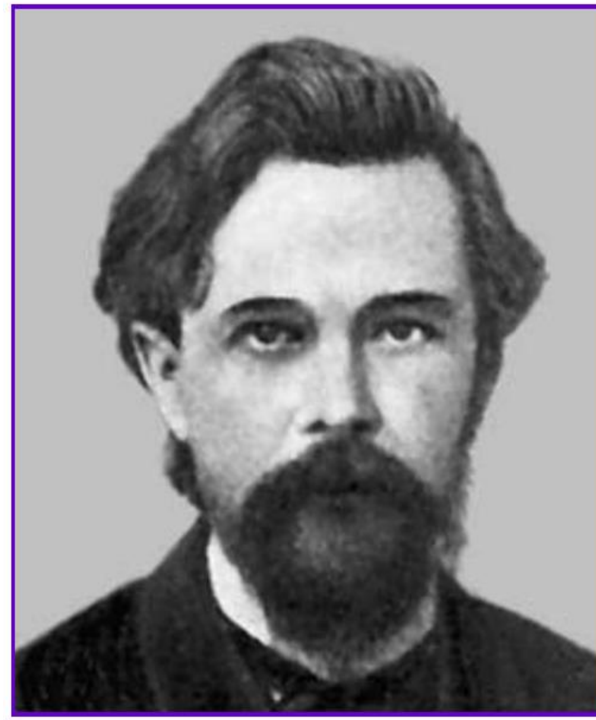
第六章
隐马尔可夫模型

目 录

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 $S_1, S_2, \cdots, S_N$，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 $S_j(1 \leq j \leq N)$ 的概率取决于前 $t-1$ 个时刻 $(1, 2, \cdots, t-1)$ 的状态，该概率为：

$$p\left(q_t = S_j \middle| q_{t-1} = S_i, q_{t-2} = S_k, \cdots\right)$$

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**一阶马尔可夫链**：

$$p(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \cdots) = p(q_t = S_j | q_{t-1} = S_i) \qquad (6\text{-}1)$$

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:

$$p\left(q_t = S_j \middle| q_{t-1} = S_i\right) = a_{ij}, 1 \leq i, j \leq N \quad (6\text{-}2)$$

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：

$$a_{ij} \geq 0 \tag{6-3}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{6-4}$$

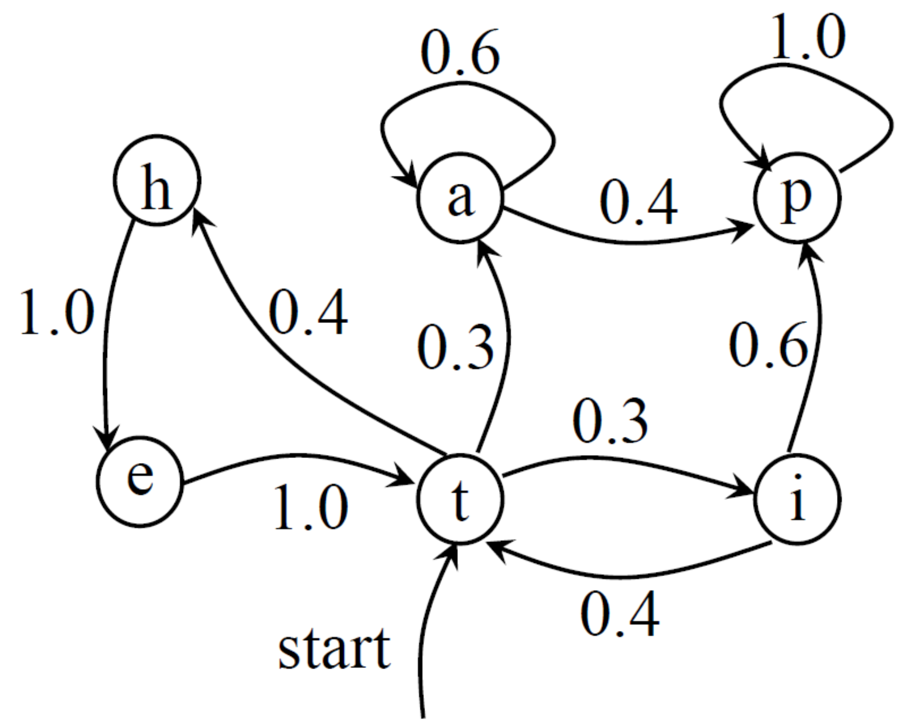
马尔可夫模型又可视为随机有限状态自动机，该有限状态自动机的每一个状态转换过程都有一个相应的概率，该概率表示自动机采用这一状态转换的可能性。

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

◆ 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）

状态序列 $S_1, \cdots, S_T$ 的概率：

$$p(S_1, \cdots, S_T)$$

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

$$= p(S_1) \times p(S_2|S_1) \times p(S_3|S_2) \times \cdots \times p(S_T|S_{T-1}) \qquad (6\text{-}5)$$

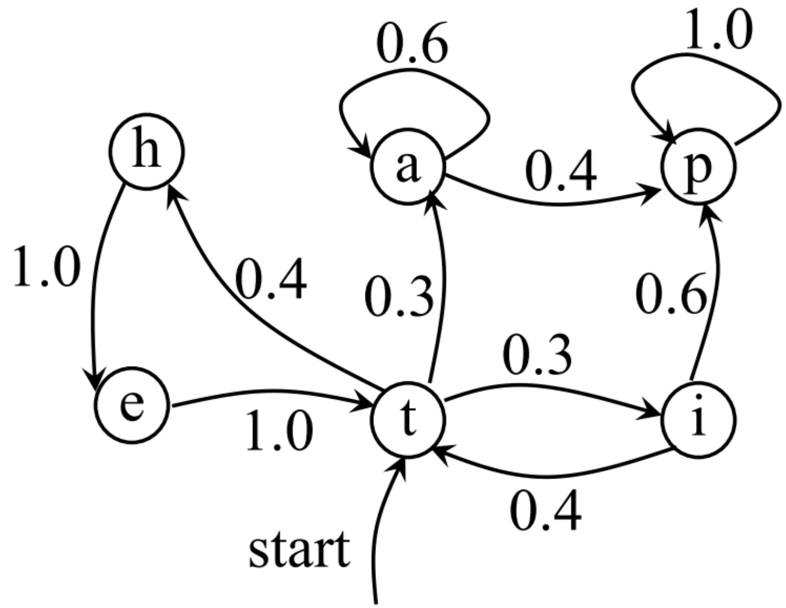
$$= \pi_{S_1} \prod_{t=1}^{T-1} a_{S_t S_{t+1}}$$

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

◆ 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)



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

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

$$= 0.18$$

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)

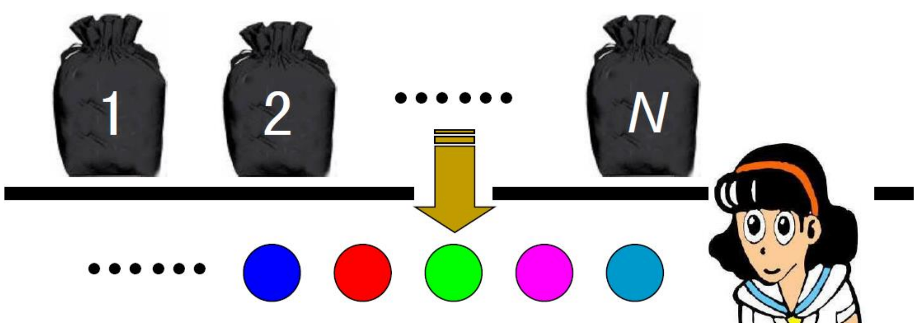
创建于1966年，美国数学家Leonard E. Baum和 J. A. EAGON提出。

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

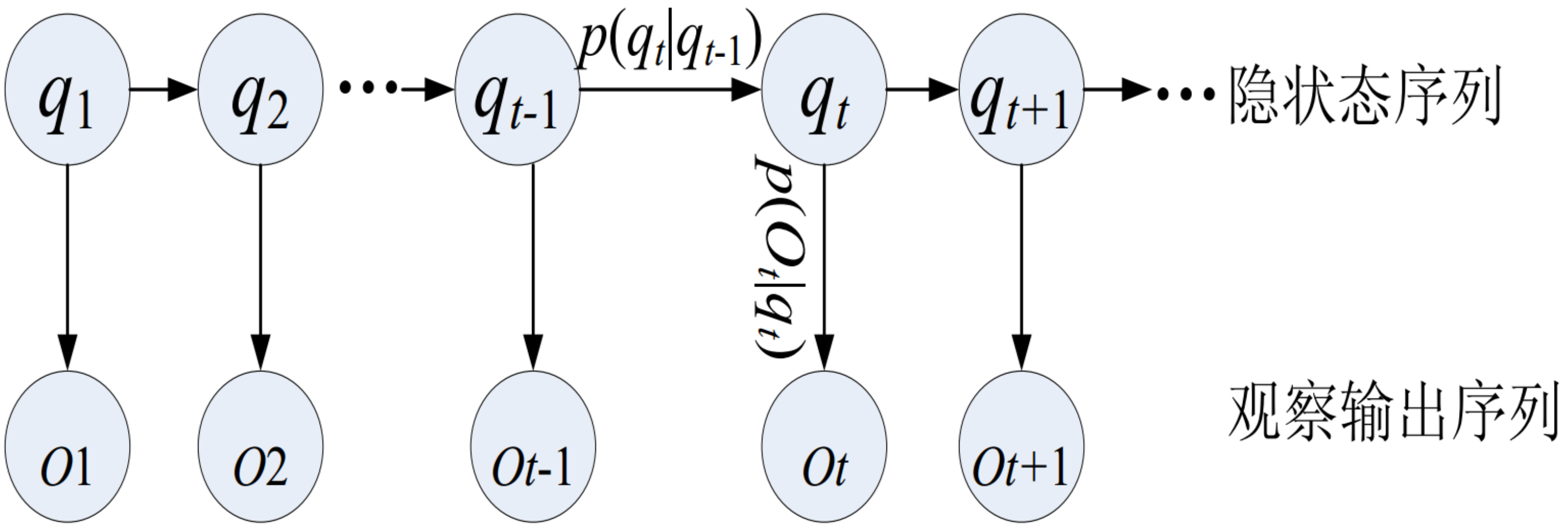
例如：

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

◆ 隐马尔可夫模型 (Hidden Markov Model, HMM)

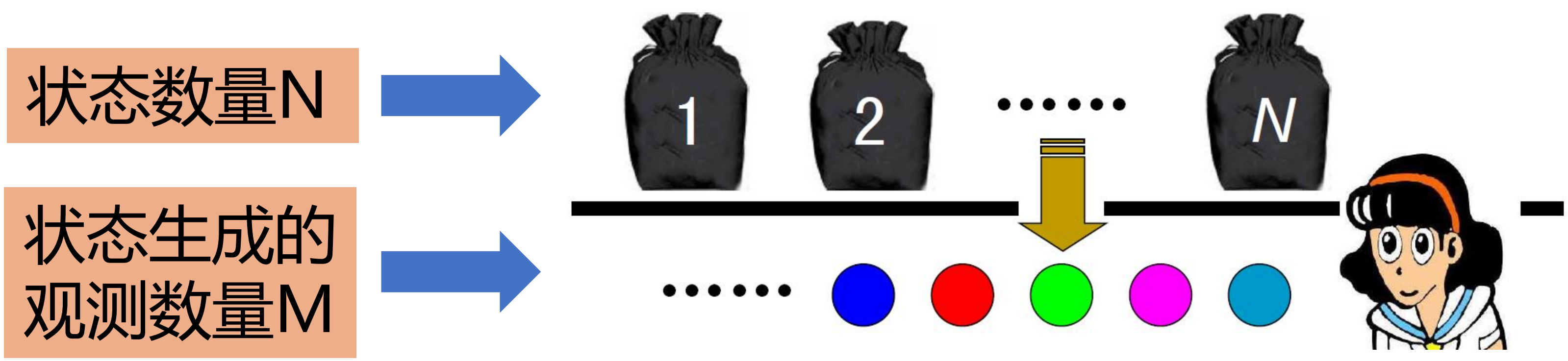


HMM 图解

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

◆ HMM的组成：

3. 模型中状态转移概率矩阵 $A = a_{ij}$ (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,

$$\begin{cases} a_{ij} = p\left(q_{t+1} = S_j \middle| q_t = S_i\right) \\ a_{ij} \geq 0 \\ \sum_{j=1}^{N} a_{ij} = 1 \end{cases} \tag{6-6}$$

6.2 隐马尔可夫模型

◆HMM的组成：

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，

$$\begin{cases} b_j(k) = p\left(O_t = v_k \middle| q_t = S_j\right), 1 \le j \le N, 1 \le k \le M \\ b_j(k) \ge 0 \\ \sum_{k=1}^{M} b_j(k) = 1 \end{cases}$$

(6-7)

6.2 隐马尔可夫模型

◆HMM的组成：

5\. 从初始状态的概率分布为：$\pi = \pi_i$，其中，

$$\begin{cases} \pi_i = p(q_1 = S_i) \\ \pi_i \geq 0 \\ \sum_{i=1}^{N} \pi_i = 1 \end{cases} \tag{6-8}$$

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 $O = O_1 O_2 \cdots O_T$:

(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，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 $O = O_1 O_2 \cdots O_T$ 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 $O = O_1 O_2 \cdots O_T$ 的情况下，如何选择在一定意义下“最优”的状态序列 $Q = q_1 q_2 \cdots q_T$，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 $O = O_1 O_2 \cdots O_T$，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$\mu = (A, B, \pi)$ 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 $p(O|\mu)$

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

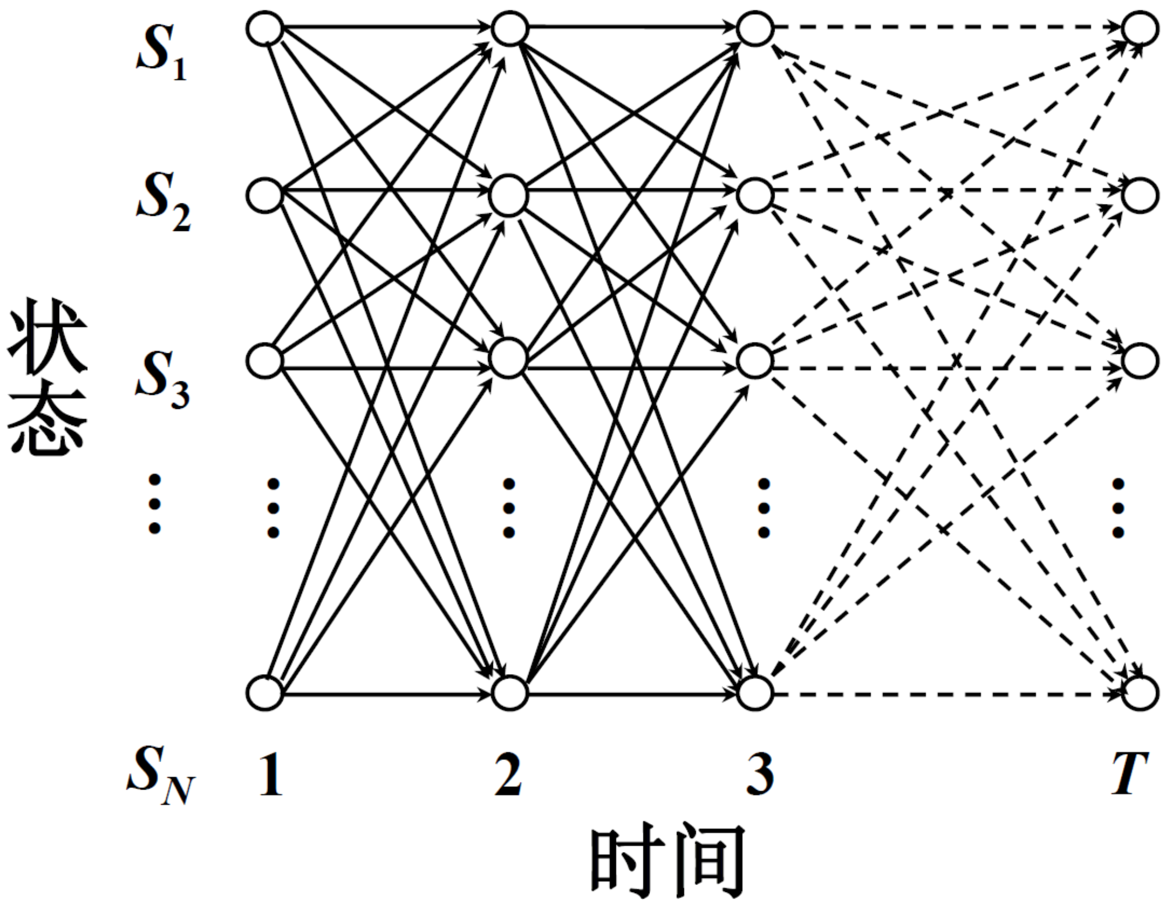
对于给定的状态序列 $Q=q_1q_2\cdots q_T$，$p(O|\mu)=?$

$$p(O|\mu)=\sum_Q p(O,Q|\mu)=\sum_Q \boxed{p(Q|\mu)}\cdot\boxed{p(O|Q,\mu)} \quad (6\text{-}9)$$

$$p(Q|\mu)=\pi_{q_1}a_{q_1q_2}a_{q_2q_3}\cdots a_{q_{t-1}q_T} \quad (6\text{-}10)$$

$$p(O|Q,\mu)=b_{q_1}(O_1)\cdot b_{q_2}(O_2)\cdots b_{q_T}(O_T) \quad (6\text{-}11)$$

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 $\alpha_t(i)$ 是在时间 t HMM 输出了序列 $O_1O_2\cdots O_t$，并且位于状态 S_i 的概率：

$$\alpha_t(i) = p(O_1O_2\cdots O_t, q_t = S_i|\mu) \qquad \text{(6-12)}$$

前向算法的主要思想：如果可以高效地计算前向变量 $\alpha_t(i)$，就可以根据 $\alpha_t(i)$ 求得 $p(O|\mu)$。

6.3 前向算法

因为 $p(O|\mu)$ 是在所有状态 q_T 下观察到序列 $O=O_1O_2\cdots O_T$ 的概率(所有可能的概率之和):

$$p(O|\mu)=\sum_{S_i}p(O_1O_2\cdots O_T,q_T=S_i|\mu)$$
$$=\sum_{i=1}^{N}\alpha_T(i) \qquad (6\text{-}13)$$

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:

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j(O_{t+1}) \qquad (6\text{-}14)$$

6.3 前向算法

$$\alpha_{t+1}(j) = p(O_1 O_2 \cdots O_{t+1}, q_{t+1} = S_j | \mu)$$

$$= p(O_1 O_2 \cdots O_{t+1} | q_{t+1} = S_j, \mu) \times p(q_{t+1} = S_j | \mu)$$

$$= p(O_{t+1} | q_{t+1} = S_j, \mu) \times p(O_1 O_2 \cdots O_t | q_{t+1} = S_j, \mu) \times p(q_{t+1} = S_j | \mu)$$

$$= p(O_{t+1} | q_{t+1} = S_j, \mu) \times p(O_1 O_2 \cdots O_t, q_{t+1} = S_j | \mu)$$

$$= p(O_{t+1} | q_{t+1} = S_j, \mu) \times \sum_{i=1}^{N} p(O_1 O_2 \cdots O_t, q_t = S_i, q_{t+1} = S_j | \mu)$$

$$= p(O_{t+1} | q_{t+1} = S_j, \mu) \times \sum_{i=1}^{N} p(O_1 O_2 \cdots O_t, q_t = S_i | \mu) \times p(q_{t+1} = S_j | O_1 O_2 \cdots O_t, q_t = S_i, \mu)$$

6.3 前向算法

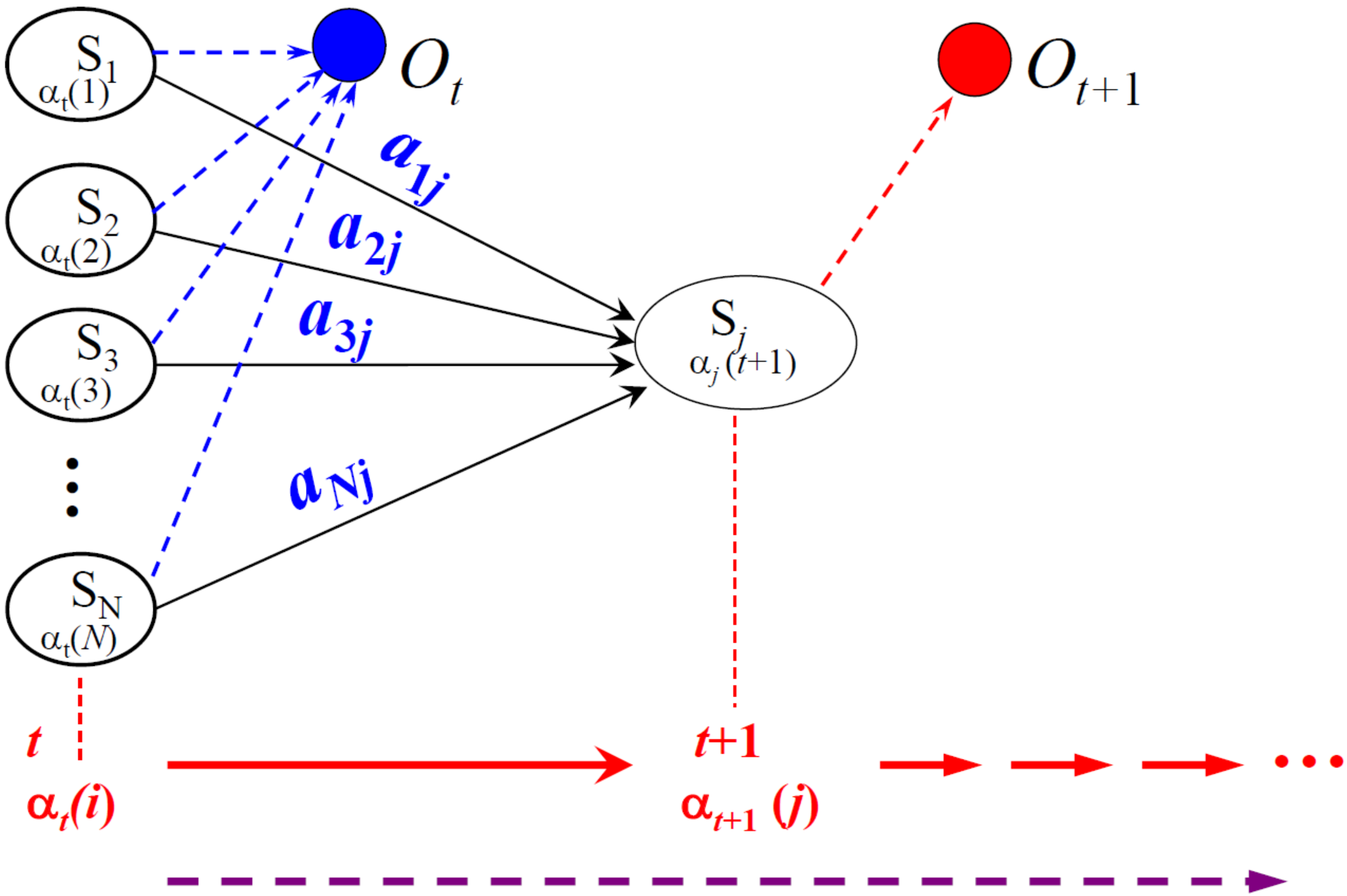
$$= p\left(O_{t+1} \middle| q_{t+1} = S_j, \mu\right)$$
$$\times \sum_{i=1}^{N} p(O_1 O_2 \cdots O_t, q_t = S_i | \mu) \times p\left(q_{t+1} = S_j \middle| O_1 O_2 \cdots O_t, q_t = S_i, \mu\right)$$

$$= p\left(O_{t+1} \middle| q_{t+1} = S_j, \mu\right)$$
$$\times \sum_{i=1}^{N} p(O_1 O_2 \cdots O_t, q_t = S_i | \mu) \times p\left(q_{t+1} = S_j \middle| q_t = S_i, \mu\right)$$

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

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

6.3 前向算法



S_1
$\alpha_t(1)$
S_2
$\alpha_t(2)$
S_3
$\alpha_t(3)$
S_N
$\alpha_t(N)$
O_t
O_{t+1}
a_{1j}
a_{2j}
a_{3j}
a_{Nj}
S_j
$\alpha_j(t+1)$
t
$\alpha_t(i)$
$t+1$
$\alpha_{t+1}(j)$

6.3 前向算法

◆ 算法6.1：前向算法

(1) 初始化：

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

(2) 循环计算：

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

(3) 结束，输出：

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

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：$\alpha_t(1),\alpha_t(2),\cdots,\alpha_t(N)$，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

◆ 问题1：快速计算观察序列概率 $p(O|\mu)$

在给定模型 $\mu = (A, B, \pi)$ 和观察序列 $O = O_1 O_2 \cdots O_T$，快速计算 $p(O|\mu)$：

➢ 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：

$$\beta_t(i) = p(O_{t+1} O_{t+2} \cdots O_T | q_t = S_i, \mu) \quad (6\text{-}15)$$

6.4 后向算法

◆ 后向算法 (The backward procedure):

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 $O_{t+2}O_{t+3}\cdots O_T$。

6.4 后向算法

◆ 后向算法 (The backward procedure):

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：

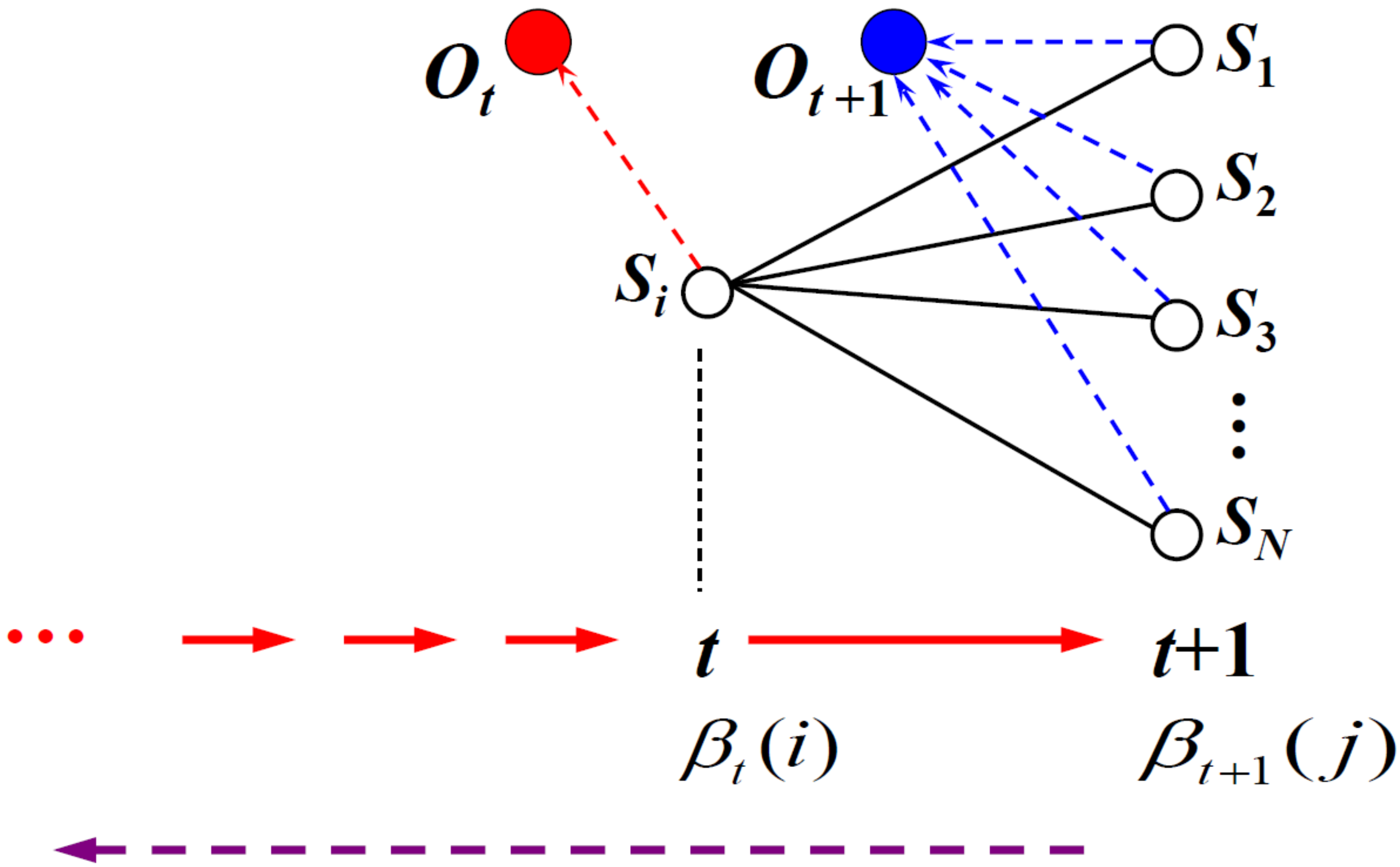
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \qquad (6\text{-}16)$$

归纳顺序：

$$\beta_T(x), \beta_{T-1}(x), \cdots \beta_1(x) \quad (x \text{ 为模型的状态})$$

6.4 后向算法

算法图解：



6.4 后向算法

◆ 算法6.2：后向算法

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

(2) 循环计算：

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

$$T - 1 \ge t \ge 1, 1 \le i \le N$$

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

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?

问题的答案不是唯一的，因为关键在于如何理解“最优状态序列”?

一种解释：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 $\gamma_t(i) = p(q_t = S_i | O, \mu)$ 最大。

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?

$$\gamma_t(i) = p(q_t = S_i | O, \mu) = \frac{p(q_t = S_i, O | \mu)}{p(O|\mu)} \quad (6\text{-}17)$$

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 $O = O_1O_2\cdots O_t$。根据前向变量的定义，实现这一步的概率为 $\alpha_t(i)$。

(2) 从时间 t，状态 S_i 出发，模型输出 $O = O_{t+1}O_{t+2}\cdots O_T$，根据后向变量定义，实现这一步的概率为 $\beta_t(i)$。于是:

$$p(q_t = S_i, O|\mu) = \alpha_t(i) \times \beta_t(i) \qquad (6\text{-}18)$$

6.5 Viterbi 搜索算法

分解过程:

而 $p(O|\mu)$ 与时间 t 的状态无关，因此:

$$p(O|\mu) = \sum_{i=1}^{N} \alpha_t(i) \times \beta_t(i) \tag{6-19}$$

将公式(6.18)和(6.19)带入(6.17)式得:

$$\gamma_t(i) = \frac{\alpha_t(i) \times \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \times \beta_t(i)} \tag{6-20}$$

t 时刻的最优状态为: $\hat{q}_t = \underset{1 \le i \le N}{\operatorname{argmax}} \left(\gamma_t(i)\right)$

6.5 Viterbi 搜索算法

注：观察序列概率计算

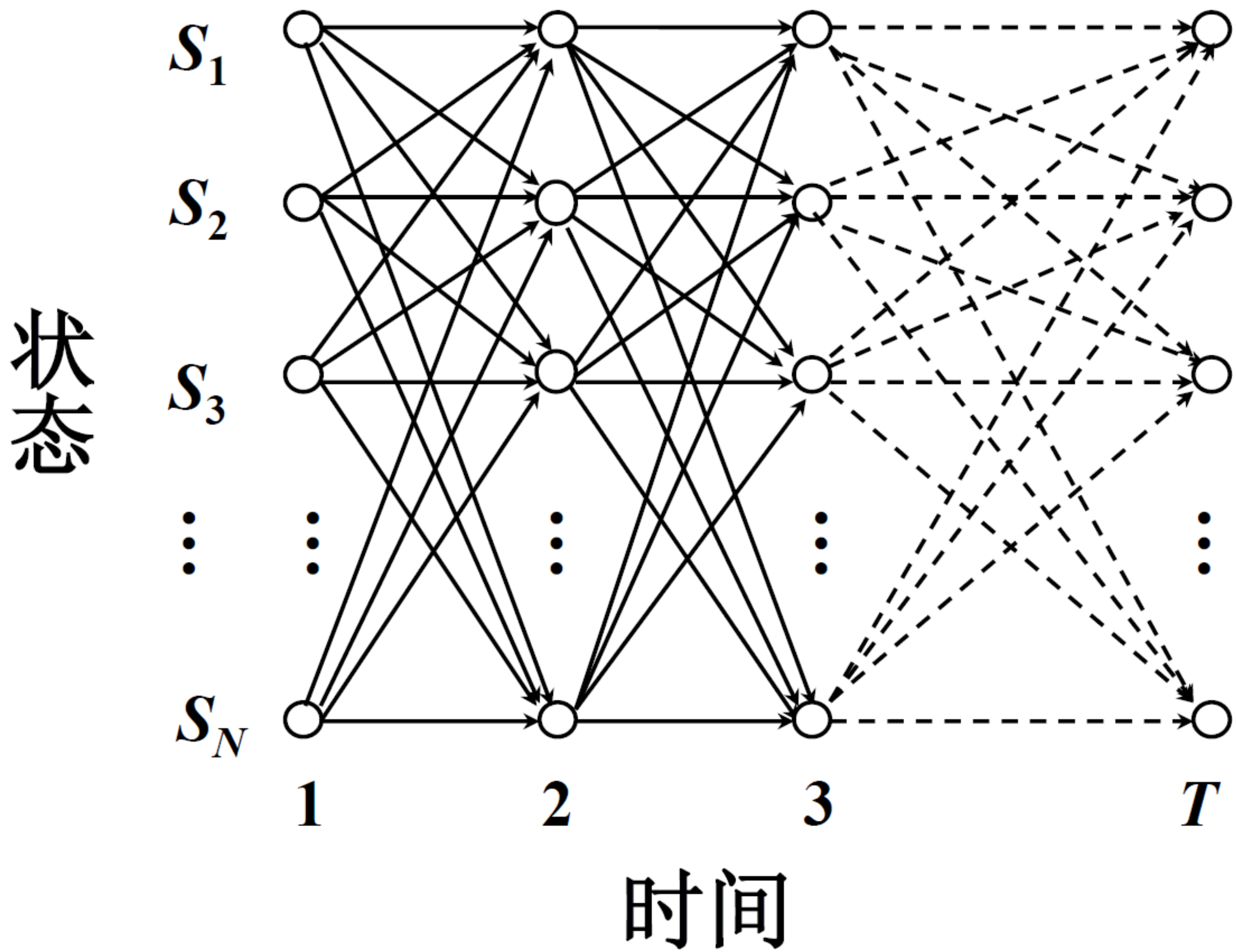
$$
\begin{aligned}
p(q_t = S_i, O|\mu) \\
&= p(O_1 \cdots O_T, q_t = S_i|\mu) \\
&= p(O_1 \cdots O_t, q_t = S_i, O_{t+1} \cdots O_T|\mu) \\
&= p(O_1 \cdots O_t, q_t = S_i|\mu) \times p(O_{t+1} \cdots O_T \,|O_1 \cdots O_t, q_t = S_i, \mu) \\
&= p(O_1 \cdots O_t, q_t = S_i|\mu) \times p(O_{t+1} \cdots O_T \,|q_t = S_i, \mu) \\
&= \alpha_t(i) \times \beta_t(i)
\end{aligned}
$$

6.5 Viterbi 搜索算法

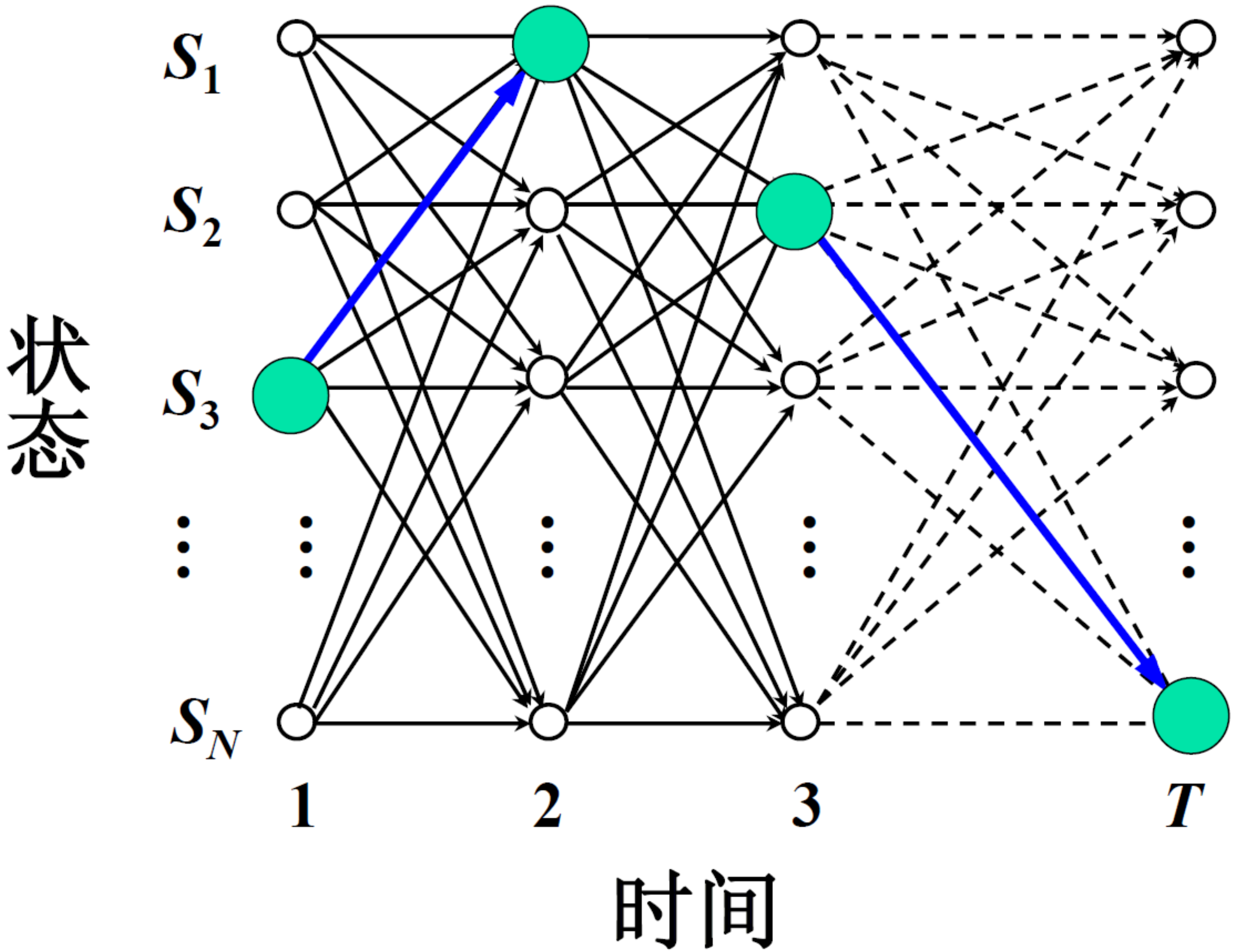
问题：

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 $a_{\hat{q}_t\hat{q}_{t+1}} = 0$。

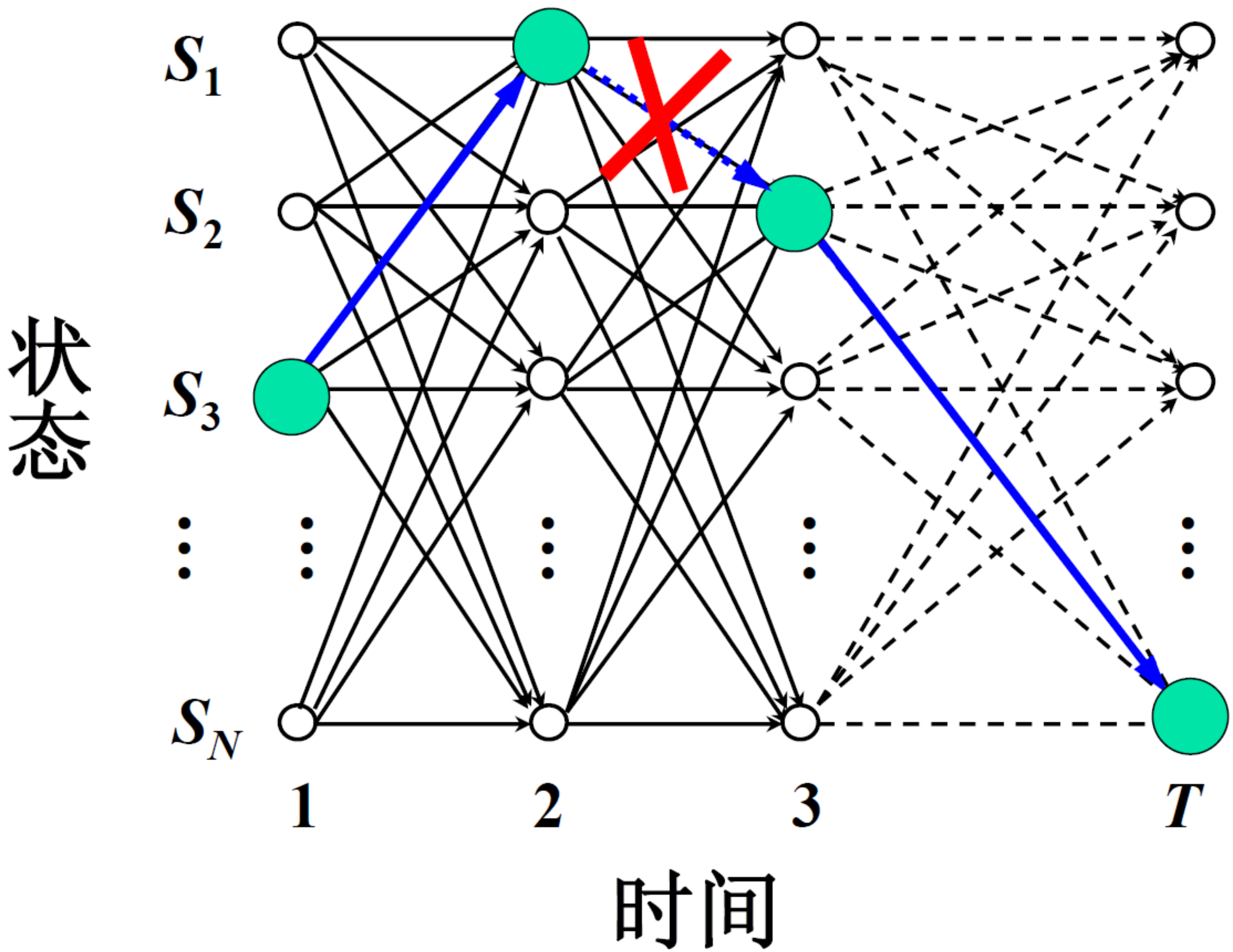
6.5 Viterbi 搜索算法



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：

$$\hat{Q} = \underset{Q}{\operatorname{argmax}}\, p(Q|O,\mu) \tag{6-21}$$

➤ 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 $O = O_1O_2\cdots O_t$ 的最大概率：

$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} p(q_1,q_2,\cdots,q_t = S_i, O_1O_2\cdots O_t|\mu) \tag{6-22}$$

6.5 Viterbi 搜索算法

递归计算：$\delta_{t+1}(j)=\max\limits_{i}\left[\delta_t(i)\cdot a_{ij}\right]\cdot b_j(O_{t+1})$　　(6-23)

◆ 算法6.3：Viterbi 算法

(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：

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

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

记忆回退路径

6.5 Viterbi 搜索算法

◆ 算法6.3：Viterbi 算法

(3) 结束:

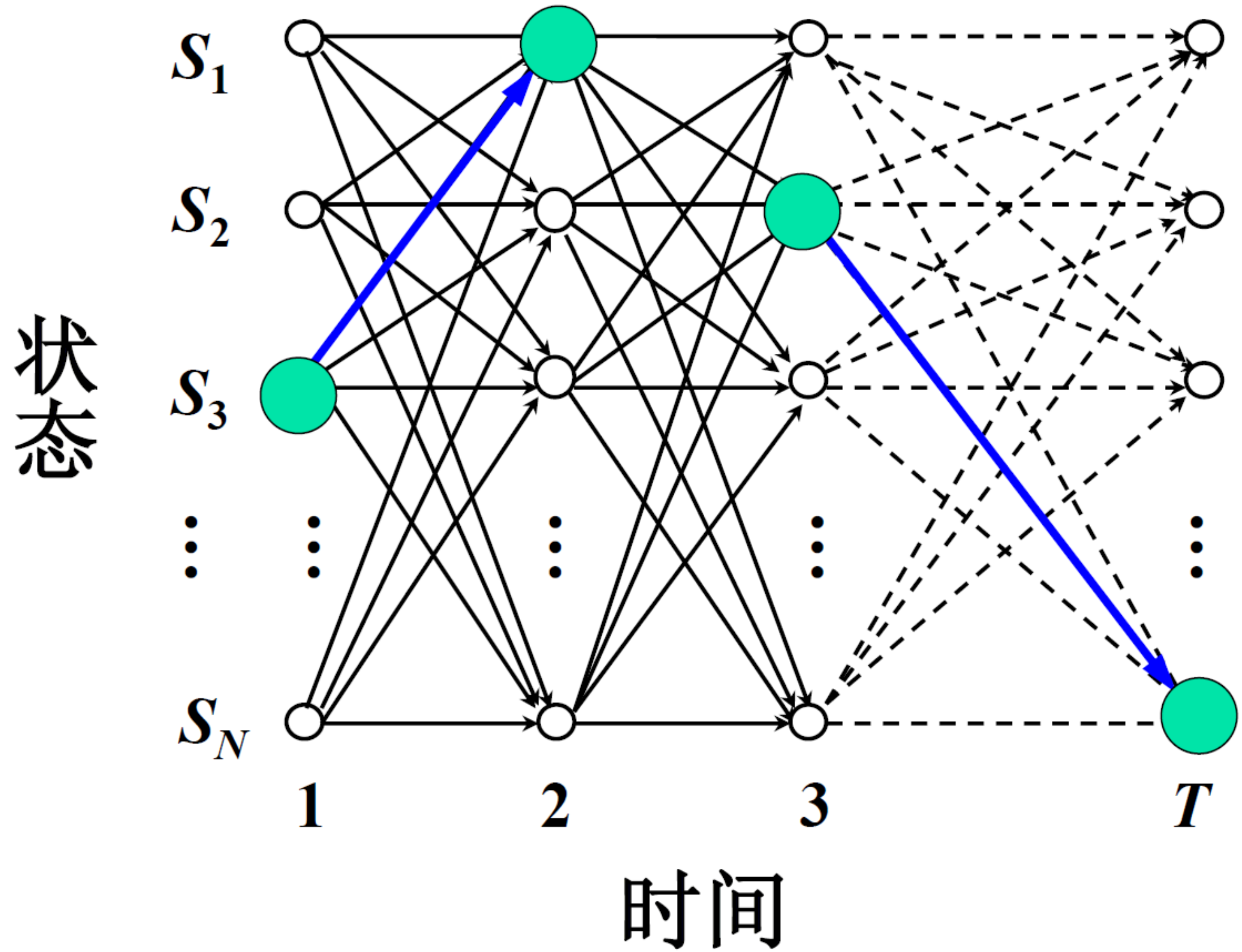
$$\hat{Q}_T = \underset{1 \le i \le N}{\operatorname{argmax}}[\delta_T(i)], \quad \hat{p}\left(\hat{Q}_T\right) = \max_{1 \le i \le N}[\delta_T(i)]$$

(4) 通过回溯得到路径(状态序列):

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

算法的时间复杂性：$O(N^2T)$

6.5 Viterbi 搜索算法



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 $O = O_1 O_2 \cdots O_T$，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 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，即 $\hat{\mu} = \max_{\mu} p(O|\mu)$

前向后向算法

(Baum-Welch or forward-backward procedure)

6.6 参数学习

◆ 最大似然估计

如果产生观察序列 O 的状态 $Q = q_1 q_2 \cdots q_T$ 已知，可以用最大似然估计来计算 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，$\delta(x, y)$ 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

◆ 最大似然估计

类似的

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

$$= \frac{\sum_{t=1}^{T} \delta\left(q_t, S_j\right) \times \delta(O_t, v_k)}{\sum_{t=1}^{T} \delta\left(q_t, S_j\right)} \qquad (6\text{-}25)$$

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

◆ 期望最大化算法(Expectation-Maximization,EM)

基本思想：初始化时随机地给模型的参数赋值（该赋值遵循模型对参数的限制如：从某一状态出发的转移概率总和为1），得到模型 μ_0，然后根据 μ_0 可以得到模型中隐变量的期望。例如，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实际次数，这样可以得到模型参数的新估计值，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，由此重新估计模型的参数。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

◆ Baum-Welch算法(前向后向算法)

给定HMM模型 μ 和观察序列 $O = O_1 O_2 \cdots O_T$，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：

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

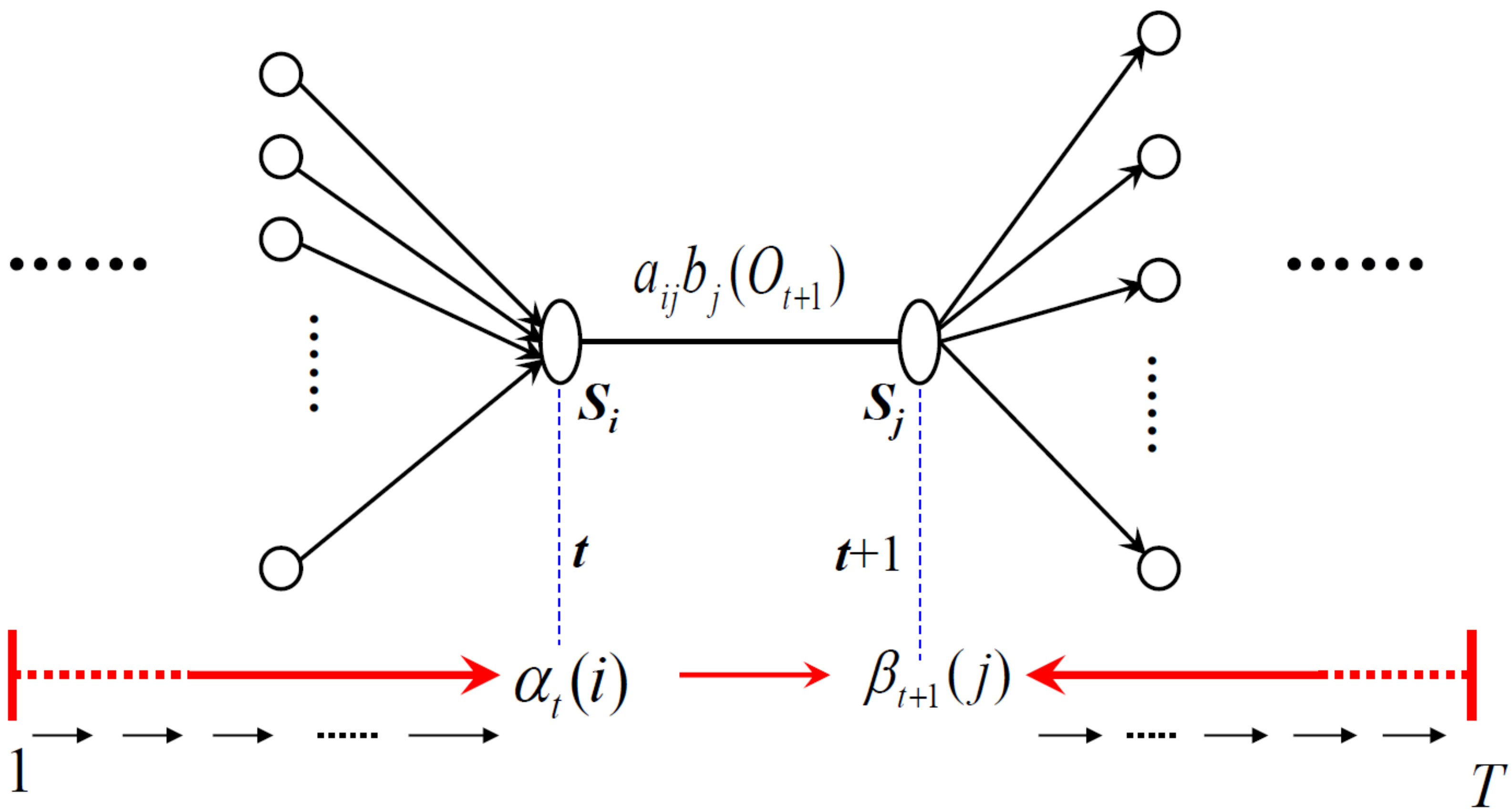
$$= \frac{\alpha_t(i) \times a_{ij} b_j(O_{t+1}) \times \beta_{t+1}(j)}{p(O \mid \mu)}$$

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

6.6 参数学习

◆ Baum-Welch算法(前向后向算法)

图解搜索过程:



6.6 参数学习

◆ Baum-Welch算法(前向后向算法)

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：

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{6-27}$$

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：

$$\pi_i = p(q_1 = S_i | O) = \gamma_1(i) \tag{6-28}$$

6.6 参数学习

◆ Baum-Welch算法(前向后向算法)

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

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (6\text{-}29)$$

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

$$= \frac{\sum_{t=1}^{T} \gamma_t(j) \times \delta(O_t, v_k)}{\sum_{t=1}^{T} \gamma_t(j)} \quad (6\text{-}30)$$

$$\boxed{\begin{aligned} \gamma_t(i) &= p(q_t = S_i \mid O, \mu) \\ \xi_t(i,j) &= p(q_t = S_i, q_{t+1} = S_j \mid O, \mu) \end{aligned}}$$

6.6 参数学习

◆算法6.4：Baum-Welch算法(前向后向算法)

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

$$\begin{cases} \sum_{i=1}^{N} \pi_i = 1 & \\ \sum_{j=1}^{N} a_{ij} = 1 & 1 \le i \le N \\ \sum_{k=1}^{M} b_i(k) = 1 & 1 \le i \le N \end{cases} \tag{6-31}$$

由此得到模型 μ_0，令 $i = 0$。

6.6 参数学习

◆ 算法6.4：Baum-Welch算法(前向后向算法)

(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)至(6-30) 重新估计 $\pi_i, a_{ij}, b_j(k)$ 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, Fundamentals of Speech Recognition, 1993, pp: 365-368]

- HMM开源工具

  —参阅 http://htk.eng.cam.ac.uk/

6.6 参数学习

◆ HMM使用中注意的问题

庄炳湟

美国工程院院士

IEEE Fellow

佐治亚理工学院教授

上海交通大学顾问教授

6.7 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?

6.7 应用举例

◆ 用 HMM 来解决这一问题：

- 状态转移模型
- 状态到观察序列的生成模型

思路：

如果把汉语自动分词结果作为观察序 $O = O_1 O_2 \cdots O_T$，

那么，我们要求解的是：$\hat{O} = \underset{O}{\operatorname{argmax}}\, p(O|\mu)$。

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

6.7 应用举例

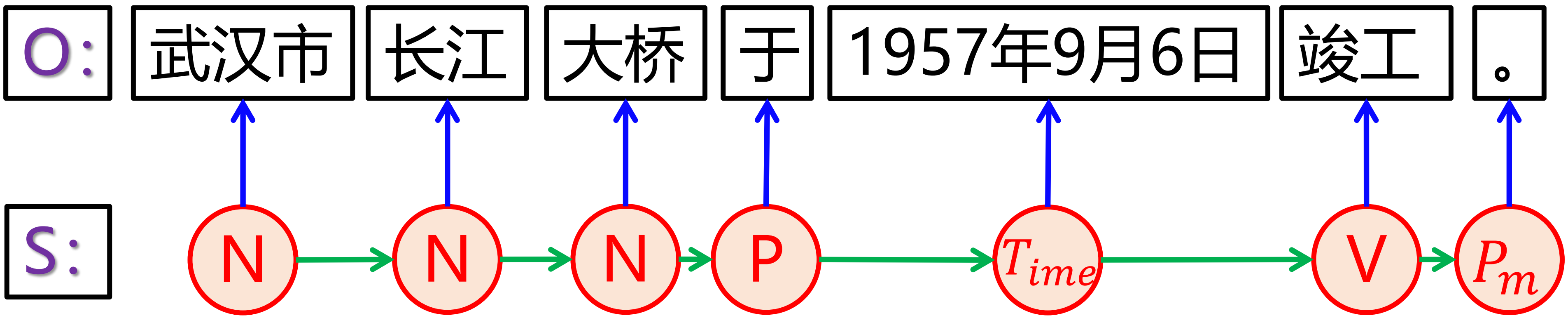
◆进一步解释:

(1) 估计HMM模型 $\mu = (A, B, \pi)$ 的参数;

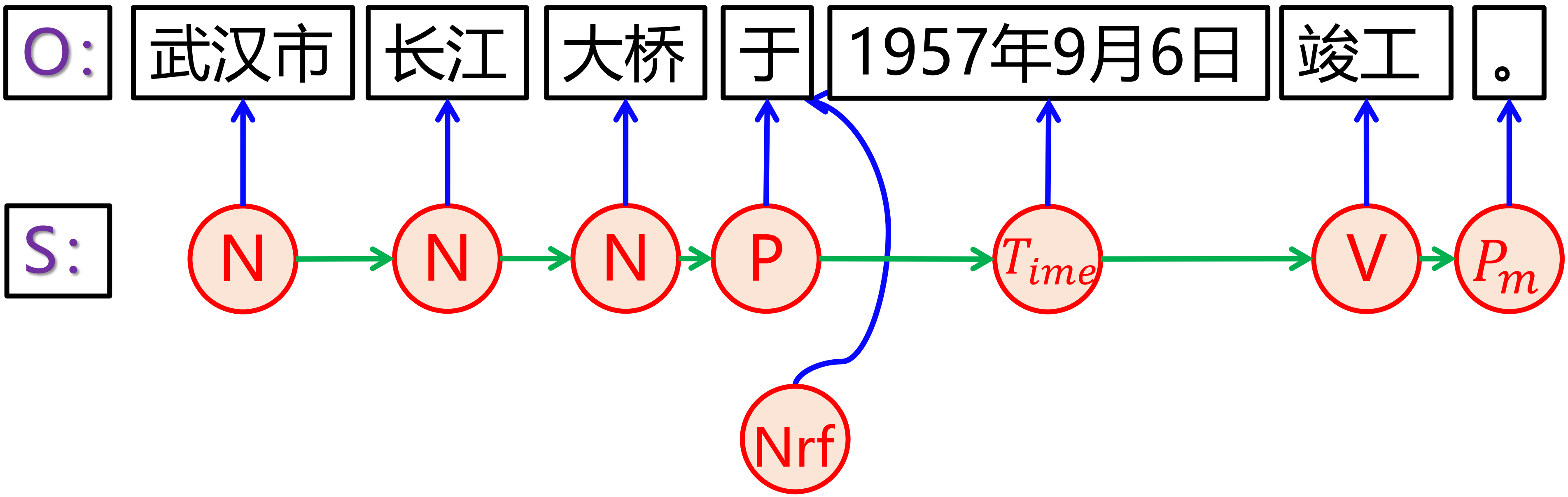
(2) 对于任意给定的一个输入句子及其可能的输出序列 O,求找所有可能的 O 中使概率 $p(O|\mu)$ 最大的解;

(3) 快速地选择“最优”的状态序列(词性序列),使其最好地解释观察序列(分词结果)。

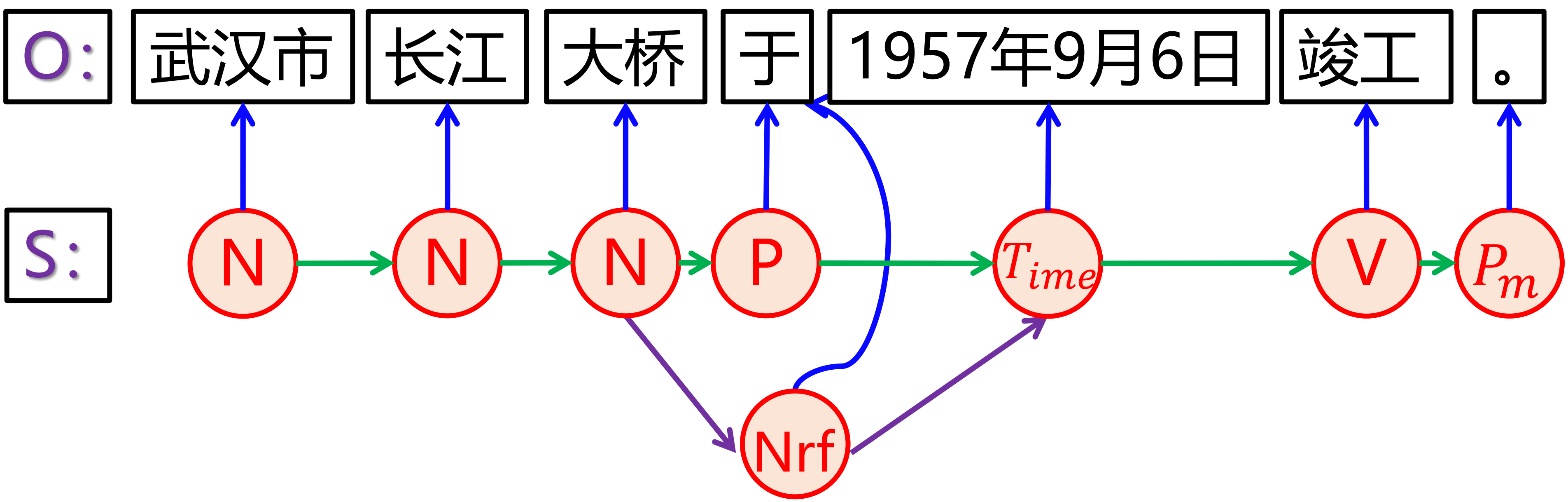
6.7 应用举例



6.7 应用举例



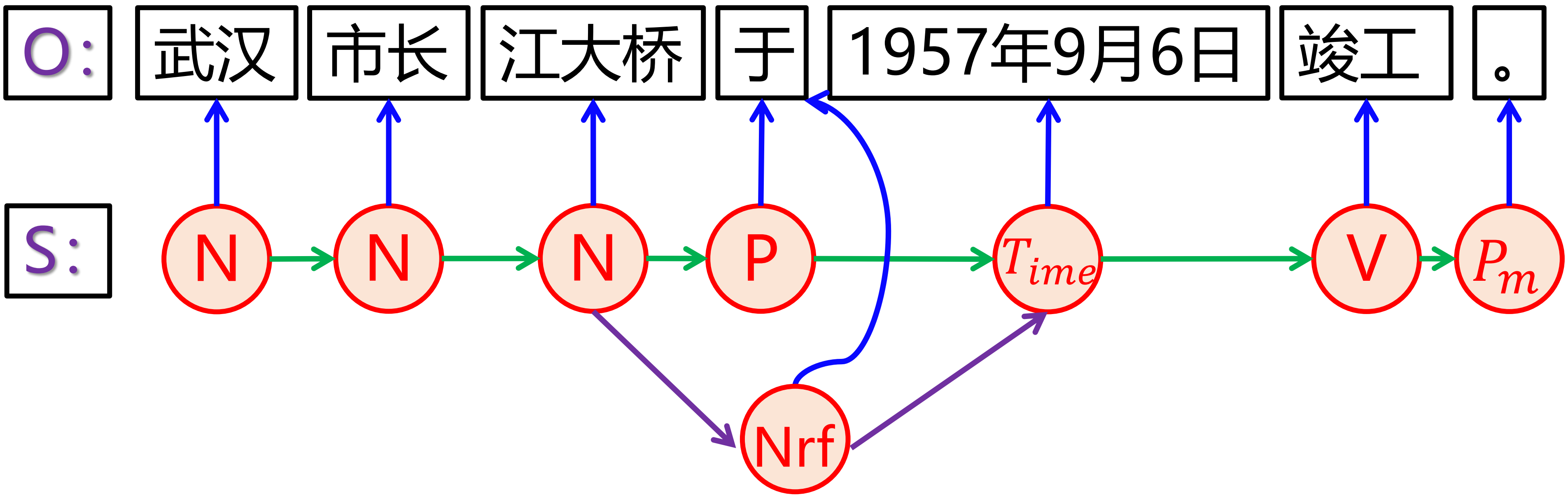
6.7 应用举例



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，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

◆问题1：模型参数——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，采用无监督学习方法：

(a) 获取词类个数(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

◆问题1：模型参数——参数估计

(2) 若有大规模分词和词性标注语料：监督学习方法

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

◆问题1：模型参数——参数估计

(2) 若有大规模分词和词性标注语料：监督学习方法

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

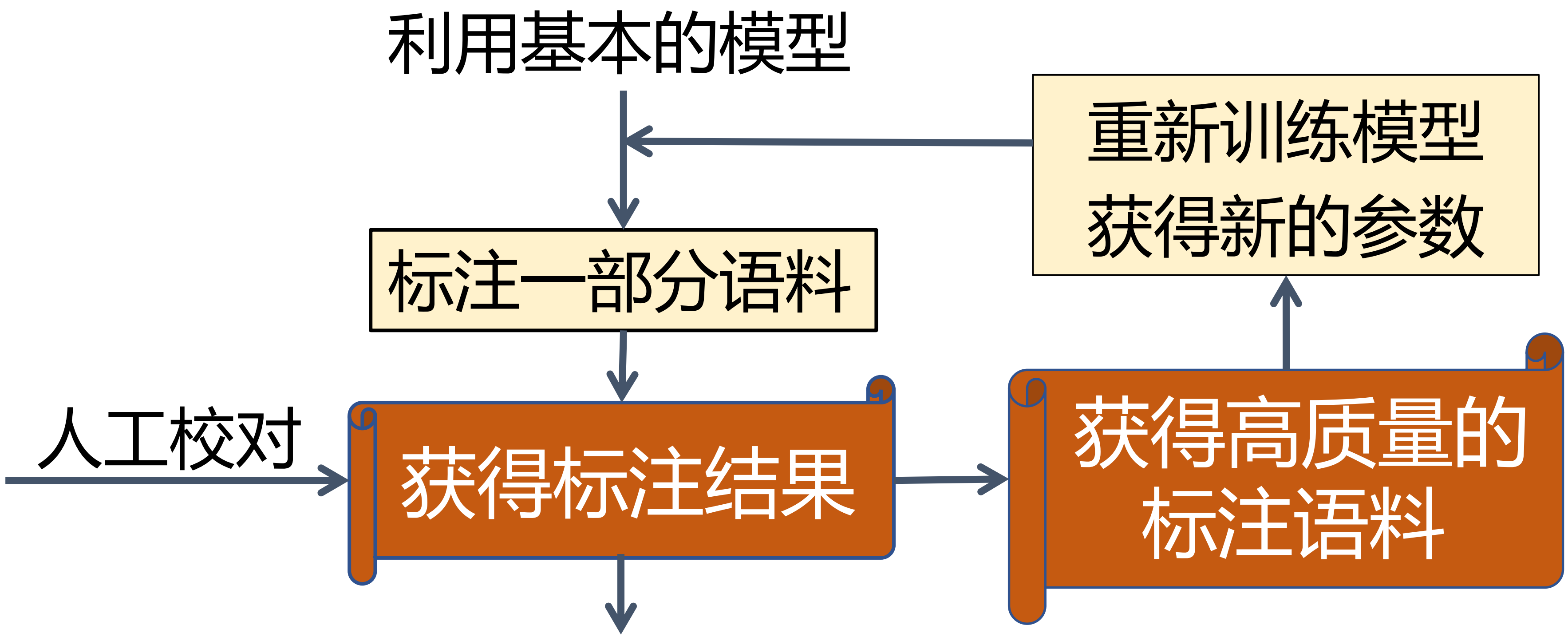
$$\bar{a}_{ij} = \frac{\text{从词类}\mathrm{POS}_i\text{转移到}\mathrm{POS}_j\text{的次数}}{\text{所有从状态}\mathrm{POS}_i\text{转移到另一}\mathrm{POS}(\text{包括}\mathrm{POS}_j)\text{的总数}}$$

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

6.7 应用举例

◆问题1：模型参数

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

◆ 问题2：如何获取观察序列?

➢ 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	33469	25475	24232	11453	4550	25670	184488	42674	
总计	34473	25479	24243	11512	4552	25674	184763	42724	

6.7 应用举例

◆ 问题2：如何获取观察序列?

➢ 分词实验：以“负责任”为例

$O_1 = w_1 w_2 =$ 负责/任 $\qquad p(O_1|\mu) = 5.4 \times 10^{-6}$

$O_2 = w_1 w_2 =$ 负/责任 $\qquad p(O_2|\mu) = 9.3 \times 10^{-4}$

$O_3 = w_1 w_2 w_3 =$ 负/责/任 $\qquad p(O_3|\mu) = 4.3 \times 10^{-6}$

$$\boxed{p(O_2|\mu) > p(O_1|\mu) > p(O_3|\mu)}$$

第二种切分结果可能性较大：负/ 责任

6.7 应用举例

◆问题3：求分词结果和词性标注结果

(1) 分词结果：根据模型参数 μ 和输出序列 O，计算所有可能的 O 中使概率 $p(O|\mu)$ 最大的解：

$$\hat{O} = \underset{O}{\operatorname{argmax}}\, p(O|\mu)$$

(2) 词性标注结果：根据模型参数 μ 和输出序列 O，计算“最优”状态序列(词性序列)，使其最好地解释观察序列(分词结果)：

$$\hat{Q} = \underset{Q}{\operatorname{argmax}}\, p(Q|O,\mu)$$

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%。

请参阅：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

➢ 词性标注性能测试：

(1) 采用有监督的参数估计方法；

(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

请参阅：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.

6.7 应用举例

- 如何实际应用隐马尔可夫模型?
- 对于一个给定的句子，如何计算概率?

本章小结

◆HMM的构成概念

- 状态数
- 输出符号数
- 初始状态的概率分布
- 状态转移的概率
- 输出概率

本章小结

- HMM 的三个基本问题：

  (1) 快速计算给定模型的观察序列的概率

  —前向算法或后向算法

  (2) 求最优状态序列

  — Viterbi 算法

  (3) 参数估计

  — Baum-Welch (Forward-Backward) 算法

- 模型应用实例

习题

6-1. 下载 HTK (http://htk.eng.cam.ac.uk/)，了解相应工具的使用方法。

6-2. 利用HTK工具，实现一个简单的汉语音字转换程序或汉语分词与词性标注程序。

编程工具

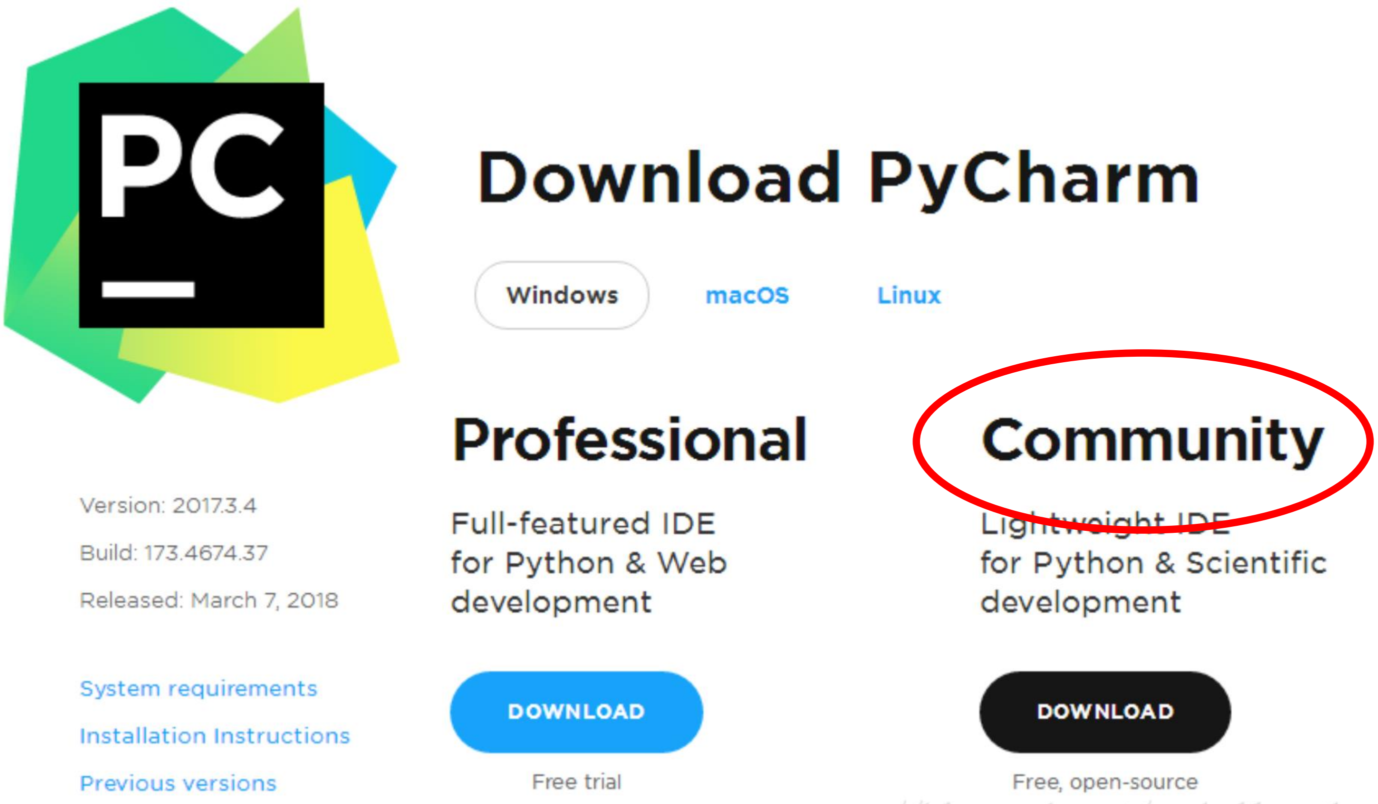
◆ 语言：Python

- Python官网：www.python.org
  —指令pip安装各种包
- 开源的Python发行版本：Anaconda
  —官网：https://www.anaconda.com/download/
  —指令conda安装各种包

编程工具

◆ Python集成开发环境(IDE)

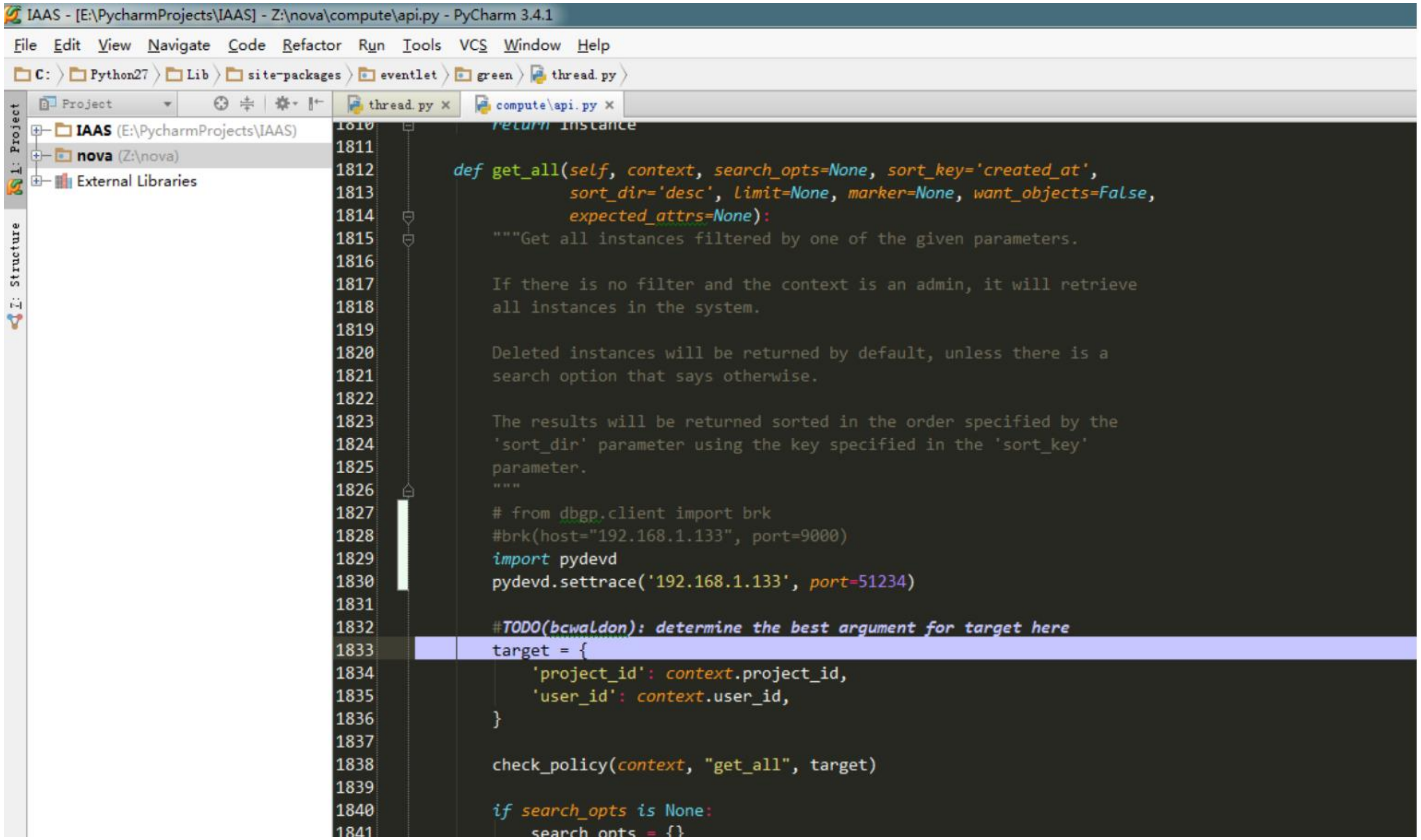
➢ PyCharm官网：https://www.jetbrains.com/pycharm/



编程工具

◆ Python集成开发环境(IDE)

➢ PyCharm



谢谢！