(O|\mu)=\sum_Q p(O,Q|\mu) = \sum_Q p(Q|\mu)\times p(O|Q,\mu) \qquad \ldots (9)$$

前向算法

◆ 问题1：快速计算观察序列概率$p(O|\mu)$

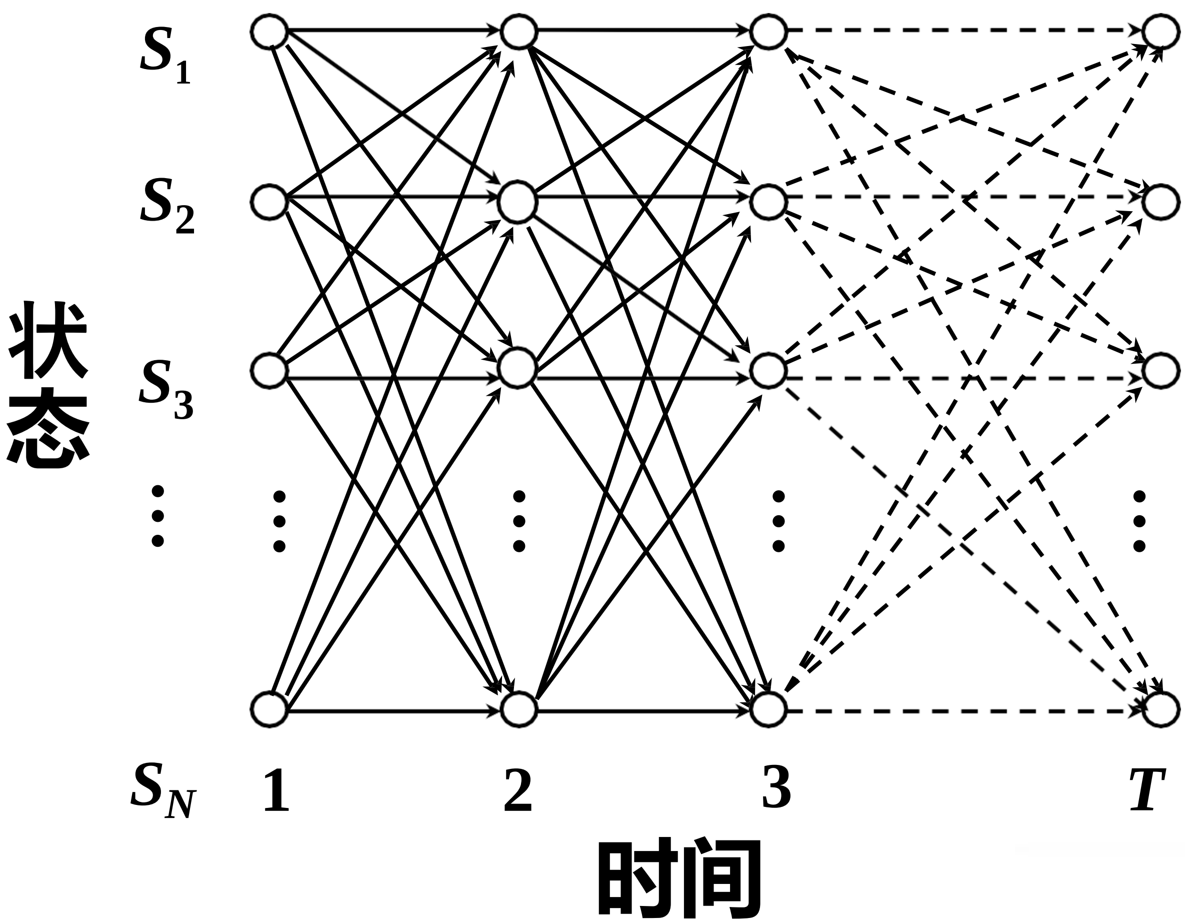
给定模型 $\mu=(A, B, \pi)$ 和观察序列$O = O_1O_2 \ldots O_T$，快速计算 $p(O|\mu)$：

$$p(O|\mu) = \sum_{Q} p(O,Q|\mu) = \sum_{Q} \boxed{p(Q|\mu)} \times \boxed{p(O|Q,\mu)} \qquad \ldots (9)$$

$$p(Q|\mu) = \pi_{q_1} \times a_{q_1 q_2} \times a_{q_2 q_3} \times \cdots \times a_{q_{t-1} q_T} \qquad \ldots (10)$$

$$p(O|Q,\mu) = b_{q_1}(O_1) \times b_{q_2}(O_2) \times \cdots \times b_{q_T}(O_T) \qquad \ldots (11)$$

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。

前向算法

- 解决办法：动态规划

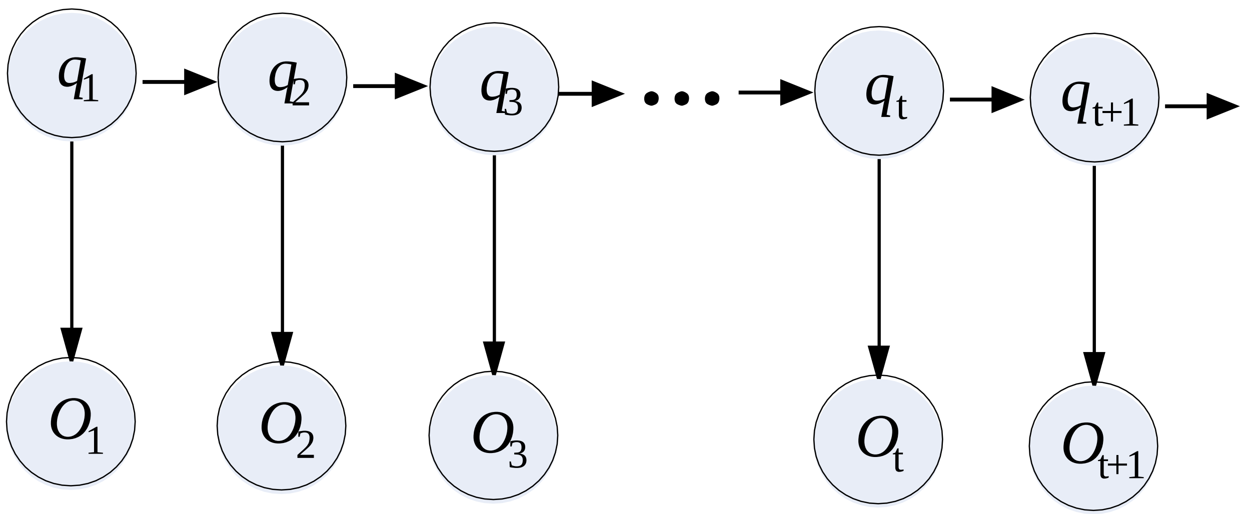
  前向算法(The forward procedure)

- 基本思想：定义前向变量 $\alpha_t(i)$：

$$\alpha_t(i) = p(O_1 O_2 \cdots O_t, q_t = S_i \mid \mu) \qquad \ldots(12)$$

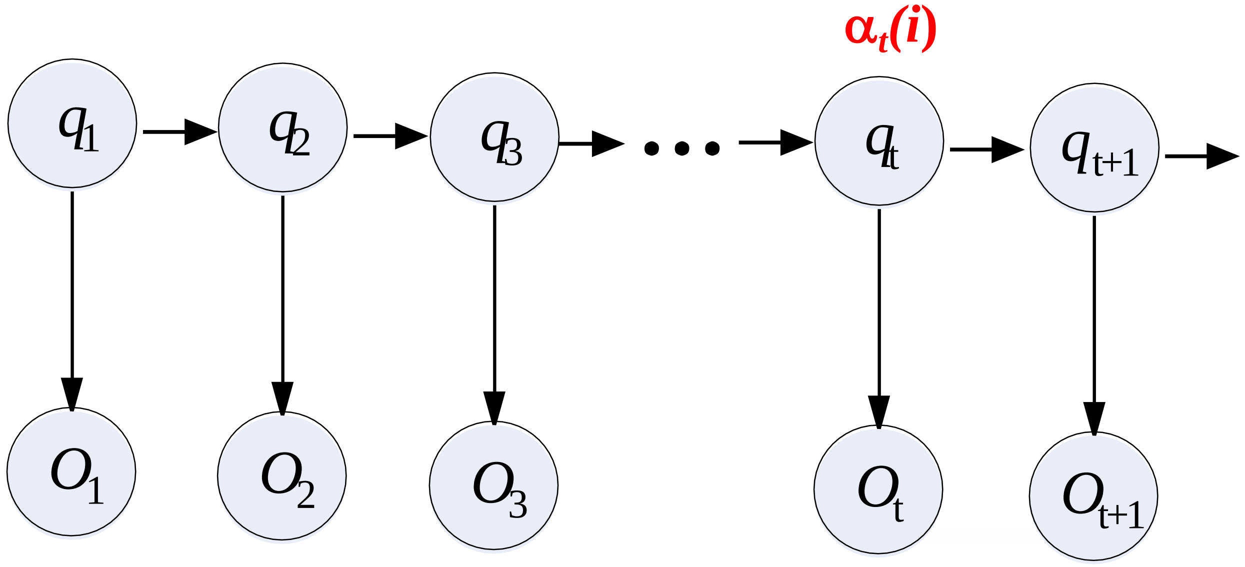
如果可以高效地计算$\alpha_t(i)$，就可以高效地求得 $p(O|\mu)$。

前向算法



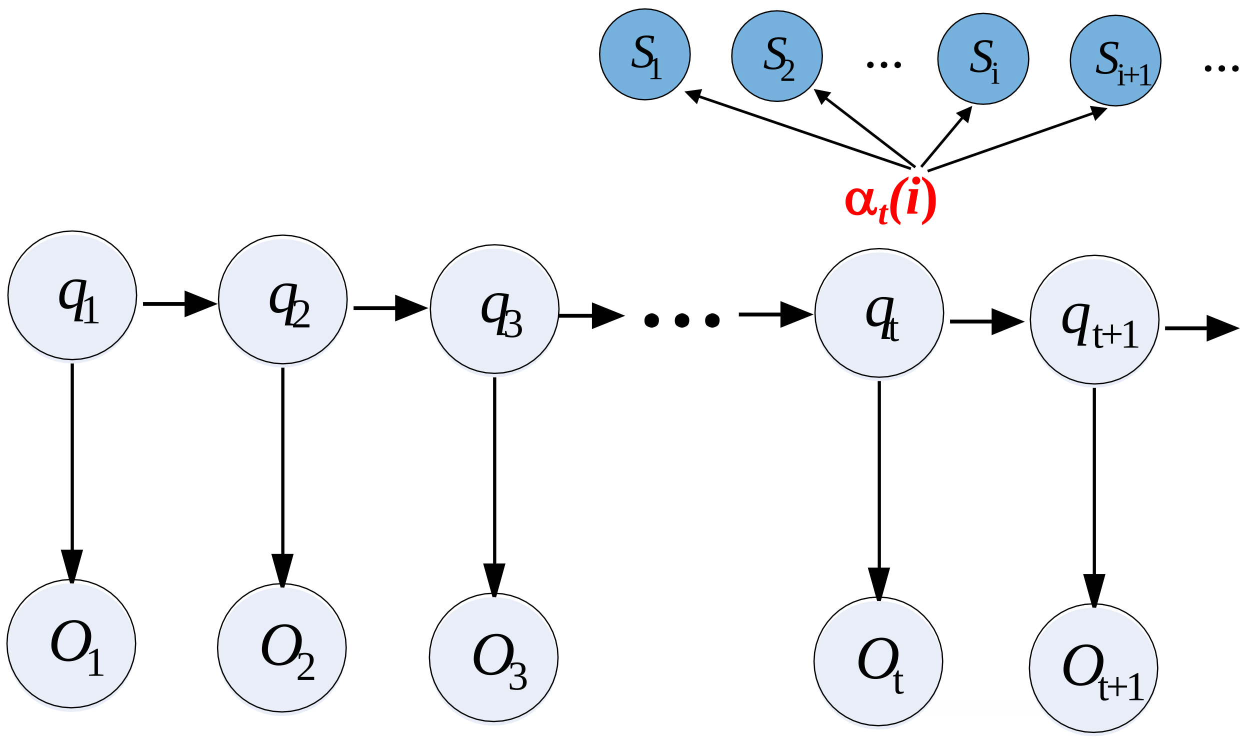
前向算法

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



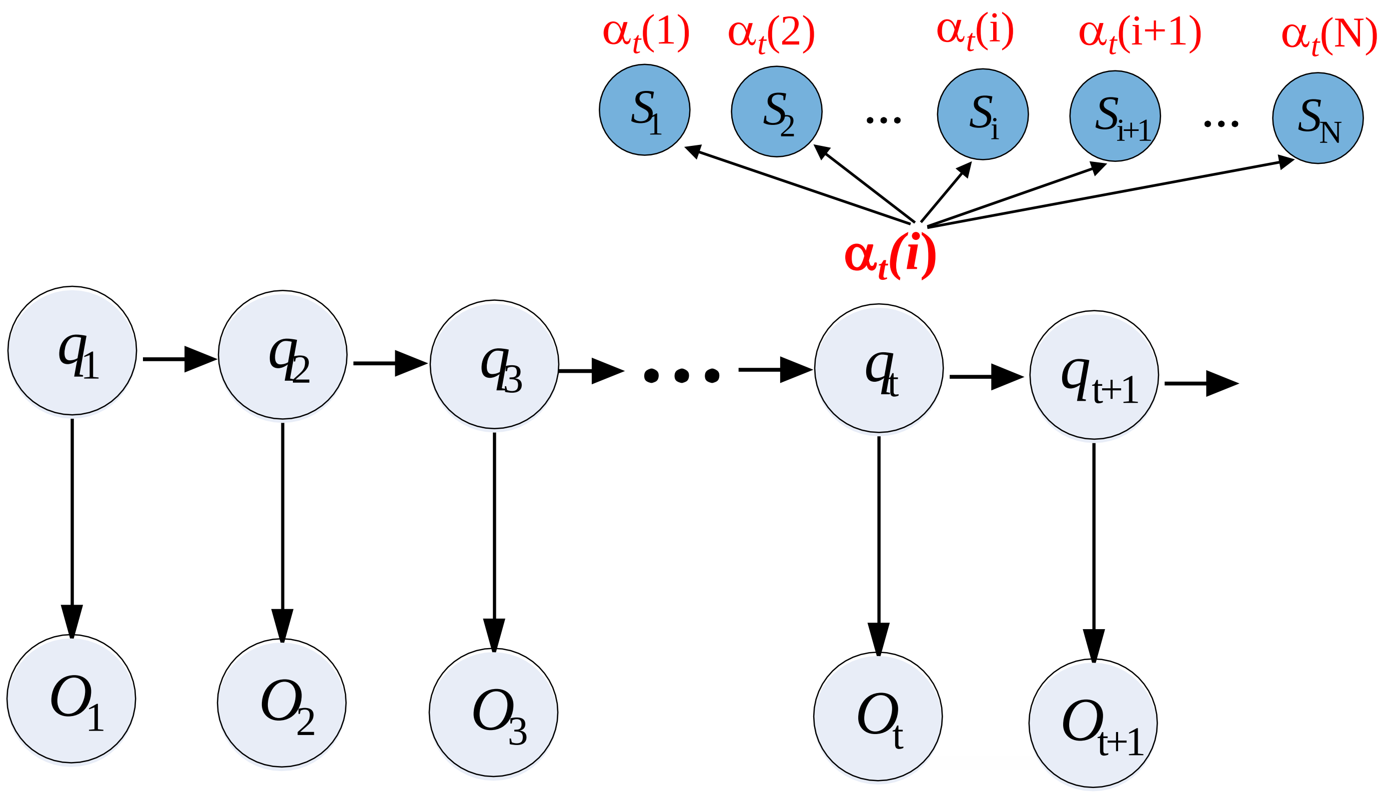
前向算法

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



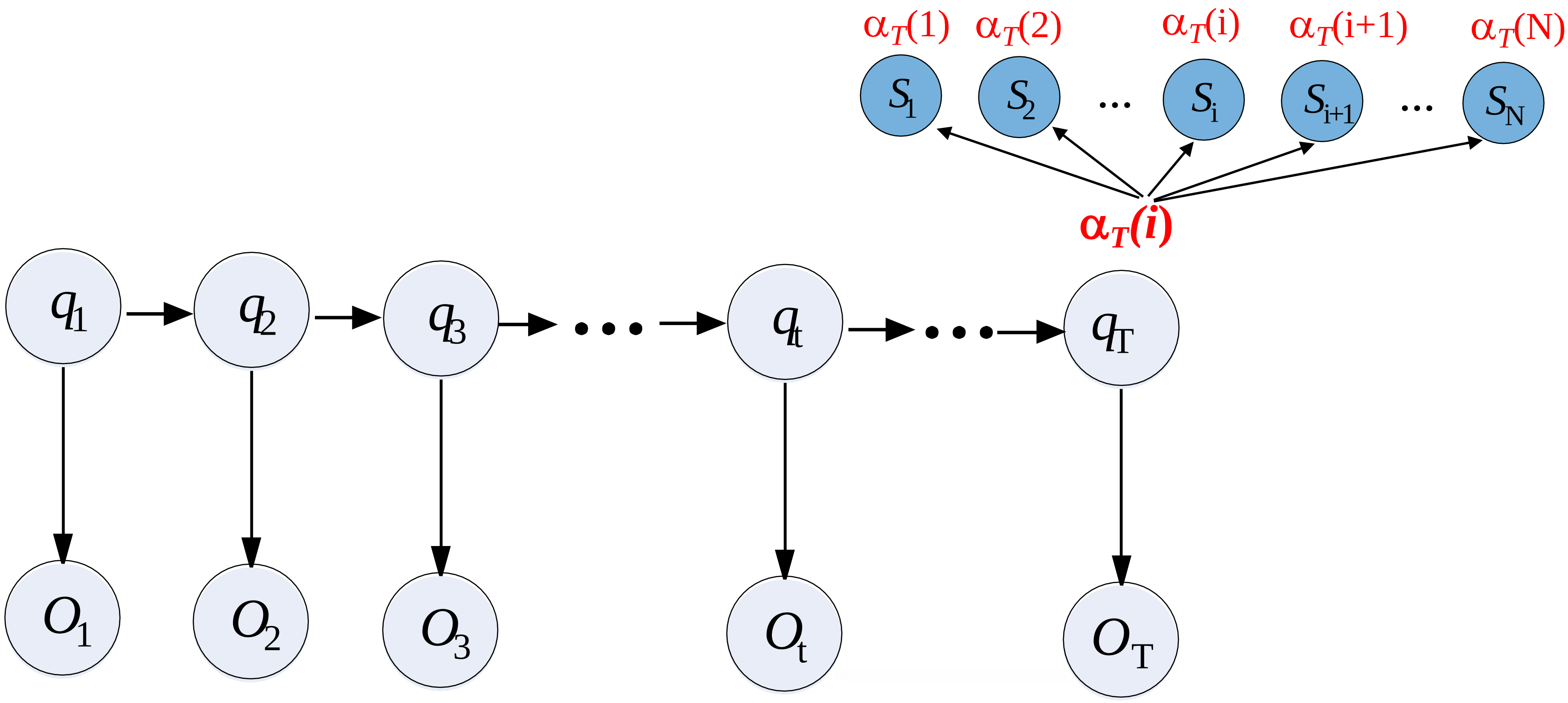
前向算法

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



前向算法

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



前向算法

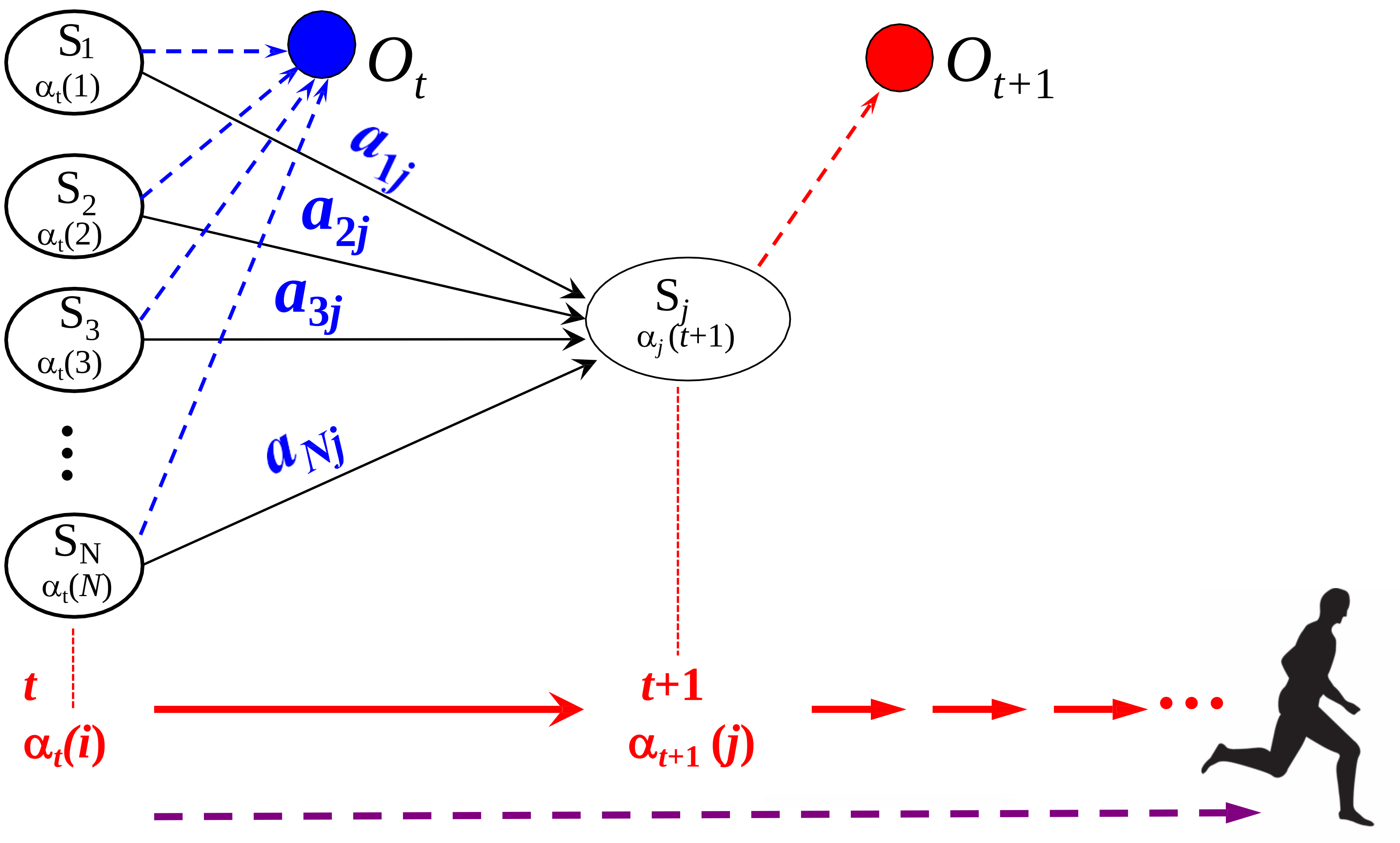
因为 $p(O|\mu)$ 是在到达状态 q_T 时观察到序列$O = O_1$ O_2 … O_T 的概率(所有可能的概率之和)：

$$p(O|\mu)=\sum_{S_i} p(O_1O_2\cdots O_T, q_T=S_i|\mu) = \sum_{i=1}^{N}\alpha_T(i) \quad \ldots (13)$$

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：

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

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)

前向算法

- 算法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：

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

前向算法

$\alpha_1(1)$

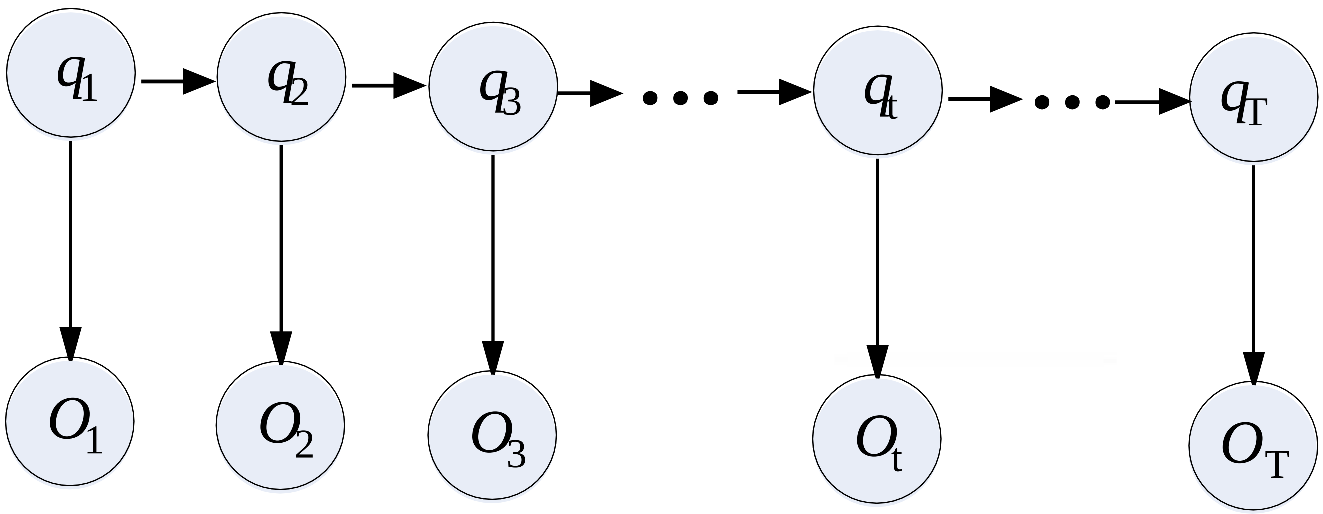
$\alpha_1(2)$

$\vdots$

$\alpha_1(i)$

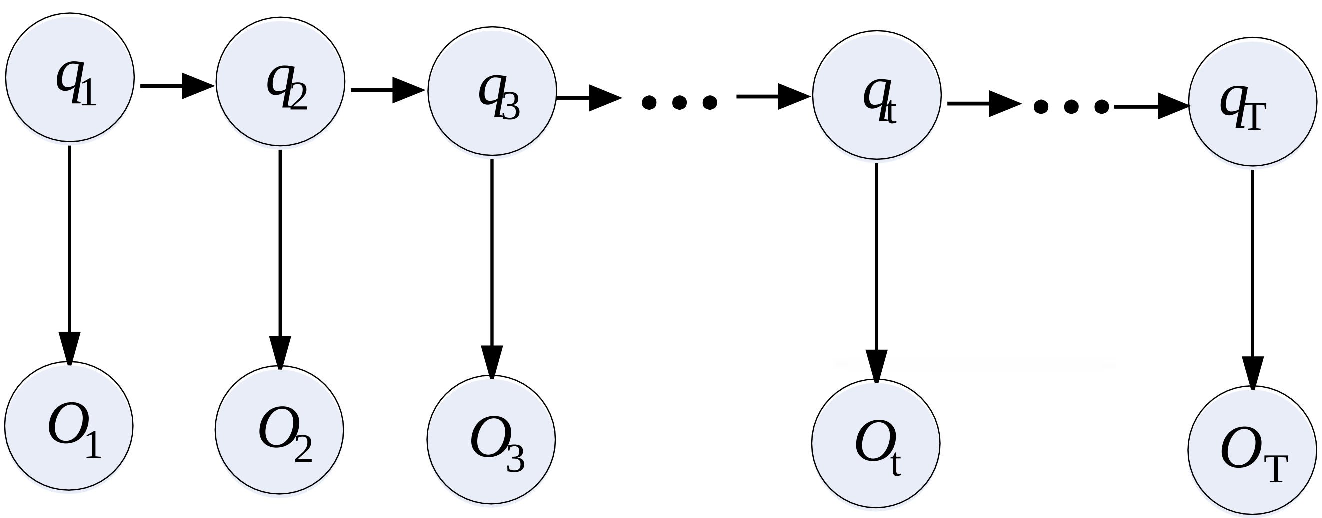
$\vdots$

$\alpha_1(N)$



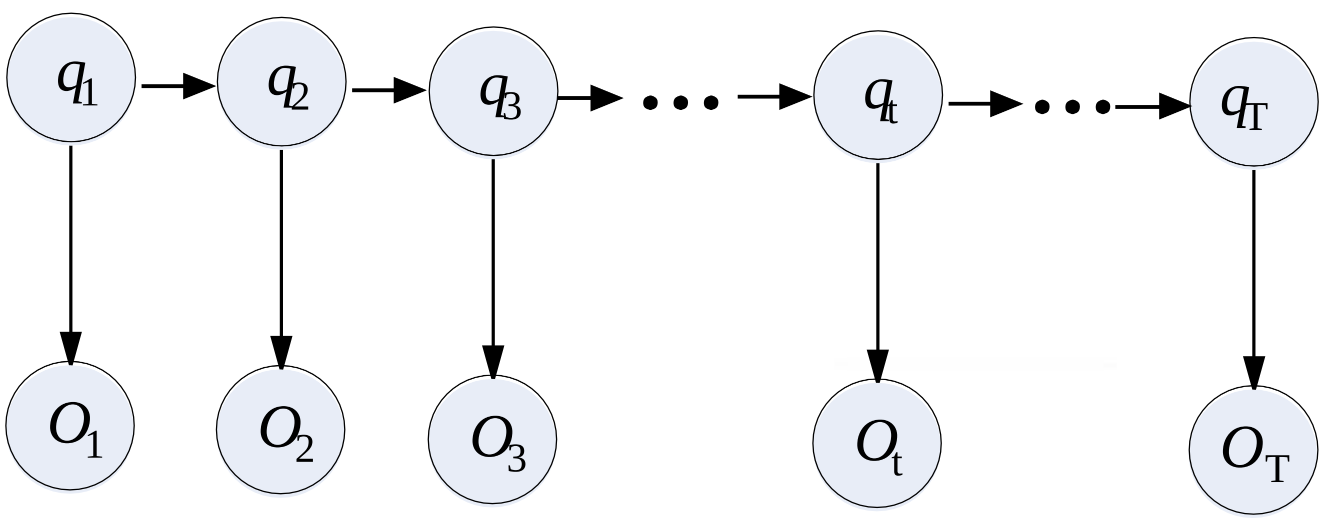
前向算法

$\alpha_1(1)$	$\alpha_2(1)$
$\alpha_1(2)$	$\alpha_2(2)$
$\vdots$	$\vdots$
$\alpha_1(i)$	$\alpha_2(i)$
$\vdots$	$\vdots$
$\alpha_1(N)$	$\alpha_2(N)$



前向算法

$$\begin{pmatrix} \alpha_1(1) & \alpha_2(1) & \cdots & \alpha_T(1) \\ \alpha_1(2) & \alpha_2(2) & & \alpha_T(2) \\ \vdots & \vdots & & \vdots \\ \alpha_1(i) & \alpha_2(i) & & \alpha_T(i) \\ \vdots & \vdots & & \vdots \\ \alpha_1(N) & \alpha_2(N) & \cdots & \alpha_T(N) \end{pmatrix}$$



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：$O(N^2T)$。

Lecture 5.4 后向算法

后向算法

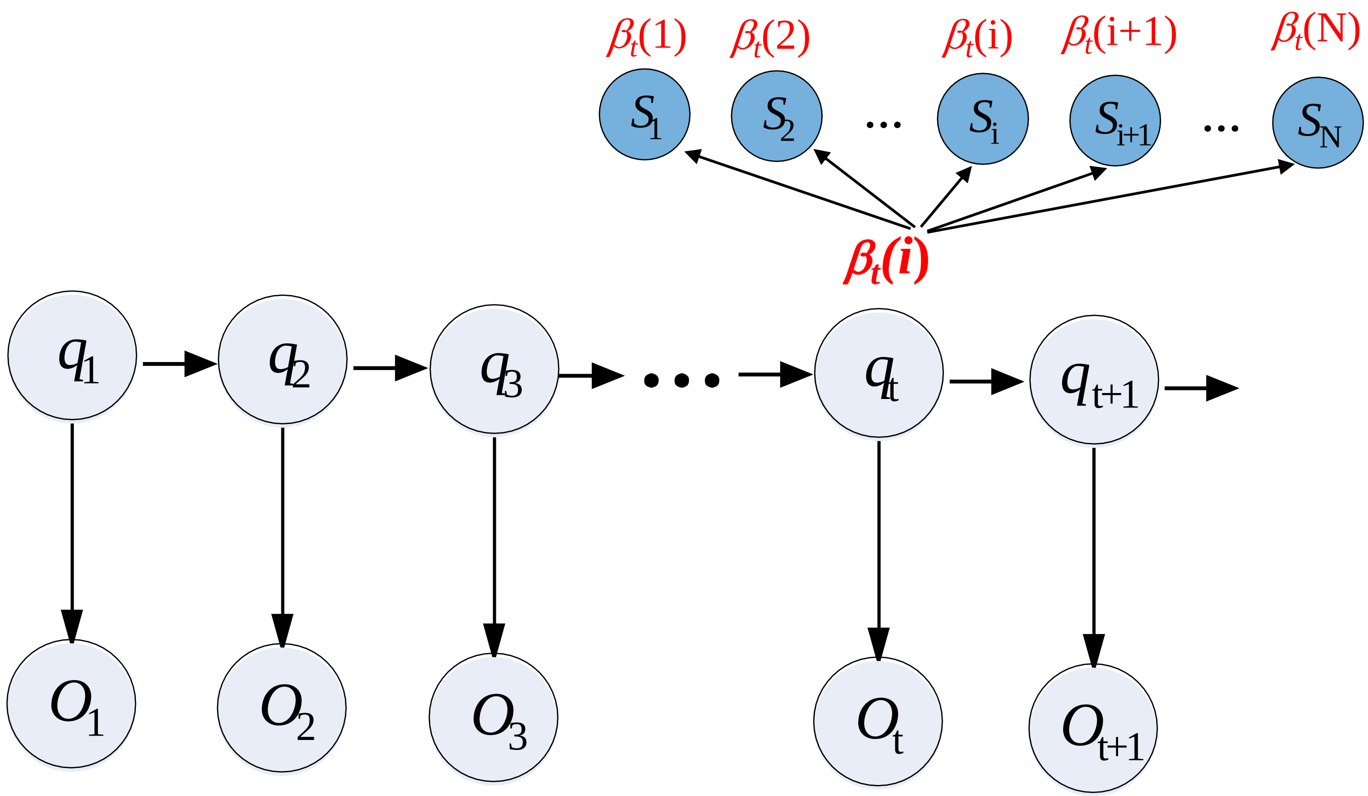
- 后向算法（The backward procedure）

  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：

$$\beta_t(i)=p(O_{t+1}O_{t+2}\cdots O_T\,|\,q_t=S_i,\mu) \qquad \ldots (15)$$

后向算法

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



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 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 $O_{t+2}O_{t+3}\cdots O_T$。

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：

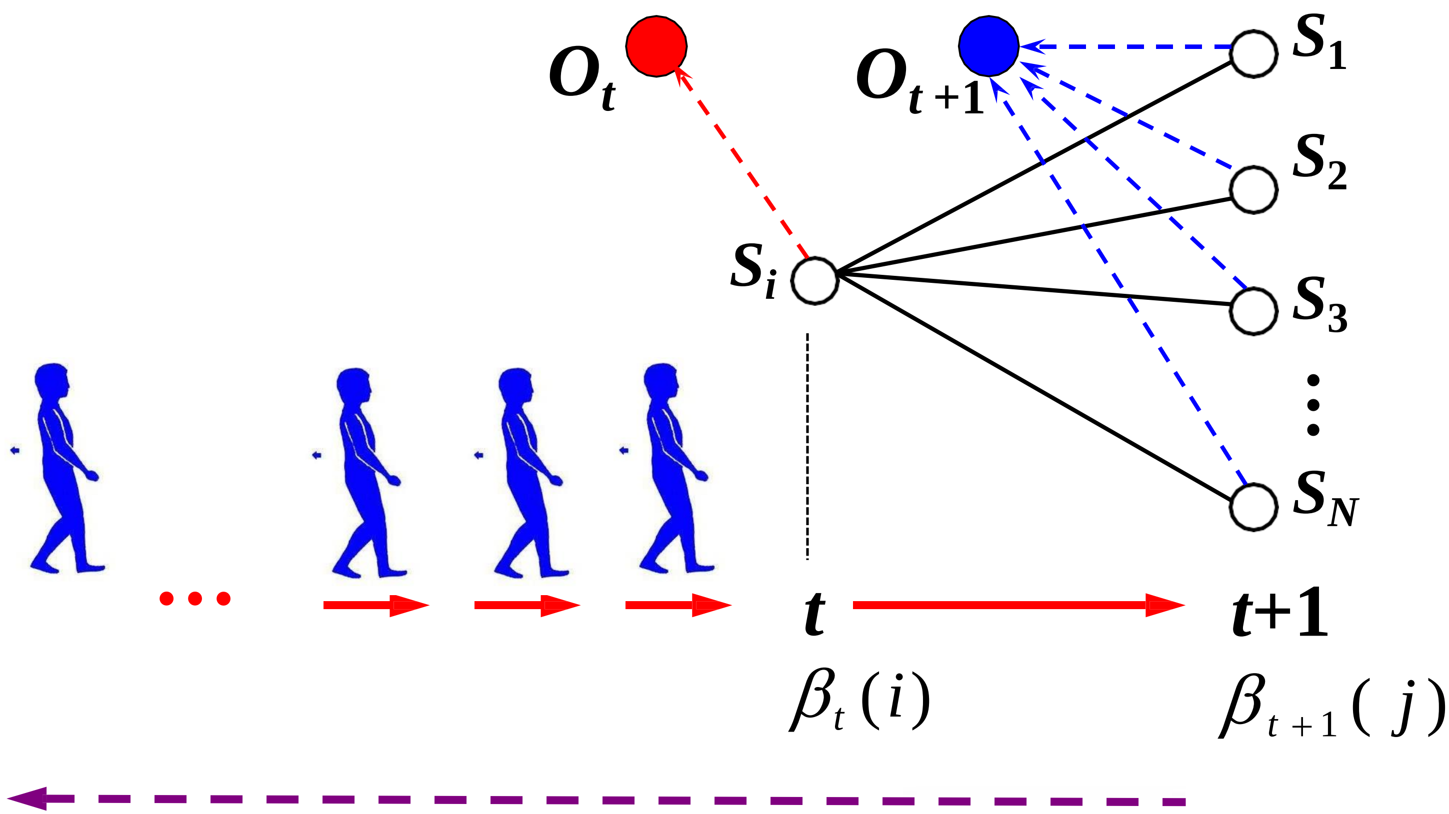
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \times \beta_{t+1}(j) \qquad \ldots (16)$$

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

(x 为模型的状态)

后向算法

算法图解：



后向算法

- 算法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：

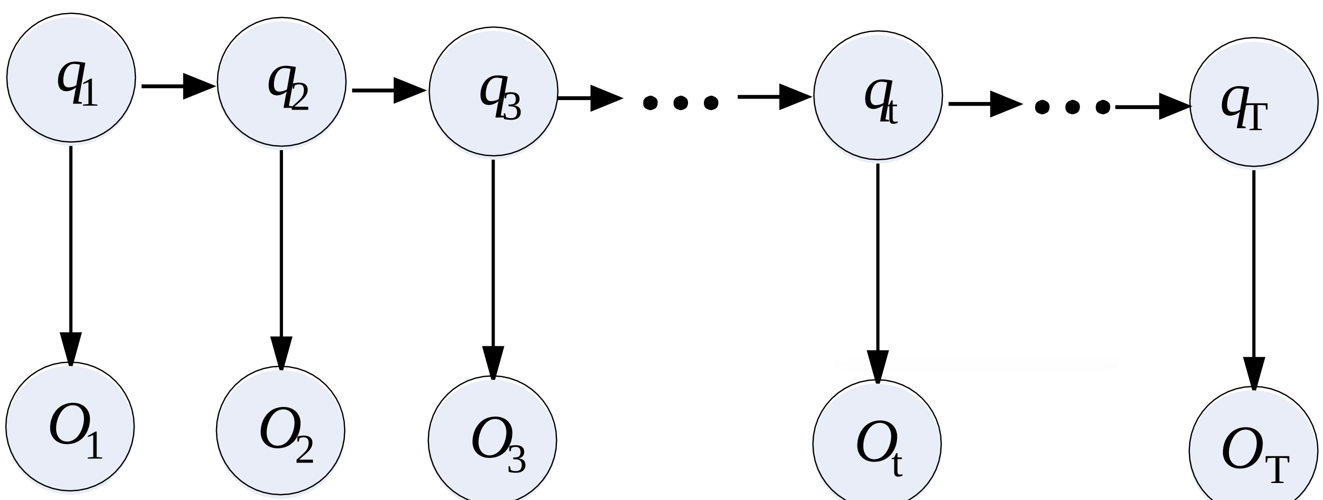
$$\beta_t(i)=\sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \times \beta_{t+1}(j), \quad T-1 \ge t \ge 1, \quad 1 \le i \le N$$

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

算法的时间复杂性：$O(N^2T)$

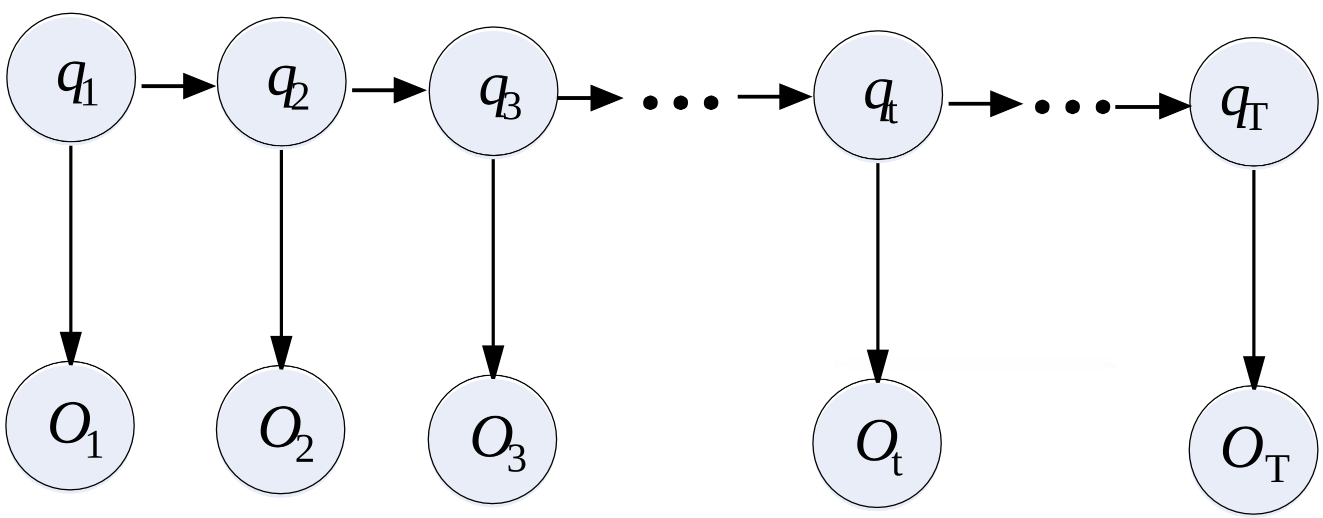
前向算法

$$\begin{pmatrix} & \beta_T(1) \\ & \beta_T(2) \\ & \vdots \\ & \beta_T(i) \\ & \vdots \\ & \beta_T(N) \end{pmatrix}$$



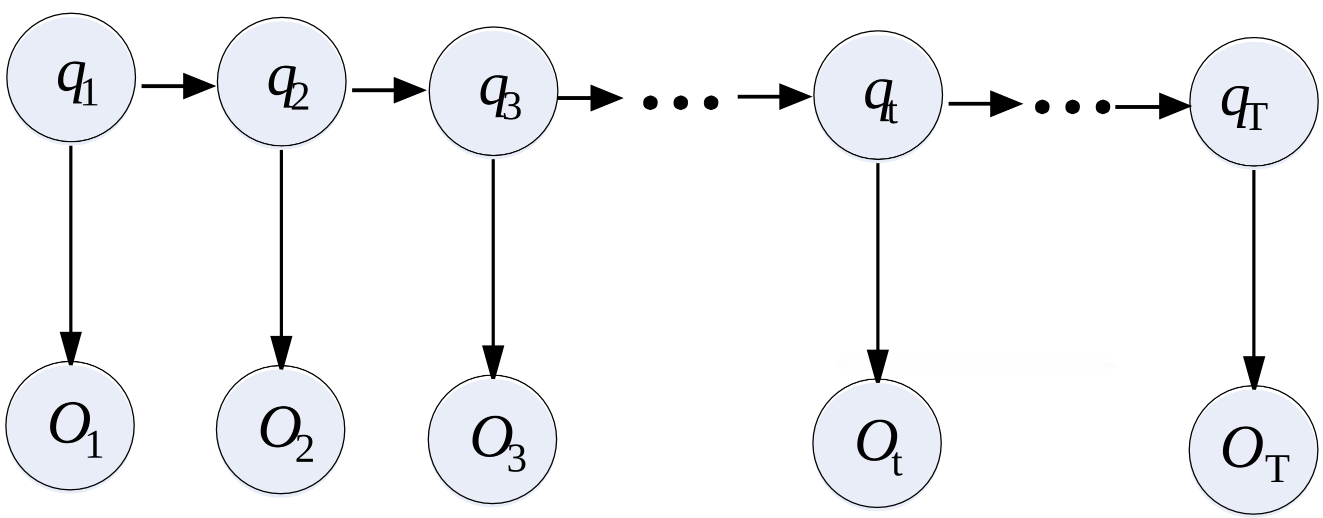
前向算法

$$
\begin{pmatrix}
\cdots & \beta_{T-1}(1) & \beta_T(1) \\
\cdots & \beta_{T-1}(2) & \beta_T(2) \\
 & \vdots & \vdots \\
\cdots & \beta_{T-1}(i) & \beta_T(i) \\
 & \vdots & \vdots \\
\cdots & \beta_{T-1}(N) & \beta_T(N)
\end{pmatrix}
$$



前向算法

$$
\begin{pmatrix}
\beta_1(1) & \cdots & \beta_{T-1}(1) & \beta_T(1) \\
\beta_1(2) & & \beta_{T-1}(2) & \beta_T(2) \\
\vdots & & \vdots & \vdots \\
\beta_1(i) & & \beta_{T-1}(i) & \beta_T(i) \\
\vdots & & \vdots & \vdots \\
\beta_1(N) & \cdots & \beta_{T-1}(N) & \beta_T(N)
\end{pmatrix}
$$



Lecture 5.5 Viterbi搜索算法

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?

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：

$$\widehat{Q} = \arg\max_{Q} p(Q \mid O, \mu) \qquad \ldots (21)$$

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?

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：

$$\widehat{Q} = \arg\max_{Q} p(Q \mid O,\mu) \qquad \ldots (21)$$

$$= \arg\max_{Q} \frac{p(Q, O|\mu)}{p(O|\mu)}$$

Viterbi 搜索算法

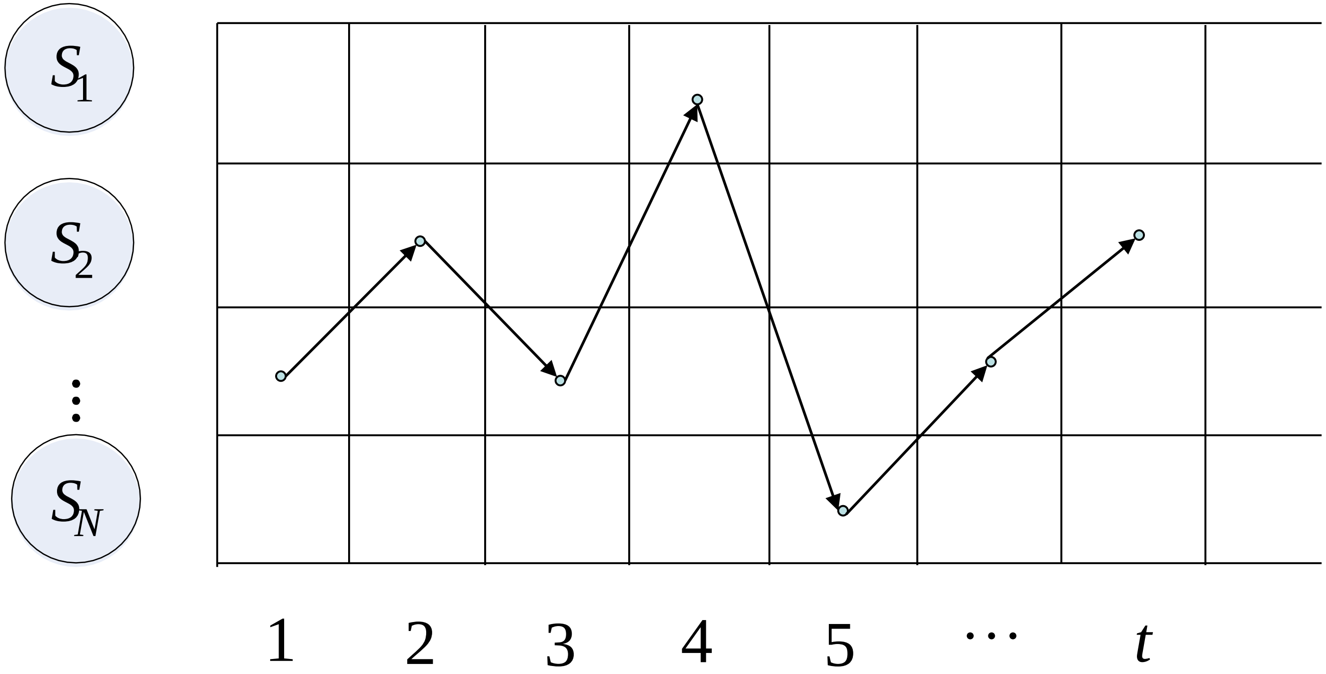
Viterbi 算法：动态搜索最优状态序列。

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，输出观察序列 $O = O_1O_2 \ldots O_t$ 的最大概率为：

$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} p(q_1,q_2,\cdots,q_t = S_i, O_1O_2\cdots O_t|\mu) \quad \ldots (22)$$

Viterbi 搜索算法

$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} p(q_1,q_2,\cdots,q_t = S_i, O_1O_2\cdots O_t|\mu) \quad \ldots (22)$$



递归计算：$\delta_{t+1}(i) = \max_j[\delta_t(j)\cdot a_{ji}]\cdot b_i(O_{t+1}) \quad \ldots (23)$

Viterbi 搜索算法

- 算法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：

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

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

Viterbi 搜索算法

(3) 结束：

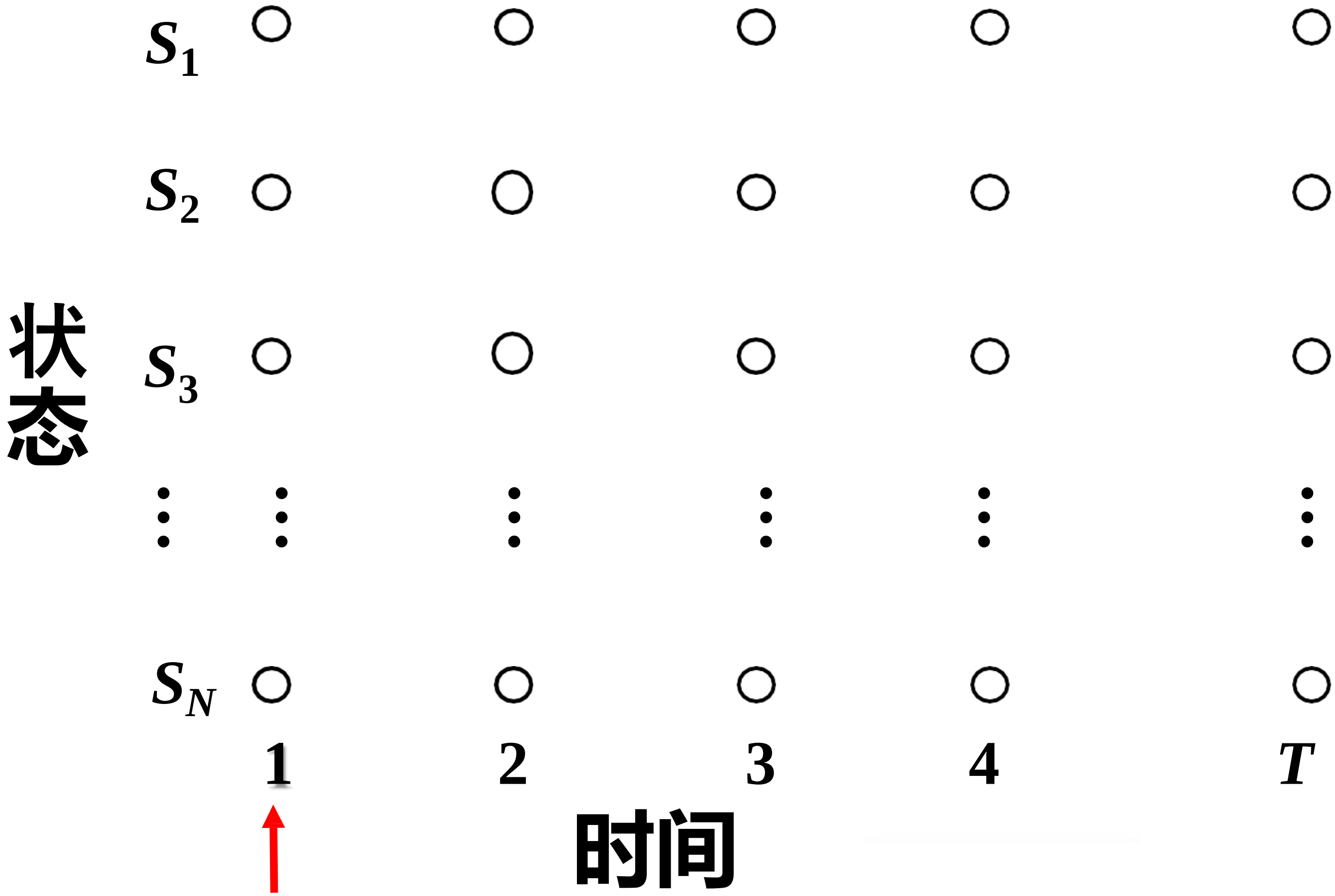
$$\widehat{Q}_T = \underset{1 \le i \le N}{\operatorname{argmax}}[\delta_T(i)], \qquad \widehat{p}(\widehat{Q}_T) = \max_{1 \le i \le N} \delta_T(i)$$

(4) 通过回溯得到路径（状态序列）：

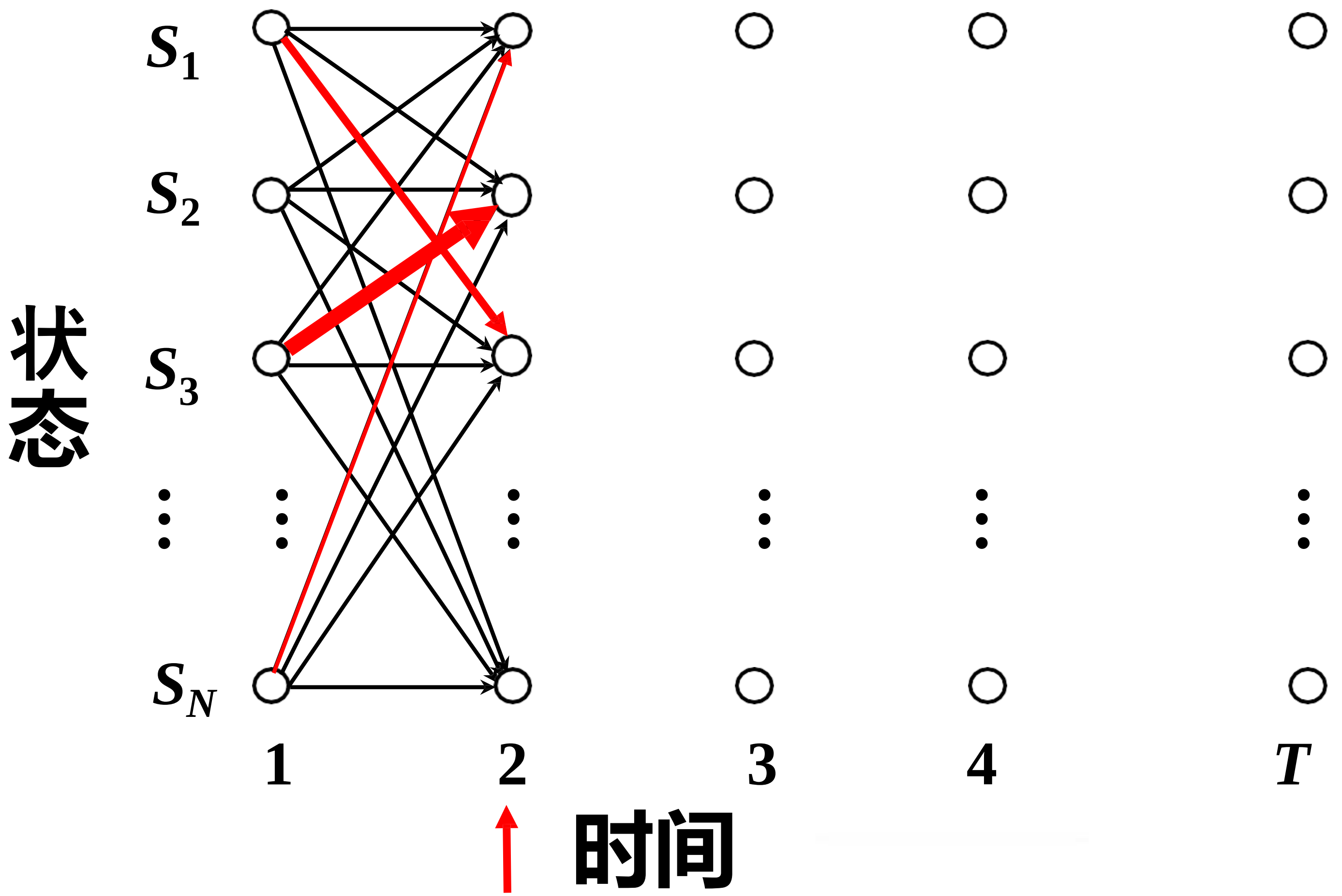
$$\widehat{q}_t = \psi_{t+1}(\widehat{q}_{t+1}), \quad t = T-1,\ T-2,\ \cdots,\ 1$$

算法的时间复杂度：$O(N^2 T)$

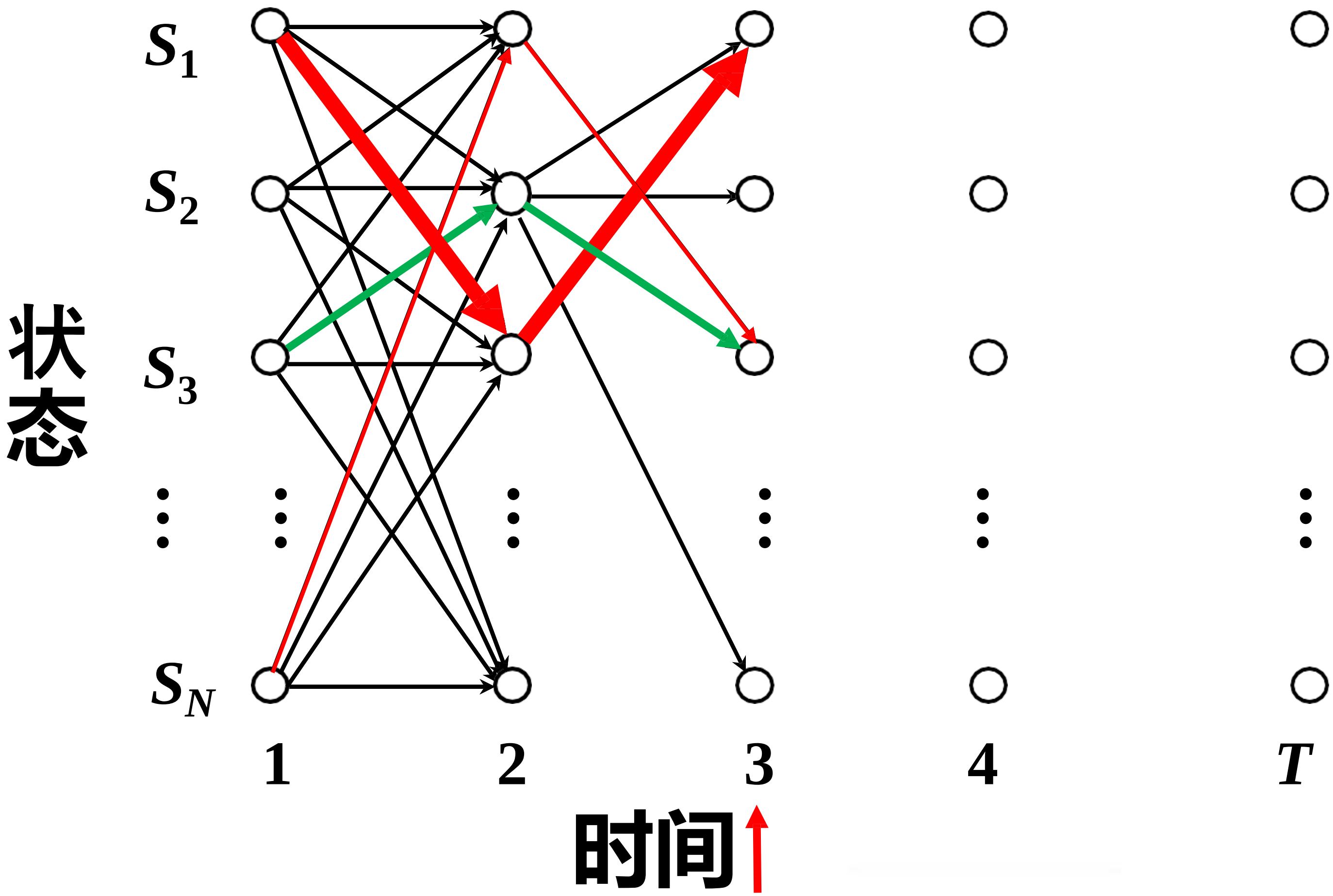
Viterbi 搜索算法



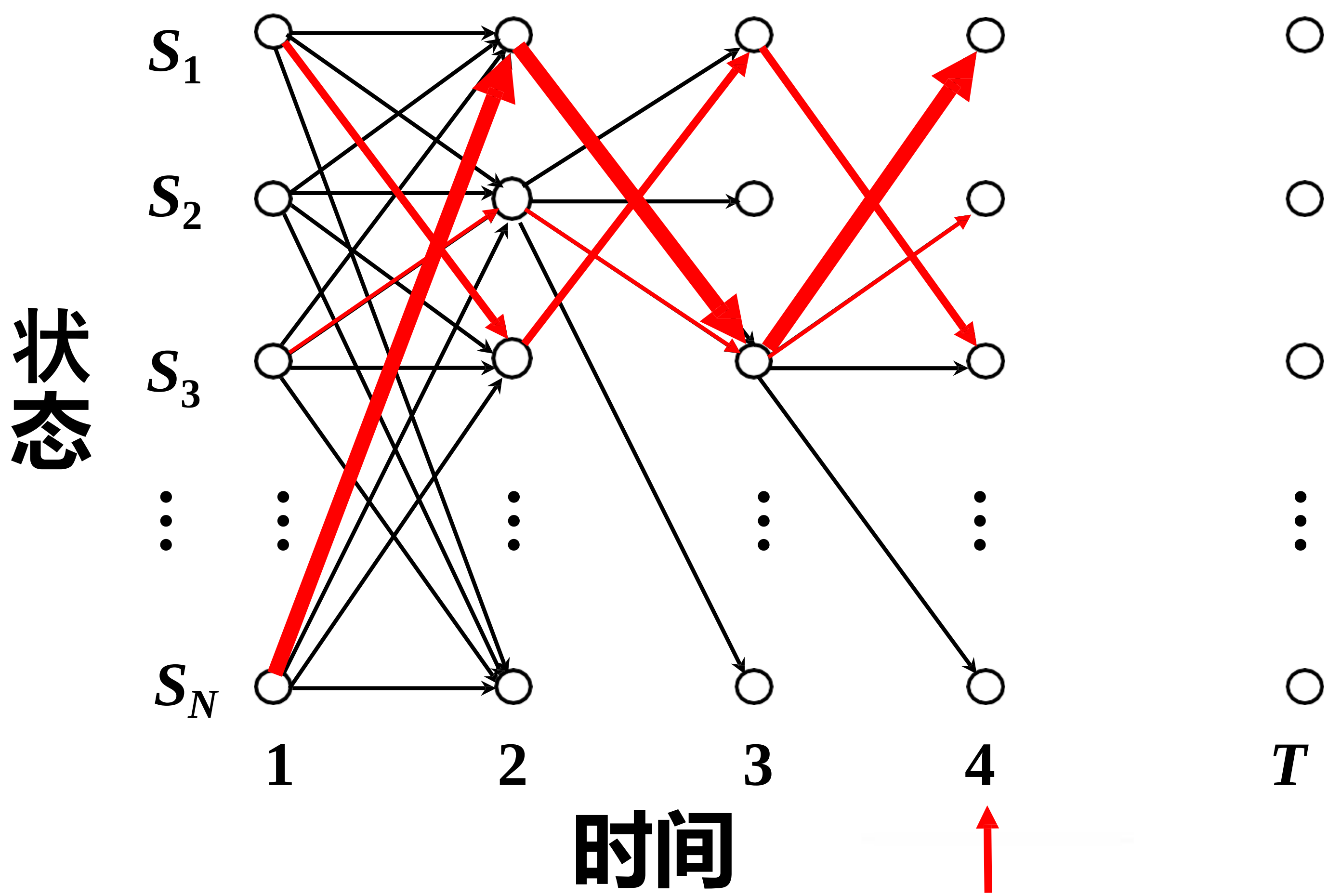
Viterbi 搜索算法



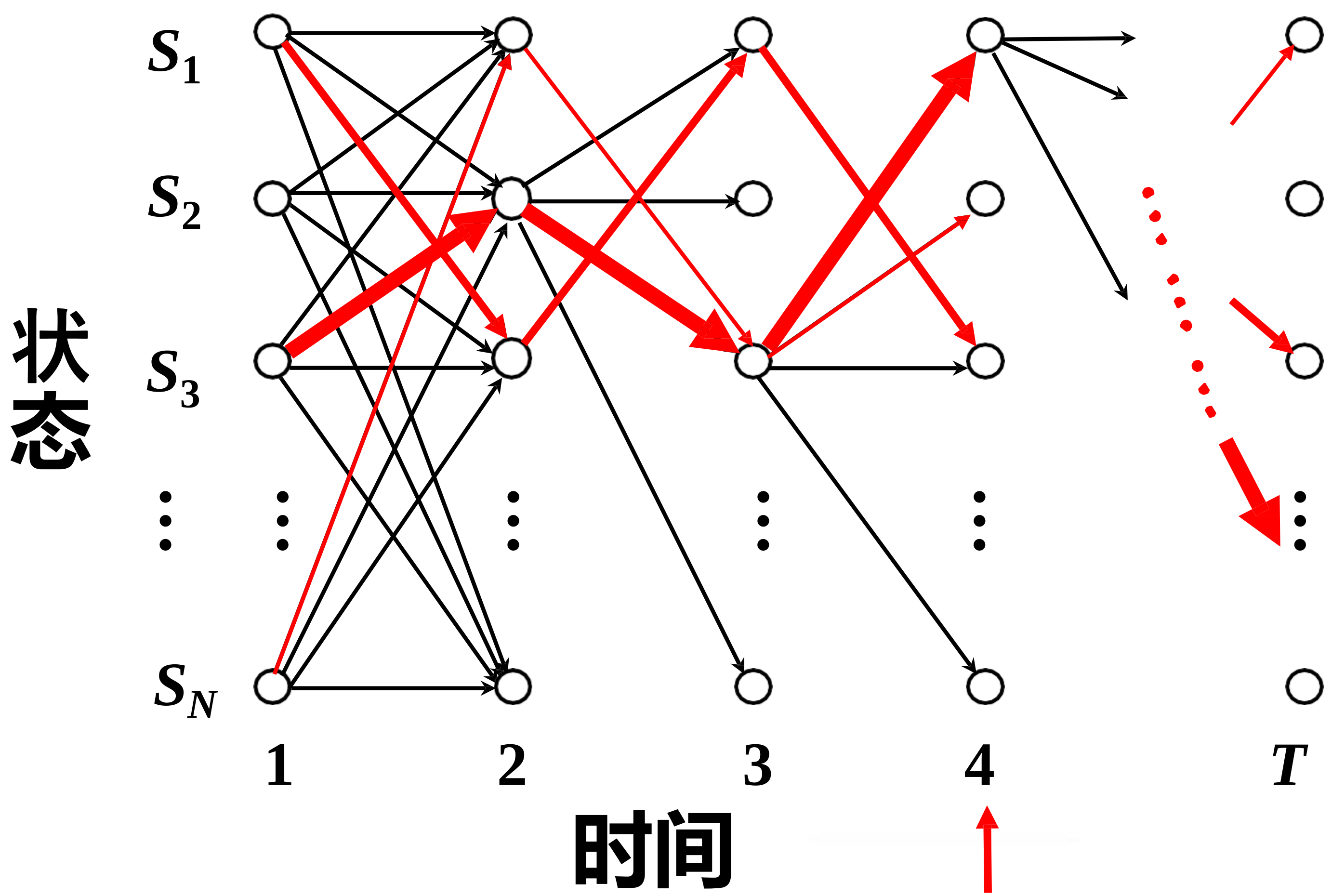
Viterbi 搜索算法



Viterbi 搜索算法



Viterbi 搜索算法



Lecture 5.6 参数学习

参数学习

◆问题3－模型参数学习

给定一个观察序列 $O = O_1O_2 \dots O_T$，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。

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(存在大量标注的样本), 可以用极大似然估计来计算μ的参数：

$$\pi_i = \delta(q_1, S_i)$$

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

$$\ldots (24)$$

参数学习

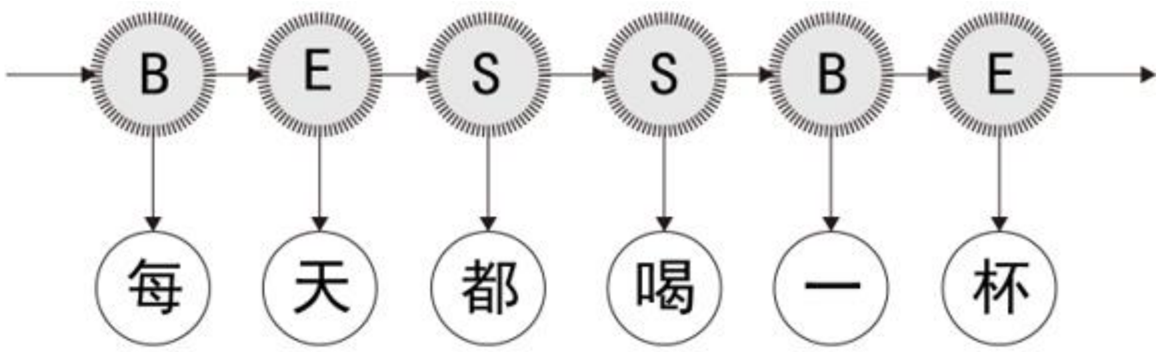
类似地，

$$\bar{b}_j(k) = \frac{Q\text{中从状态}q_j\text{输出符号}v_k\text{的次数}}{Q\text{到达}q_j\text{的总次数}} \quad \ldots (6.25)$$

Lecture 5.7 HMM应用举例

HMM应用举例

- 将状态值集合Q置为{B, E, M, S}，分别表示词的开始、结束、中间（begin、end、middle）及字符独立成词（single）；观测序列即为中文句子。
- 例如，“每天都喝一杯”通过HMM(Viterbi 算法)求解得到状态序列“B E S S B E”，则分词结果为“每天/都/喝/一杯”。



Thank you!

权小军 中山大学数据科学与计算机学院