Fundamentals of Hidden Markov Model (HMM)

Ren-yuan Lyu
Dept of Computer Science & Information Engineering,
Chang Gung University,
Guei-shan, Taoyuan, Taiwan
rylyu@mail.cgu.edu.tw

Reference:

1. X. Huang, “Spoken Language Processing”, Chap 8
2. L. Rabiner, “Fundamentals of Speech Recognition”, Chap 6
3. HTK Book, http://htk.eng.cam.ac.uk/

Hidden Markov Model (HMM)

- A ***Hidden Markov Model*** is a (powerful) statistical model to describe very complex random processes (sequences) with time-varying characteristics by
  - Building parametric models,
  - Incorporating Dynamic Programming,
  - Providing a unified scheme for pattern segmentation and pattern classification (recognition).

Applications of HMM

- Automatic Speech Recognition
- Statistical Language Modeling
- Machine Translation
- Bioinformatics
- …..
- …..

Definition of HMM

- The "Hidden" State Sequence ***S***(t)
  - A HMM can be viewed as a *doubly embedded random process* with a "***hidden***" random process, usually called the state process S(t), which is not directly observable.
  - S(t) is assumed a (1st order) ***Markov*** chain
- The "Observable" Data Sequence ***O***(t)
  - The observable random process O(t) in a HMM is probabilistically associated with the hidden random process S(t).
  - O(t) is assumed dependent only on the value of S(t) at the same time index t.

Ren-yuan Lyu
@ Chang Gung University

Specifications of HMM

- The State sequence (process), S(t),
  - with the state space Ω_S

$t \in Z$: a set of the time index,
for a finite duration random process, $Z = [1 ... T]$

$S \in \Omega_S$: a set of state value, also called a state space,
usually being a finite set with N distinct values,
A typical example is $\Omega_S = \{1, 2, 3, ..., N\}$

- The Observation sequence, O(t),
  - with the observation space Ω_O

$t \in Z$: a set of the time index,

for a finite duration random process, $Z = [1 \ldots T]$

$O \in \Omega_O$: a set of observation value, also called a observation space,

It may be a finite set with M distinct values,

A typical example is $\Omega_O = \{1, 2, 3, \ldots, M\}$

It could also be a continuous set of values.

$e.g.,\ \Omega_O = R$: the real number

Or even a continuous vector space

$e.g.,\ \Omega_O = R^D$

- The state transitional probability distribution **A**=[a_{ij}], and initial state probability distribution **π**=[π_{i}]

By assumming $S(t)$ be a 1st-order Markov chain,
the state transitional probability, $P(S(t)=j\,|\,S(t-1)=i)$, is time-invariant,
and can statistically completely describe the random process $S(\mathrm{t})$,
with the initial state probability $P(S(1)=i)$.

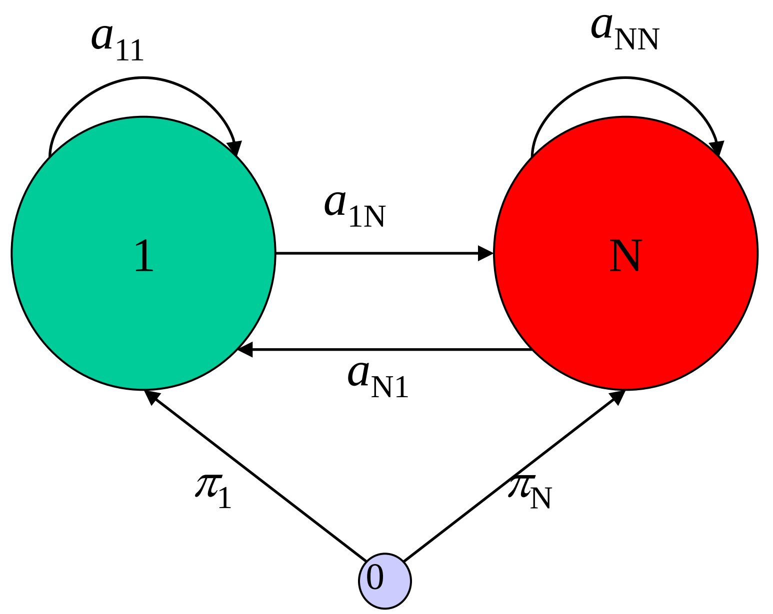
$$a_{ij} \equiv P(S(t)=j\,|\,S(t-1)=i), \quad i,j \in \Omega_S = \{1,2,3,\ldots,N\}$$

$$\pi_i \equiv P(S(1)=i)$$

$$\underline{\underline{A}} = [a_{ij}] = \begin{bmatrix} a_{11} & .. & a_{1N} \\ : & & : \\ a_{N1} & .. & a_{NN} \end{bmatrix}, \qquad \text{where} \sum_{j=1}^{N} a_{ij} = 1,\ \forall i \in \Omega_S$$

$$\vec{\pi} = [\pi_i] = \begin{bmatrix} \pi_1 \\ : \\ \pi_N \end{bmatrix}, \qquad \text{where} \sum_{i=1}^{N} \pi_i = 1$$

- The state diagram



- The (state-dependent) observation probability distribution, **B**= {$b_i(o)$}

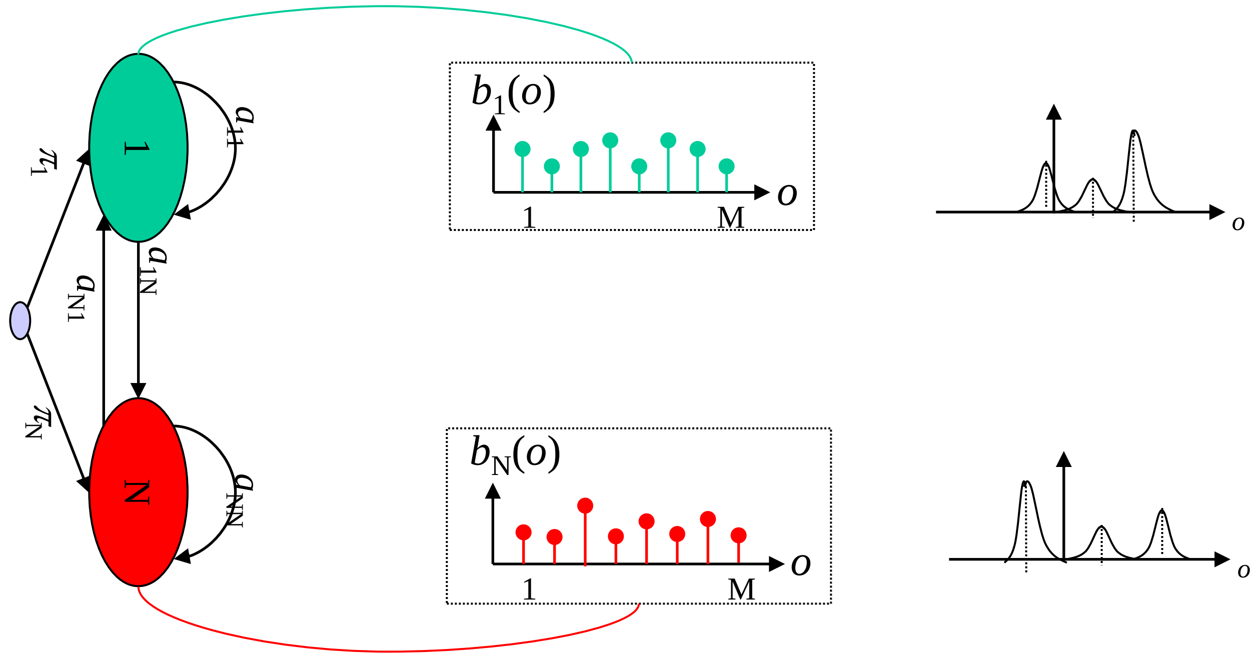
By assumming $O(t)$ be a state-dependent random process,
it is enough to specify $P(O(t)=o \mid S(t)=i)$ to completely describe $O(t)$,
as long as $S(t)$ is given.
Let

$$b_i(o) = P(O(t)=o \mid S(t)=i), \qquad i \in \Omega_S = \{1,2,3,\ldots,N\},$$
$$o \in \Omega_O = \{1,2,3,\ldots,M\},$$

$$\underline{B} = \{b_i(o)\} = \begin{Bmatrix} b_1(1) & \ldots. & b_1(M) \\ : & & : \\ b_N(1) & \ldots. & b_N(M) \end{Bmatrix}, \quad \text{where } \sum_{o=1}^{M} b_i(o) = 1, \ \forall i \in \Omega_S$$

If $o \in \Omega_O = R$(the set of real number),
then $b_i(o)$ should be a continuous probability density function,
e.g., the Gaussian distribution function
or the others, like Mixtured Gaussian distribution function

- The state-dependent observation distributions

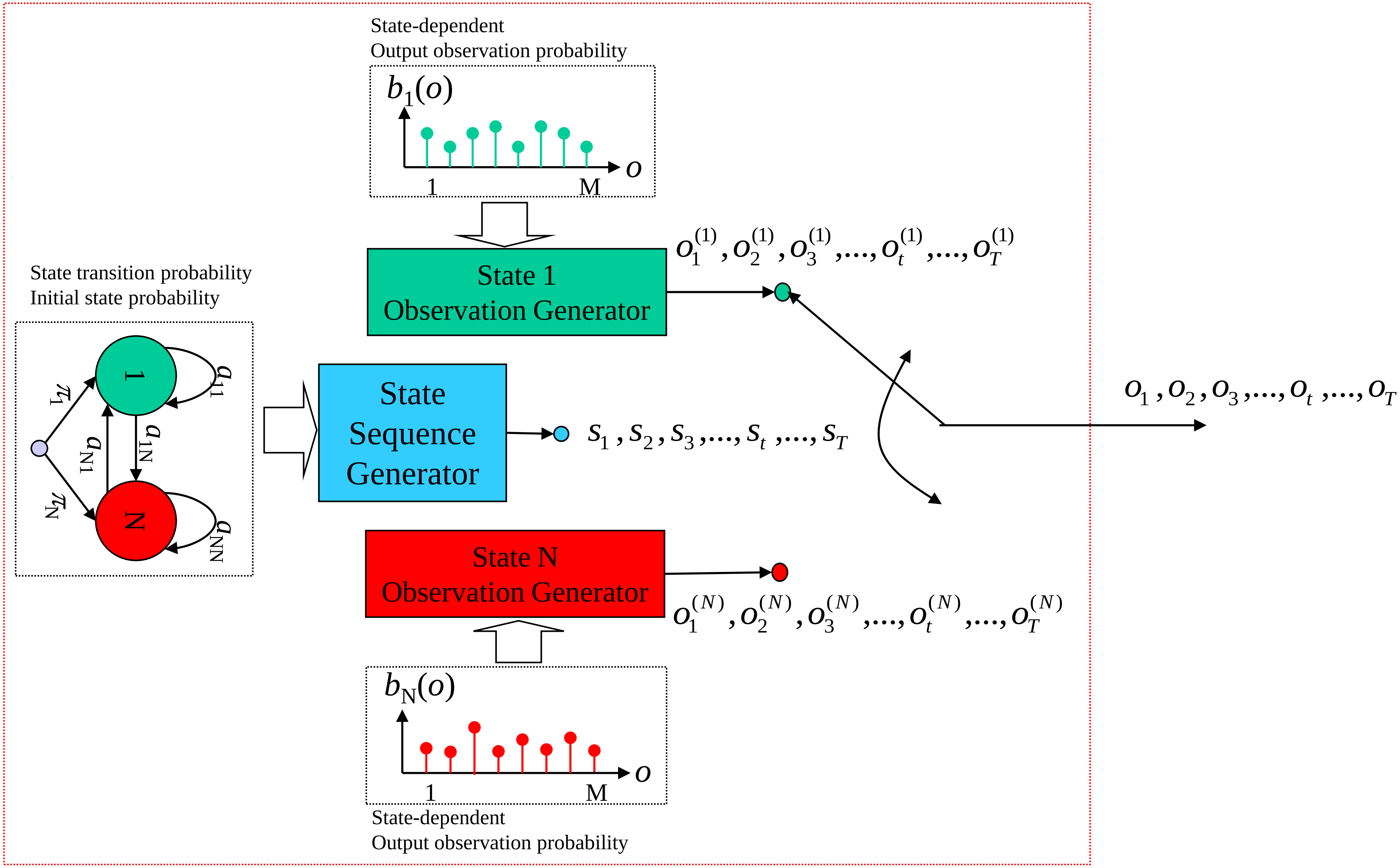


Hidden Markov Machine

- To put all the elements of HMM together, we design a Hidden Markov Machine as follows:
  - There is a ***state sequence generator***, which generates random sequence S(t) under the control of HMM parameters $\{\boldsymbol{\pi}, \mathbf{A}\}$.
  - There are N state-dependent ***observation generators*****,** each of which generates observation sequence under the control of HMM parameter $\underline{\mathbf{B}}=\{b_i(o)\}$
  - The state value s_t, generated by the ***state sequence generator*** at time t, determines which observation generator's output, $o^{(i)}{}_t$, will be used as the output observation o_t at time t.

Ren-yuan Lyu
@ Chang Gung University

Hidden Markov Machine



3 Basic Problems of HMM

- The Evaluation Problem

$$P(o_1, o_2, o_3, \dots, o_t, \dots, o_T) = ?$$

- The Estimation (Learning) Problem

$$(\vec{\pi}^*, \underline{\underline{A}}^*, \underline{B}^*) = \underset{\forall(\vec{\pi}, \underline{\underline{A}}, \underline{B})}{\boldsymbol{Argmax}} \; P(o_1, o_2, o_3, \dots, o_t, \dots, o_T \mid \vec{\pi}, \underline{\underline{A}}, \underline{B}) = ?$$

- The Decoding Problem

$$(s_1^*, s_2^*, s_3^*, \dots, s_t^*, \dots, s_T^*) = \underset{\forall(s_1, s_2, s_3, \dots, s_t, \dots, s_T)}{\boldsymbol{Argmax}} \; P(s_1, s_2, s_3, \dots, s_t, \dots, s_T \mid o_1, o_2, o_3, \dots, o_t, \dots, o_T) = ?$$

The Evaluation Problem of HMM

- Given a HMM, with parameters {π, **A**, **B**},

$$P(o_1, o_2, o_3, \ldots, o_t, \ldots, o_T) = ?$$

Ren-yuan Lyu
@ Chang Gung University
2013/10/29

Markov Assumption

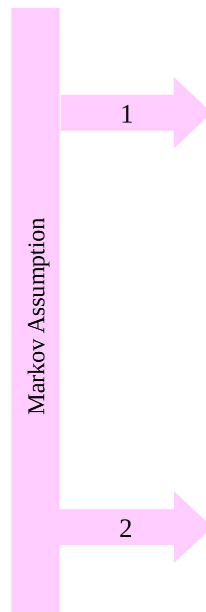
$$\underline{O} \equiv o_1, o_2, \ldots o_t, \ldots, o_T$$
$$\underline{S} \equiv s_1, s_2, \ldots s_t, \ldots, s_T$$

$$P(\underline{O}) = \sum_{\underline{S}} P(\underline{O}, \underline{S}) = \sum_{\underline{S}} P(\underline{O} \mid \underline{S}) \cdot P(\underline{S})$$

1. The current state depends only on the previous state
2. The observation depends only on the current state

$$\begin{aligned} P(\underline{S}) = P(s_1, s_2, \ldots s_t, \ldots, s_T) &= P(s_1) \\ &\cdot P(s_2 \mid s_1) \\ &\cdot P(s_3 \mid s_2, \cancel{s_1}) \\ &\ldots. \\ &\cdot P(s_t \mid s_{t-1}, \cancel{s_{t-2}, \ldots, s_2, s_1}) \\ &\ldots. \\ &\cdot P(s_T \mid s_{T-1}, \cancel{s_{T-2}, \ldots, s_t, \ldots, s_2, s_1}) \end{aligned}$$

$$\begin{aligned} P(\underline{O} \mid \underline{S}) = P(o_1, o_2, \ldots o_t, \ldots, o_T \mid s_1, s_2, \ldots s_t, \ldots, s_T) &= P(o_1 \mid s_1, \cancel{s_2, \ldots, s_t, \ldots, s_T}) \\ &\cdot P(o_2 \mid \cancel{o_1, s_1}, s_2 \cancel{, \ldots, s_t, \ldots, s_T}) \\ &\cdots \\ &\cdot P(o_t \mid \cancel{o_{t-1}, \ldots o_2, o_1, s_1, s_2, \ldots} s_t \cancel{, \ldots, s_T}) \\ &\cdots \\ &\cdot P(o_T \mid \cancel{o_{T-1}, \ldots o_2, o_1, s_1, s_2, \ldots s_t, \ldots} s_T) \end{aligned}$$



$$\begin{aligned} P(\underline{S}) = P(s_1, s_2, \ldots s_t, \ldots, s_T) &= P(s_1) \\ &\cdot P(s_2 \mid s_1) \\ &\cdot P(s_3 \mid s_2) \\ &\ldots. \\ &\cdot P(s_t \mid s_{t-1}) \\ &\ldots. \\ &\cdot P(s_T \mid s_{T-1}) \end{aligned}$$

$$\begin{aligned} P(\underline{O} \mid \underline{S}) = P(o_1, o_2, \ldots o_t, \ldots, o_T \mid s_1, s_2, \ldots s_t, \ldots, s_T) &= P(o_1 \mid s_1) \\ &\cdot P(o_2 \mid s_2) \\ &\cdots \\ &\cdot P(o_t \mid s_t) \\ &\cdots \\ &\cdot P(o_T \mid s_T) \end{aligned}$$

Ren-yuan Lyu
@ Chang Gung University

Direct computation of $P(\underline{O})$

$$\underline{O} \equiv o_1, o_2, \ldots o_t, \ldots, o_T$$
$$\underline{S} \equiv s_1, s_2, \ldots s_t, \ldots, s_T$$

$$P(\underline{O}) = \sum_{\underline{S}} P(\underline{O}, \underline{S}) = \sum_{\underline{S}} P(\underline{S}) \cdot P(\underline{O} \mid \underline{S})$$

$$= \sum_{\underline{S}} \begin{Bmatrix} P(s_1) \quad \cdot P(s_2 \mid s_1) \cdots \cdot P(s_t \mid s_{t-1}) \cdots P(s_T \mid s_{T-1}) \\ \cdot P(o_1 \mid s_1) \cdot P(o_2 \mid s_2) \cdots \cdot P(o_t \mid s_t) \ \cdots P(o_T \mid s_T) \end{Bmatrix}$$

$$\begin{cases} P(s_1) = \pi_{s_1} = a_{s_0 s_1}, s_0 \equiv 0 = \text{"}B\text{"}: \text{the beginning state at } t = 0 \\ P(s_t \mid s_{t-1}) = a_{s_{t-1} s_t} \\ P(o_t \mid s_t) = b_{s_t}(o_t) \end{cases}$$

$$= \sum_{\underline{S}} \begin{Bmatrix} a_{s_0 s_1} \quad \cdot a_{s_1 s_2} \cdots \quad a_{s_{t-1} s_t} \cdots \quad a_{s_{T-1} s_T} \\ \cdot b_{s_1}(o_1) \cdot b_{s_2}(o_2) \cdots b_{s_t}(o_t) \cdots b_{s_T}(o_T) \end{Bmatrix}$$

$$= \sum_{s_T=1}^{N} \ldots \sum_{s_t=1}^{N} \ldots \sum_{s_2=1}^{N} \sum_{s_1=1}^{N} \begin{Bmatrix} a_{s_0 s_1} \quad \cdot a_{s_1 s_2} \cdots \quad a_{s_{t-1} s_t} \cdots \quad a_{s_{T-1} s_T} \\ \cdot b_{s_1}(o_1) \cdot b_{s_2}(o_2) \cdots b_{s_t}(o_t) \cdots b_{s_T}(o_T) \end{Bmatrix}$$

Totally, there are N^T possible sequences for $\underline{S}$,
i.e., N^T terms of { .. } need to be calculated.

Ren-yuan Lyu
@ Chang Gung University

Forward/Backward Algorithm

- The forward algorithm

$$P(\underline{O}) = \sum_{s_T=1}^{N} \cdots \sum_{s_t=1}^{N} \cdots \sum_{s_2=1}^{N} \sum_{s_1=1}^{N} \underbrace{\underbrace{\underbrace{\underbrace{a_{s_0 s_1} \cdot b_{s_1}(o_1)}_{\alpha_{s_1}(1)} \cdot a_{s_1 s_2} \cdot b_{s_2}(o_2)}_{\alpha_{s_2}(2)} \cdots a_{s_{t-1} s_t} \cdot b_{s_t}(o_t)}_{\alpha_{s_t}(t)} \cdots a_{s_{T-1} s_T} \cdot b_{s_T}(o_T)}_{\alpha_{s_T}(T)}$$

$$P(\underline{O}) = \sum_{s_T=1}^{N} .. \sum_{s_t=1}^{N} .. \sum_{s_2=1}^{N} \sum_{s_1=1}^{N} a_{s_0 s_1} \cdot b_{s_1}(o_1) \cdot a_{s_1 s_2} \cdot b_{s_2}(o_2) \cdots a_{s_{t-1} s_t} \cdot b_{s_t}(o_t) \cdots a_{s_{T-1} s_T} \cdot b_{s_T}(o_T)$$

$$\alpha_{s_1}(1) = a_{s_0 s_1} \cdot b_{s_1}(o_1) \qquad , s_1 \in [1..N], s_0 \equiv 0, a_{s_0 s_1} \equiv \pi_{s_1}$$

$$\alpha_{s_2}(2) = \sum_{s_1=1}^{N} \alpha_{s_1}(1) \cdot a_{s_1 s_2} \cdot b_{s_2}(o_2) \qquad , s_2 \in [1..N]$$

$$\vdots$$

$$\alpha_{s_t}(t) = \sum_{s_{t-1}=1}^{N} \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1} s_t} \cdot b_{s_t}(o_t) \qquad , s_t \in [1..N]$$

$$\vdots$$

$$\alpha_{s_T}(T) = \sum_{s_{T-1}=1}^{N} \alpha_{s_{T-1}}(T-1) \cdot a_{s_{T-1} s_T} \cdot b_{s_T}(o_T) \qquad , s_T \in [1..N]$$

$$\sum_{s_T=1}^{N} \alpha_{s_T}(T) ==== P(\underline{O})$$

The computation complexity reduces to be $T \cdot N^2$

Ren-yuan Lyu
@ Chang Gung University

- The meaning of the forward probability

$$\alpha_{s_1}(1) = a_{s_0 s_1} \cdot b_{s_1}(o_1)$$
$$= P(s_1 | s_0) \cdot P(o_1 | s_1)$$
$$= P(s_1) \cdot P(o_1 | s_1) \text{.........}\{\because s_0 \equiv 0 \text{ is a determined value}\}$$
$$= P(o_1, s_1)$$

$$\alpha_{s_2}(2) = \sum_{s_1=1}^{N} \alpha_{s_1}(1) \cdot a_{s_1 s_2} \cdot b_{s_2}(o_2)$$
$$= \sum_{s_1=1}^{N} P(o_1, s_1) \cdot P(s_2 | s_1) \cdot P(o_2 | s_2)$$

$$\begin{cases} \because \\ P(o_1, o_2, s_1, s_2) = P(o_1, s_1) \cdot P(o_2, s_2 | o_1, s_1) \\ P(o_2, s_2 | o_1, s_1) = P(s_2 | o_1, s_1) \cdot P(o_2 | o_1, s_1, s_2) \\ P(s_2 | o_1, s_1) = P(s_2 | s_1) \text{... this is Markov assumption} \\ P(o_2 | o_1, s_1, s_2) = P(o_2 | s_2) \text{... the assumption} \\ \qquad \text{that observation denpends only on current state} \end{cases}$$

$$= \sum_{s_1=1}^{N} P(o_1, o_2, s_1, s_2)$$
$$= P(o_1, o_2, s_2)$$

$$\alpha_{s_t}(t) = \sum_{s_{t-1}=1}^{N} \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t) = P(o_1, o_2, \ldots, o_t, s_t)$$

$$\alpha_{s_T}(T) = \sum_{s_{T-1}=1}^{N} \alpha_{s_{T-1}}(T-1) \cdot a_{s_{T-1}s_T} \cdot b_{s_T}(o_T) = P(o_1, o_2, \ldots, o_T, s_T)$$

$$\sum_{s_T=1}^{N} \alpha_{s_T}(T) = \sum_{s_T=1}^{N} P(o_1, o_2, \ldots, o_T, s_T) = P(o_1, o_2, \ldots, o_T) = P(\underline{O})$$

Ren-yuan Lyu
@ Chang Gung University

- The backward algorithm

$$P(\underline{O}) = \sum_{s_T=1}^{N} .. \sum_{s_t=1}^{N} .. \sum_{s_2=1}^{N} \sum_{s_1=1}^{N} a_{s_0 s_1} \cdot b_{s_1}(o_1) \cdot a_{s_1 s_2} \cdot b_{s_2}(o_2) \cdots a_{s_{t-1} s_t} \cdot b_{s_t}(o_t) \cdots a_{s_{T-1} s_T} \cdot b_{s_T}(o_T)$$

$$= \underbrace{\sum_{s_1=1}^{N} \underbrace{\sum_{s_2=1}^{N} .. \underbrace{\sum_{s_t=1}^{N} \underbrace{\sum_{s_{t+1}=1}^{N} .. \underbrace{\sum_{s_{T-1}=1}^{N} \underbrace{\sum_{s_T=1}^{N} b_{s_T}(o_T) \cdot a_{s_{T-1} s_T}}_{\beta_{s_{T-1}}(T-1)} \cdot b_{s_{T-1}}(o_{T-1}) \cdot a_{s_{T-2} s_{T-1}}}_{\beta_{s_{T-2}}(T-2)} \cdots b_{s_{t+1}}(o_{t+1}) \cdot a_{s_t s_{t+1}}}_{\beta_{s_t}(t)} \cdot b_{s_t}(o_t) \cdot a_{s_{t-1} s_t}}_{\beta_{s_{t-1}}(t-1)} \cdots b_{s_2}(o_2) \cdot a_{s_1 s_2}}_{\beta_{s_1}(1)} \cdot b_{s_1}(o_1) \cdot a_{s_0 s_1}}_{\beta_{s_0}(0)}$$

$$\beta_{s_{T-1}}(T-1) = \sum_{s_T=1}^{N} b_{s_T}(o_T) \cdot a_{s_{T-1}s_T}$$

$$\beta_{s_{T-2}}(T-2) = \sum_{s_{T-1}=1}^{N} \beta_{s_{T-1}}(T-1) \cdot b_{s_{T-1}}(o_{T-1}) \cdot a_{s_{T-2}s_{T-1}}$$

$$\vdots$$

$$\beta_{s_t}(t) = \sum_{s_{t+1}=1}^{N} \beta_{s_{t+1}}(t+1) \cdot b_{s_{t+1}}(o_{t+1}) \cdot a_{s_t s_{t+1}}$$

$$\vdots$$

$$\beta_{s_1}(1) = \sum_{s_2=1}^{N} \beta_{s_2}(2) \cdot b_{s_2}(o_2) \cdot a_{s_1 s_2}$$

$$\beta_{s_0}(0) = \sum_{s_1=1}^{N} \beta_{s_1}(1) \cdot b_{s_1}(o_1) \cdot a_{s_0 s_1} ==== P(\underline{O})$$

- The meaning of the backward probability

$$\beta_{s_{T-1}}(T-1) = \sum_{s_T=1}^{N} b_{s_T}(o_T) \cdot a_{s_{T-1}s_T}$$

$$= \sum_{s_T=1}^{N} P(o_T \mid s_T) \cdot P(s_T \mid s_{T-1})$$

$$= \sum_{s_T=1}^{N} P(o_T \mid s_T, s_{T-1}) \cdot P(s_T \mid s_{T-1})$$

$$= \sum_{s_T=1}^{N} P(o_T, s_T \mid s_{T-1})$$

$$= P(o_T \mid s_{T-1})$$

$$\beta_{s_{T-2}}(T-2) = \sum_{s_{T-1}=1}^{N} \beta_{s_{T-1}}(T-1) \cdot b_{s_{T-1}}(o_{T-1}) \cdot a_{s_{T-2}s_{T-1}} = P(o_{T-1}, o_T \mid s_{T-2})$$

$$\vdots$$

$$\beta_{s_t}(t) = \sum_{s_{t+1}=1}^{N} \beta_{s_{t+1}}(t+1) \cdot b_{s_{t+1}}(o_{t+1}) \cdot a_{s_t s_{t+1}} = P(o_{t+1}, o_{t+2}, \ldots, o_{T-1}, o_T \mid s_t)$$

$\vdots$

$$\beta_{s_1}(1) = \sum_{s_2=1}^{N} \beta_{s_2}(2) \cdot b_{s_2}(o_2) \cdot a_{s_1 s_2} = P(o_2, o_3, \ldots, o_{T-1}, o_T \mid s_1)$$

$$\beta_{s_0}(0) = \sum_{s_1=1}^{N} \beta_{s_1}(1) \cdot b_{s_1}(o_1) \cdot a_{s_0 s_1} = P(o_1, o_2, o_3, \ldots, o_{T-1}, o_T \mid s_0)$$

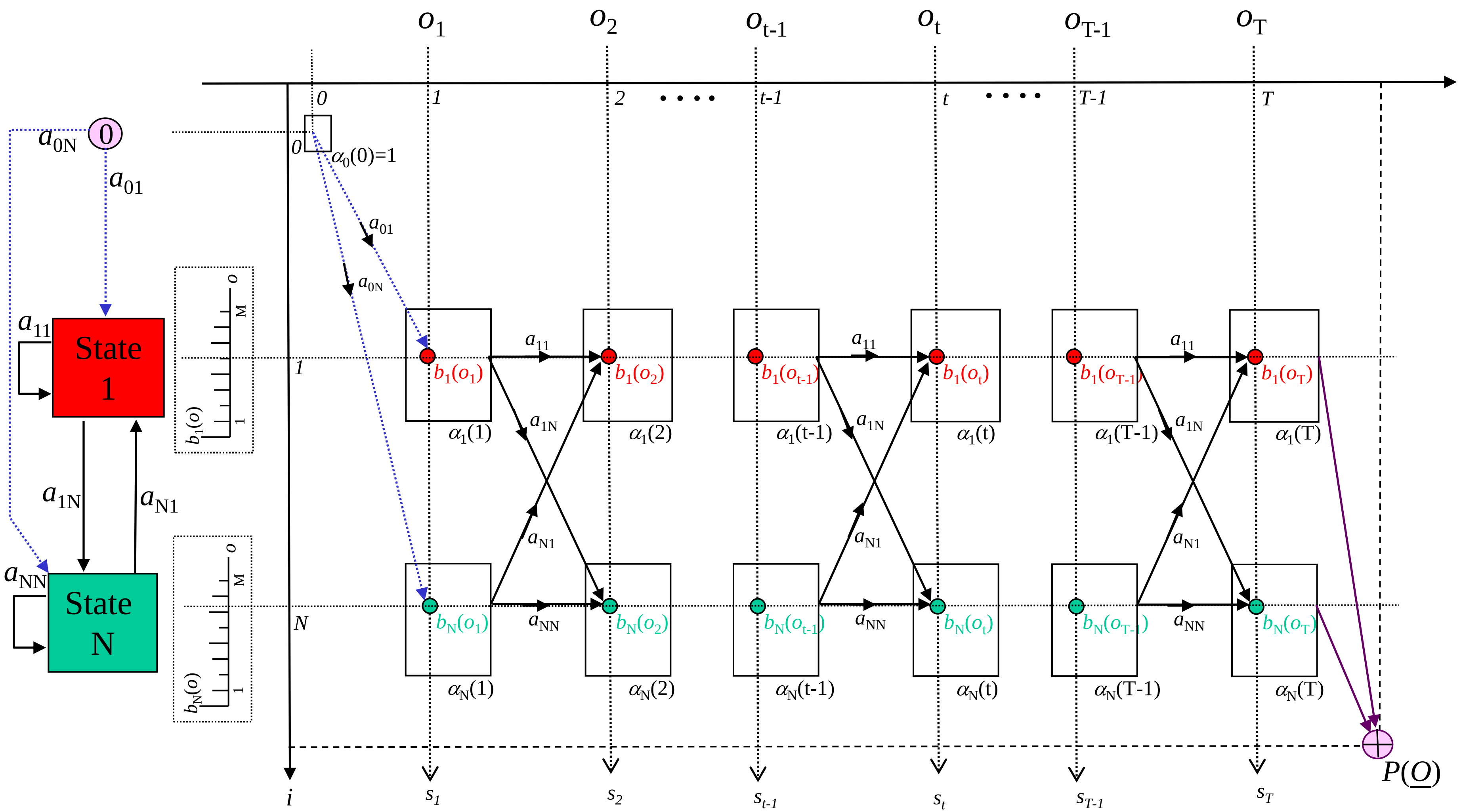
$$\begin{cases} \because s_0 (\equiv 0), \text{ is determined}, \\ \text{and independent of } o_1, \ldots o_T \end{cases}$$

$$= P(o_1, o_2, o_3, \ldots, o_{T-1}, o_T)$$

$$= P(\underline{O})$$

Ren-yuan Lyu
@ Chang Gung University
2013/10/29

Trellis view of the *forward* algorithm



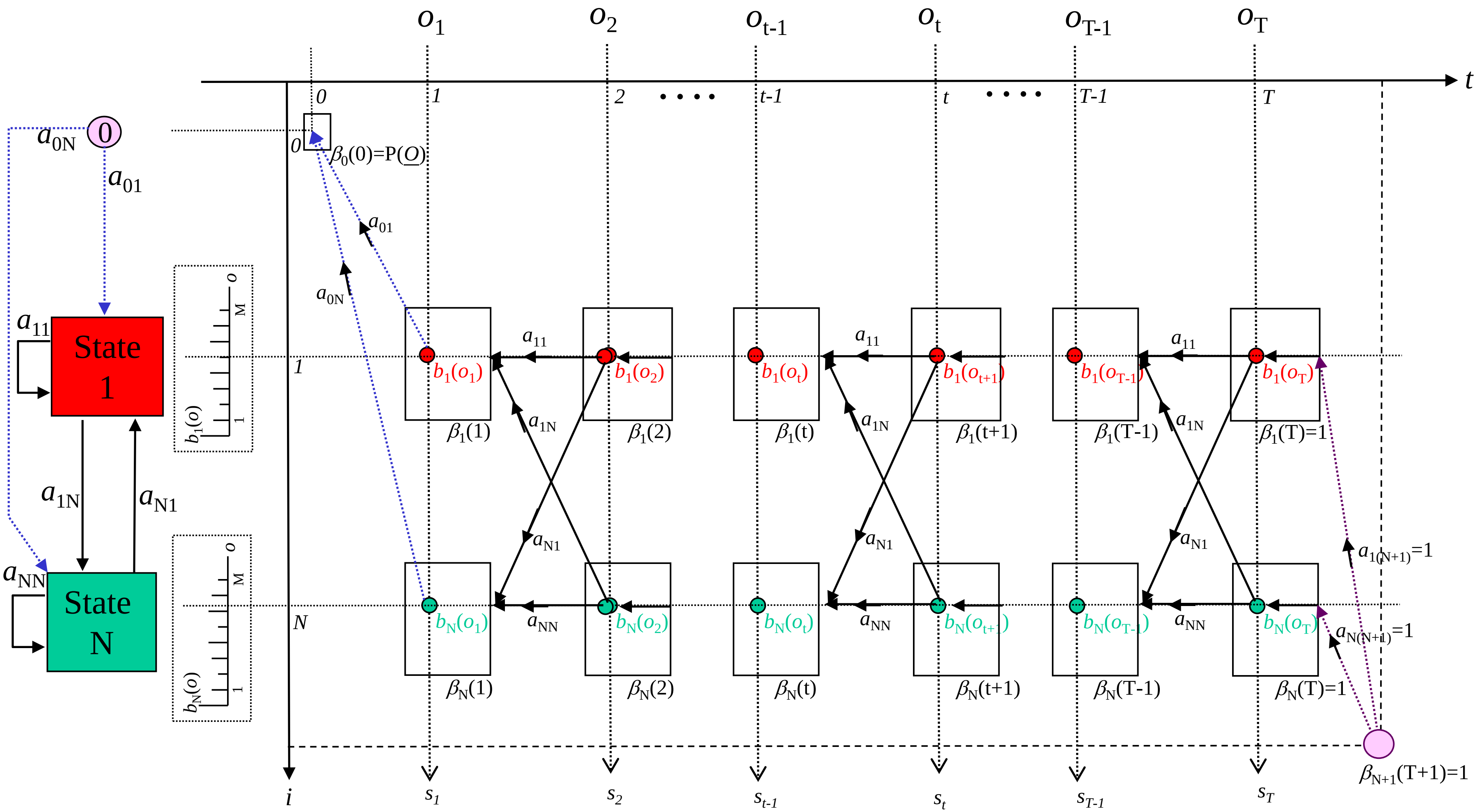
$$\alpha_{s_t}(t) = \sum_{s_{t-1}=1}^{N} \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t) \qquad , s_t \in [1..N]$$

Ren-yuan Lyu
@ Chang Gung University

2013/10/29

Trellis view of the *backward* algorithm

$$\beta_{s_t}(t) = \sum_{s_{t+1}=1}^{N} \beta_{s_{t+1}}(t+1) \cdot b_{s_{t+1}}(o_{t+1}) \cdot a_{s_t s_{t+1}}$$



The Estimation (Learning) Problem of HMM

- Given a HMM, with parameters {**π**, **A**, **B**}, and observation sequence(s)

$$o_1, o_2, o_3, \ldots, o_t, \ldots, o_T$$

How can we find "better" or even the "best" or "most likely" (new) parameters {**π***, **A***, **B***},
to help the HMMachine generate such an observation sequence with larger or even the maximal probability?
This problem can be reformulate as follows:

$$(\vec{\pi}^*, \underline{\underline{A}}^*, \underline{B}^*) = \underset{\forall(\vec{\pi}, \underline{\underline{A}}, \underline{B})}{\boldsymbol{Argmax}}\ P(o_1, o_2, o_3, \ldots, o_t, \ldots, o_T \mid \vec{\pi}, \underline{\underline{A}}, \underline{B}) = ?$$

Ren-yuan Lyu
@ Chang Gung University

The solution to the HMM Learning problem

$$\underline{S} \equiv s_1, s_2, \ldots, s_{t-1}, s_t, \ldots, s_T$$

$$\underline{O} \equiv o_1, o_2, \ldots, o_{t-1}, o_t, \ldots, o_T \qquad \text{where} \begin{cases} t \in [1..T] \\ s_t \in [1..N] \\ o_t \in [1..M] \\ \text{when } t=0, s_0 \equiv 0 \equiv \text{'Begin state'} \end{cases}$$

$$a_{s_{t-1}s_t} \equiv P(s_t \mid s_{t-1})$$

$$b_{s_t}(o_t) \equiv P(o_t \mid s_t)$$

$$\alpha_{s_t}(t) \equiv P(o_1, o_2, \ldots, o_t, s_t) = \sum_{s_{t-1}=1}^{N} \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t)$$

$$\beta_{s_t}(t) \equiv P(o_{t+1}, o_{t+2}, \ldots, o_{T-1}, o_T \mid s_t) = \sum_{s_{t+1}=1}^{N} a_{s_t s_{t+1}} \cdot b_{s_{t+1}}(o_{t+1}) \cdot \beta_{s_{t+1}}(t+1)$$

$$\alpha_{s_0}(0) \equiv 1$$

$$\beta_{s_T}(T) \equiv 1, \begin{cases} \text{in Huang's Textbook, } \beta_{s_T}(T) \equiv 1/N; \\ \text{in HTK, } \beta_{s_T}(T) \equiv a_{s_T(N+1)} \\ \qquad = P(S(T+1) = N+1 \equiv \text{'Exit state'} \mid S(T) = s_T) \end{cases}$$

The following are new : (Prove them)

$$\eta_{s_{t-1}s_t}(t) \equiv P(s_{t-1}, s_t, \underline{O}) = \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t) \cdot \beta_{s_t}(t)$$

$$\eta_{s_t}(t) \equiv P(s_t, \underline{O}) = \sum_{s_{t-1}=1}^{N} \eta_{s_{t-1}s_t}(t) = \alpha_{s_t}(t) \cdot \beta_{s_t}(t)$$

$$\eta_{s_{t-1}}(t-1) \equiv P(s_{t-1}, \underline{O}) = \begin{cases} \sum_{s_t=1}^{N} \eta_{s_{t-1}s_t}(t) \\ \sum_{s_{t-2}=1}^{N} \eta_{s_{t-2}s_{t-1}}(t-1) \end{cases} = \alpha_{s_{t-1}}(t-1) \cdot \beta_{s_{t-1}}(t-1)$$

$$P(\underline{O}) = \sum_{s_T=1}^{N} \alpha_{s_T}(T) = \beta_{s_0}(0) = \sum_{s_t=1}^{N} \eta_{s_t}(t)$$

$$\gamma_{s_{t-1}s_t}(t) \equiv P(s_{t-1}, s_t \mid \underline{O}) = \frac{P(s_{t-1}, s_t, \underline{O})}{P(\underline{O})} = \frac{\eta_{s_{t-1}s_t}(t)}{P(\underline{O})}$$

$$\gamma_{s_{t-1}}(t-1) \equiv P(s_{t-1} \mid \underline{O}) = \frac{P(s_{t-1}, \underline{O})}{P(\underline{O})} = \frac{\eta_{s_{t-1}}(t-1)}{P(\underline{O})}$$

$$\gamma_{s_t}(t) \equiv P(s_t \mid \underline{O}) = \frac{P(s_t, \underline{O})}{P(\underline{O})} = \frac{\eta_{s_t}(t)}{P(\underline{O})}$$

$$\hat{a}_{s_{t-1}s_t}|_{\underline{O}} \equiv (\text{Estimation of } a_{s_{t-1}s_t}, \text{ given } \underline{O})$$

$$= \frac{<\gamma_{s_{t-1}s_t}(t)>}{<\gamma_{s_{t-1}}(t-1)>} = \frac{\frac{1}{T}\sum_{t=1}^{T} \gamma_{s_{t-1}s_t}(t)}{\frac{1}{T}\sum_{t=1}^{T} \gamma_{s_{t-1}}(t-1)} = \frac{\sum_{t=1}^{T} \gamma_{s_{t-1}s_t}(t)}{\sum_{t=1}^{T} \gamma_{s_{t-1}}(t-1)}$$

$$= \frac{\sum_{t=1}^{T} \eta_{s_{t-1}s_t}(t)}{\sum_{t=1}^{T} \eta_{s_{t-1}}(t-1)}$$

$$\hat{b}_{s_t}(o)|_{\underline{O}} \equiv (\text{Estimation of } b_{s_t}(o), \text{ given } \underline{O})$$

$$= \frac{<\gamma_{s_t}(t) \cdot \delta(o_t - o)>}{<\gamma_{s_t}(t)>} = \frac{\frac{1}{T}\sum_{t=1}^{T} \gamma_{s_t}(t) \cdot \delta(o_t - o)}{\frac{1}{T}\sum_{t=1}^{T} \gamma_{s_t}(t)} = \frac{\sum_{t=1}^{T} \gamma_{s_t}(t) \cdot \delta(o_t - o)}{\sum_{t=1}^{T} \gamma_{s_t}(t)}$$

$$= \frac{\sum_{t=1}^{T} \eta_{s_t}(t) \cdot \delta(o_t - o)}{\sum_{t=1}^{T} \eta_{s_t}(t)}$$

$$\text{where} \begin{cases} < f(t) > \text{ denotes the time average of } f(t) \\ \delta(o_t - o) = \begin{cases} 1, & \text{if } o_t = o \\ 0, & \text{otherwise} \end{cases} \end{cases},$$

It can be shown that

$$P(\underline{O} \mid \hat{\bar{\pi}}, \hat{\underline{\underline{A}}}, \hat{\underline{\underline{B}}}) \geq P(\underline{O} \mid \bar{\pi}, \underline{\underline{A}}, \underline{\underline{B}})$$

Ideally, by several iterations,

$$(\hat{\bar{\pi}}, \hat{\underline{\underline{A}}}, \hat{\underline{\underline{B}}}) \to (\bar{\pi}^*, \underline{\underline{A}}^*, \underline{\underline{B}}^*) = \underset{\forall(\bar{\pi}, \underline{\underline{A}}, \underline{\underline{B}})}{\textbf{Argmax}}\, P(\underline{O} \mid \bar{\pi}, \underline{\underline{A}}, \underline{\underline{B}})$$

Ren-yuan Lyu
@ Chang Gung University

Underflow problem of Kernel computing in HMM

$$\alpha_{s_t}(t) \equiv P(o_1, o_2 ...., o_t, s_t) = \sum_{s_{t-1}=1}^{N} \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t)$$

$$\beta_{s_t}(t) \equiv P(o_{t+1}, o_{t+2}, ..., o_{T-1}, o_T \mid s_t) = \sum_{s_{t+1}=1}^{N} a_{s_t s_{t+1}} \cdot b_{s_{t+1}}(o_{t+1}) \cdot \beta_{s_{t+1}}(t+1)$$

$$\eta_{s_{t-1}s_t}(t) \equiv P(s_{t-1}, s_t, \underline{O}) = \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t) \cdot \beta_{s_t}(t)$$

$$\eta_{s_t}(t) \equiv P(s_t, \underline{O}) = \sum_{s_{t-1}=1}^{N} \eta_{s_{t-1}s_t}(t) = \alpha_{s_t}(t) \cdot \beta_{s_t}(t)$$

$$P(\underline{O}) = \sum_{s_T=1}^{N} \alpha_{s_T}(T) = \beta_{s_0}(0) = \sum_{s_t=1}^{N} \eta_{s_t}(t)$$

when t is large, the values of all the above functions go to 0 (Underflow!)

To avoid the underflow problem, taking *log* for all the above functions.

Ren-yuan Lyu
@ Chang Gung University

2013/10/29

Log Summation

Given x_i be very small positive numbers ($\approx 10^{-10000}$)

$y_i = \log(x_i)$

will behave much better when processed (or stored) in a fixed - precision computer

It is easy when $f(x_1,..x_i..,x_N) = \prod_{i=1}^{N} x_i$ need to be computed,

because

$$\log\left(f(x_1,..x_i..,x_N)\right) = \log\left(\prod_{i=1}^{N} x_i\right) = \sum_{i=1}^{N} \log(x_i) = \sum_{i=1}^{N} y_i$$

However, it is not trivial when $h(x_1,..x_i..,x_N) = \sum_{i=1}^{N} x_i$ need to be computed.

Question:

$$h(x_1,..x_i..,x_N) = \sum_{i=1}^{N} x_i$$

$$\log\left(h(x_1,..x_i..,x_N)\right) = \log\left(\sum_{i=1}^{N} x_i\right) = ?$$

The answer must be in terms of y_i

[Sol]:

$$\log\left(h(x_1,..x_i..,x_N)\right) = \log\left(\sum_{i=1}^{N} x_i\right)$$

$$= \log\left(\sum_{i=1}^{N} e^{\log(x_i)}\right) = \log\left(\sum_{i=1}^{N} e^{\log(x_i)} \frac{e^{\log\left(x_{i^*}\right)}}{e^{\log\left(x_{i^*}\right)}}\right)$$

$$= \log\left(\sum_{i=1}^{N} e^{\log\left(x_{i^*}\right)} \frac{e^{\log(x_i)}}{e^{\log\left(x_{i^*}\right)}}\right) = \log\left(e^{\log\left(x_{i^*}\right)} \cdot \sum_{i=1}^{N} e^{\left(\log(x_i)-\log\left(x_{i^*}\right)\right)}\right)$$

$$= \log\left(e^{\log\left(x_{i^*}\right)}\right) + \log\left(\sum_{i=1}^{N} e^{\left(\log(x_i)-\log\left(x_{i^*}\right)\right)}\right)$$

$$= \log\left(x_{i^*}\right) + \log\left(\sum_{i=1}^{N} e^{\left(\log(x_i)-\log\left(x_{i^*}\right)\right)}\right)$$

$$= y_{i^*} + \log\left(\sum_{i=1}^{N} e^{\left(y_i - y_{i^*}\right)}\right)$$

where $i^* = \arg\max_{i\in[1..N]} \{x_i\} = \arg\max_{i\in[1..N]} \{\log(x_i)\} = \arg\max_{i\in[1..N]} \{y_i\}$

$$\alpha_{s_t}(t) = \sum_{s_{t-1}=1}^{N} \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t)$$

$$\log\left(\alpha_{s_t}(t)\right) = \log\left(\sum_{s_{t-1}=1}^{N} \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t)\right)$$

$$= \log\left(\sum_{s_{t-1}=1}^{N} \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t}\right) + \log\left(b_{s_t}(o_t)\right)$$

$$= \log\left(\alpha_{s^*_{t-1}}(t-1) \cdot a_{s^*_{t-1}s_t}\right) + \log\left(\sum_{s_{t-1}=1}^{N} e^{\left(\log\left(\alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t}\right) - \log\left(\alpha_{s^*_{t-1}}(t-1) \cdot a_{s^*_{t-1}s_t}\right)\right)}\right) + \log\left(b_{s_t}(o_t)\right)$$

$$\begin{aligned} = \quad & \log\left(\alpha_{s^*_{t-1}}(t-1)\right) \\ & + \log\left(a_{s^*_{t-1}s_t}\right) \\ & + \log\left(\sum_{s_{t-1}=1}^{N} e^{\left(\log\left(\alpha_{s_{t-1}}(t-1)\right) + \log\left(a_{s_{t-1}s_t}\right) - \log\left(\alpha_{s^*_{t-1}}(t-1)\right) - \log\left(a_{s^*_{t-1}s_t}\right)\right)}\right) \\ & + \log\left(b_{s_t}(o_t)\right) \end{aligned}$$

$$\text{where } s^*_{t-1} = \underset{s_{t-1} \in [1..N]}{\operatorname{argmax}} \left\{\log\left(\alpha_{s_{t-1}}(t-1)\right) + \log\left(a_{s_{t-1}s_t}\right)\right\}$$

Ren-yuan Lyu
@ Chang Gung University

Log version of Kernel computing in HMM

$$\alpha_{s_t}(t) = \sum_{s_{t-1}=1}^{N} \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t)$$

$$\begin{aligned}
&\log\left(\alpha_{s_t}(t)\right) \\
&= \log\left(\alpha_{s^*_{t-1}}(t-1)\right) \\
&\quad + \log\left(a_{s^*_{t-1}s_t}\right) \\
&\quad + \log\left(\sum_{s_{t-1}=1}^{N} e^{\left(\log\left(\alpha_{s_{t-1}}(t-1)\right) + \log\left(a_{s_{t-1}s_t}\right) - \log\left(\alpha_{s^*_{t-1}}(t-1)\right) - \log\left(a_{s^*_{t-1}s_t}\right)\right)}\right) \\
&\quad + \log\left(b_{s_t}(o_t)\right) \\
&\quad \text{where } s^*_{t-1} = \operatorname*{argmax}_{s_{t-1}\in[1..N]} \left\{\log\left(\alpha_{s_{t-1}}(t-1)\right) + \log\left(a_{s_{t-1}s_t}\right)\right\}
\end{aligned}$$

$$\beta_{s_t}(t) = \sum_{s_{t+1}=1}^{N} a_{s_t s_{t+1}} \cdot b_{s_{t+1}}(o_{t+1}) \cdot \beta_{s_{t+1}}(t+1)$$

$$\log\left(\beta_{s_t}(t)\right) = ?$$

$$\eta_{s_{t-1}s_t}(t) = \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t) \cdot \beta_{s_t}(t)$$

$$\log\left(\eta_{s_{t-1}s_t}(t)\right) = ?$$

$$\eta_{s_t}(t) = \alpha_{s_t}(t) \cdot \beta_{s_t}(t)$$

$$\log\left(\eta_{s_t}(t)\right) = ?$$

$$P(\underline{O}) = \sum_{s_T=1}^{N} \alpha_{s_T}(T) = \beta_{s_0}(0) = \sum_{s_t=1}^{N} \eta_{s_t}(t)$$

$$\log\left(P(\underline{O})\right) = ?$$

$$\gamma_{s_{t-1}s_t}(t) = \frac{\eta_{s_{t-1}s_t}(t)}{P(\underline{O})} = \frac{e^{\log\left(\eta_{s_{t-1}s_t}(t)\right)}}{e^{\log(P(\underline{O}))}} = e^{\left(\log\left(\eta_{s_{t-1}s_t}(t)\right) - \log(P(\underline{O}))\right)}$$

$$\gamma_{s_{t-1}}(t-1) = \frac{\eta_{s_{t-1}}(t-1)}{P(\underline{O})} = e^{\left(\log\left(\eta_{s_{t-1}}(t-1)\right) - \log(P(\underline{O}))\right)}$$

$$\gamma_{s_t}(t) = \frac{\eta_{s_t}(t)}{P(\underline{O})} = e^{\left(\log\left(\eta_{s_t}(t)\right) - \log(P(\underline{O}))\right)}$$

$$\hat{a}_{s_{t-1}s_t}\big|_{\underline{O}} \equiv (\text{Estimation of } a_{s_{t-1}s_t}, \text{ given } \underline{O})$$

$$= \frac{<\gamma_{s_{t-1}s_t}(t)>}{<\gamma_{s_{t-1}}(t-1)>} = \frac{\sum_{t=1}^{T}\gamma_{s_{t-1}s_t}(t)}{\sum_{t=1}^{T}\gamma_{s_{t-1}}(t-1)} ============= \frac{\Gamma_{s_{t-1}s_t}}{\Gamma_{s_{t-1}}}$$

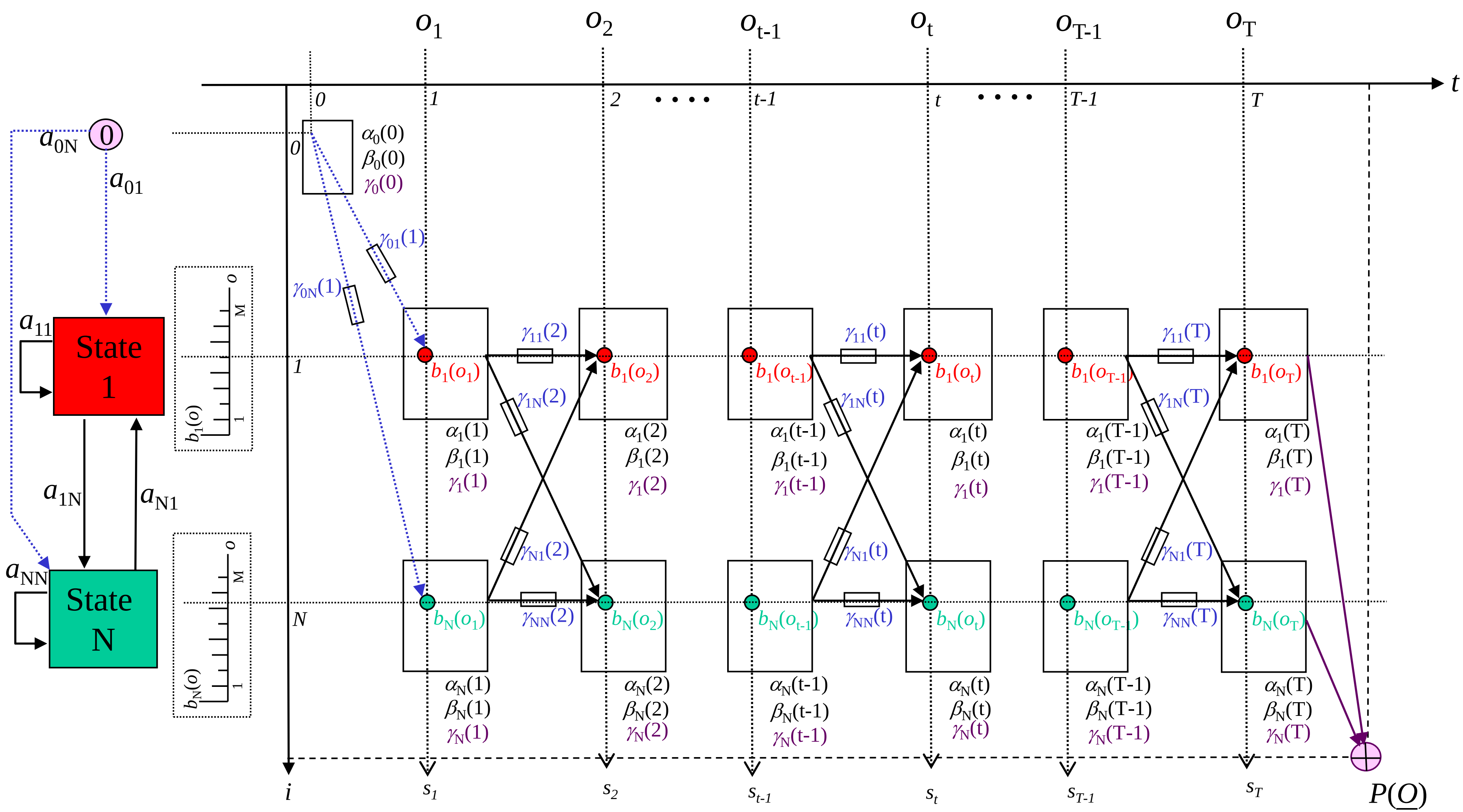
$$\hat{b}_{s_t}(o)\big|_{\underline{O}} \equiv (\text{Estimation of } b_{s_t}(o), \text{ given } \underline{O})$$

$$= \frac{<\gamma_{s_t}(t)\cdot\delta(o_t - o)>}{<\gamma_{s_t}(t)>} = \frac{\sum_{t=1}^{T}\gamma_{s_t}(t)\cdot\delta(o_t - o)}{\sum_{t=1}^{T}\gamma_{s_t}(t)} ===== \frac{\Delta_{s_t o}}{\Gamma_{s_t}}$$

$$\text{where} \quad \delta(o_t - o) = \begin{cases} 1, & \text{if } o_t = o \\ 0, & \text{otherwise} \end{cases}$$

注意: $\gamma_{s_{t-1}s_t}(t),\ \gamma_{s_{t-1}}(t-1), \gamma_{s_t}(t)$ 本身不需取 log，它們不會 Underflow

Ren-yuan Lyu
@ Chang Gung University



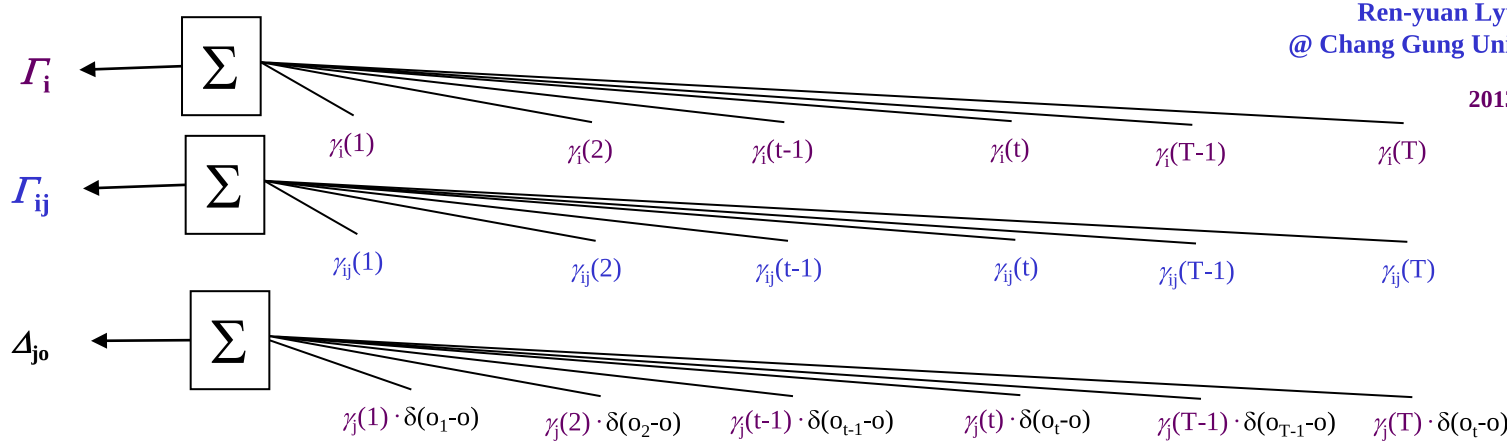
Ren-yuan Lyu
@ Chang Gung University

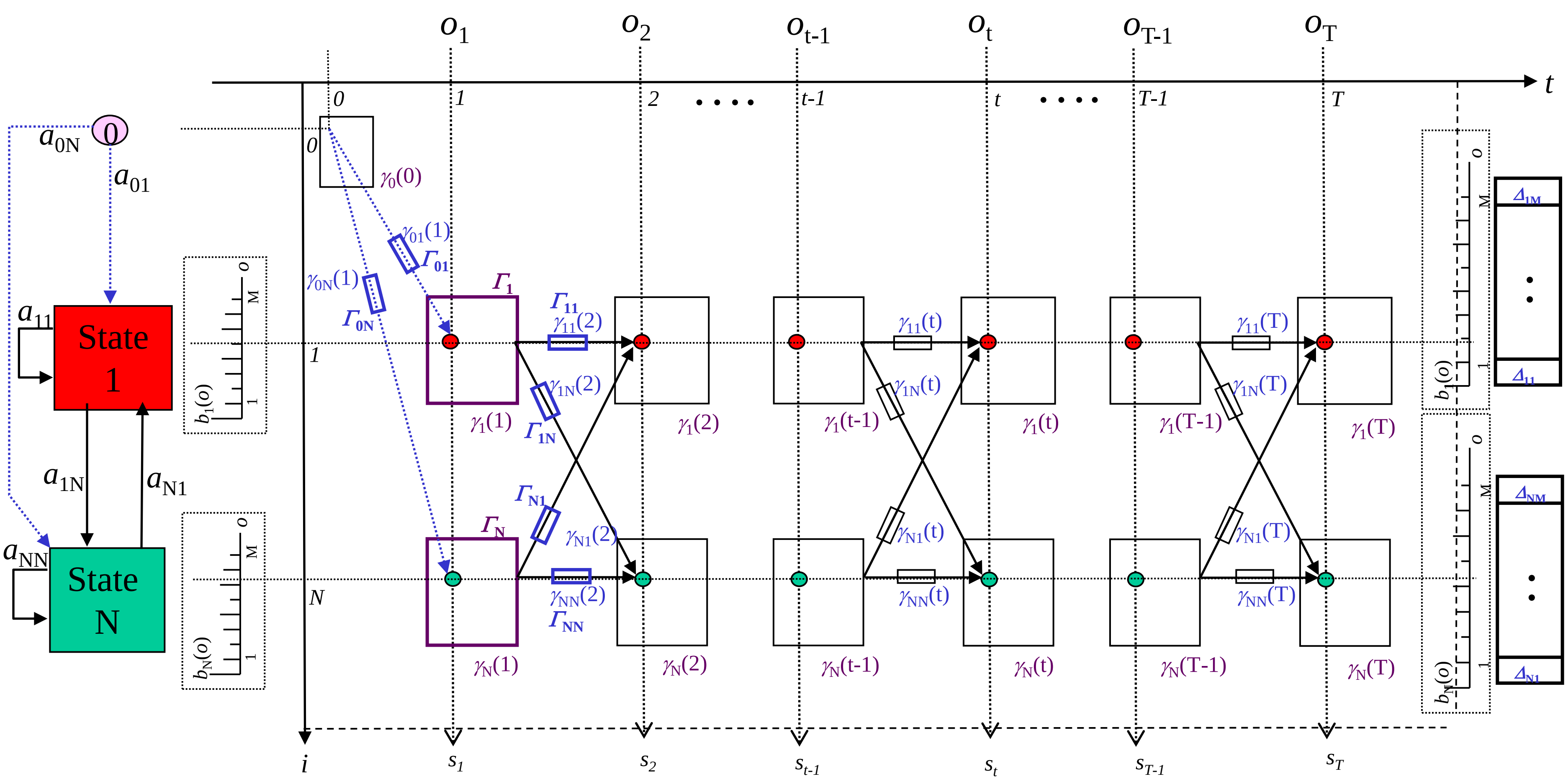
2013/10/29

$$\hat{\pi}_j = \hat{a}_{0j}\big|_{\underline{O}} = \frac{\Gamma_{0j}}{\Gamma_0}$$

$$\hat{a}_{ij}\big|_{\underline{O}} = \frac{\Gamma_{ij}}{\Gamma_i}$$

$$\hat{b}_j(o)\big|_{\underline{O}} = \frac{\Delta_{jo}}{\Gamma_j}$$





Ren-yuan Lyu
@ Chang Gung University

$$\Gamma_{s_t} = \sum_{t=1}^{T} \gamma_{s_t}(t)$$

$$\Gamma_{s_{t-1}s_t} = \sum_{t=1}^{T} \gamma_{s_{t-1}s_t}(t)$$

$$\Delta_{s_t o} = \sum_{t=1}^{T} \gamma_{s_t}(t) \cdot \delta(o_t - o)$$

- -

$$\hat{a}_{s_{t-1}s_t}\big|_{\underline{O}} = \frac{\Gamma_{s_{t-1}s_t}}{\Gamma_{s_{t-1}}}$$

$$\hat{b}_{s_t}(o)\big|_{\underline{O}} = \frac{\Delta_{s_t o}}{\Gamma_{s_t}}$$

$$s_0 \equiv 0$$

$$s_{t-1} = i \in [1..N]$$

$$s_t = j \in [1..N]$$

$$o \in [1..M]$$

$$\Gamma_i = \sum_{t=0}^{T-1} \gamma_i(t)$$

$$\Gamma_{ij} = \sum_{t=1}^{T} \gamma_{ij}(t)$$

$$\Gamma_0 = \gamma_0(0)$$

$$\Gamma_{0j} = \gamma_{0j}(1)$$

$$\Gamma_j = \sum_{t=1}^{T} \gamma_j(t)$$

$$\Delta_{jo} = \sum_{t=1}^{T} \gamma_j(t) \cdot \delta(o_t - o)$$

- -

$$\hat{\pi}_j = \hat{a}_{0j}\big|_{\underline{O}} = \frac{\Gamma_{0j}}{\Gamma_0}$$

$$\hat{a}_{ij}\big|_{\underline{O}} = \frac{\Gamma_{ij}}{\Gamma_i}$$

$$\hat{b}_j(o)\big|_{\underline{O}} = \frac{\Delta_{jo}}{\Gamma_j}$$

Some interpretations about α,β,η,γ

$$\alpha_{s_t}(t) \equiv P(o_1, o_2, \ldots, o_t, s_t) = \sum_{s_{t-1}=1}^{N} \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t)$$

$$\beta_{s_t}(t) \equiv P(o_{t+1}, o_{t+2}, \ldots, o_{T-1}, o_T \mid s_t) = \sum_{s_{t+1}=1}^{N} a_{s_t s_{t+1}} \cdot b_{s_{t+1}}(o_{t+1}) \cdot \beta_{s_{t+1}}(t+1)$$

$$\eta_{s_{t-1}s_t}(t) \equiv P(s_{t-1}, s_t, \underline{O}) = \alpha_{s_{t-1}}(t-1) \cdot a_{s_{t-1}s_t} \cdot b_{s_t}(o_t) \cdot \beta_{s_t}(t)$$

$$\eta_{s_t}(t) \equiv P(s_t, \underline{O}) = \sum_{s_{t-1}=1}^{N} \eta_{s_{t-1}s_t}(t) = \alpha_{s_t}(t) \cdot \beta_{s_t}(t)$$

$$P(\underline{O}) = \sum_{s_T=1}^{N} \alpha_{s_T}(T) = \beta_{s_0}(0) = \sum_{s_t=1}^{N} \eta_{s_t}(t)$$

$$\gamma_{s_{t-1}s_t}(t) \equiv P(s_{t-1}, s_t \mid \underline{O}) = \eta_{s_{t-1}s_t}(t) \big/ P(\underline{O})$$

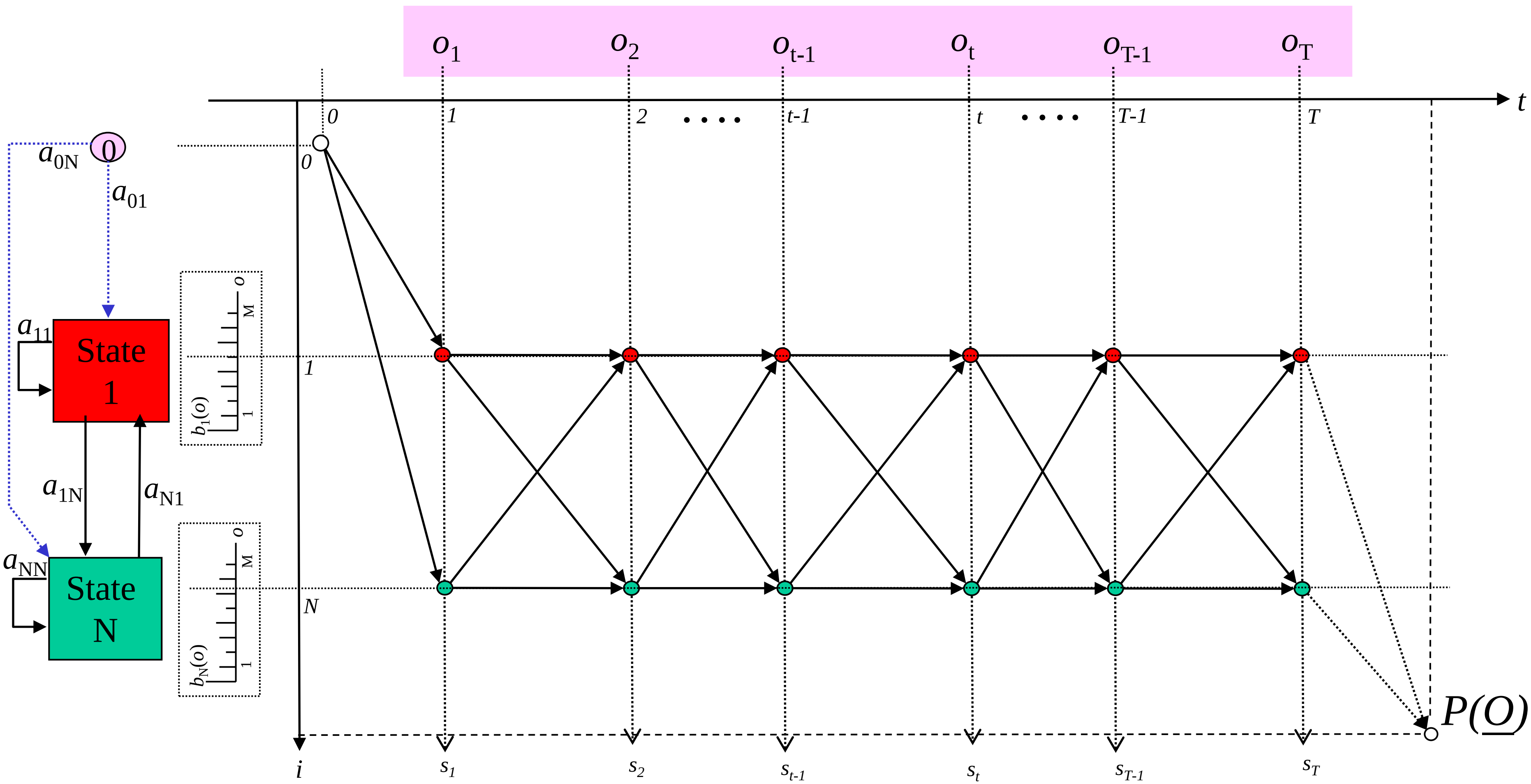
$$\gamma_{s_t}(t) \equiv P(s_t \mid \underline{O}) = \eta_{s_t}(t) \big/ P(\underline{O})$$

Ren-yuan Lyu
@ Chang Gung University

$P(\underline{O})$

代表所有

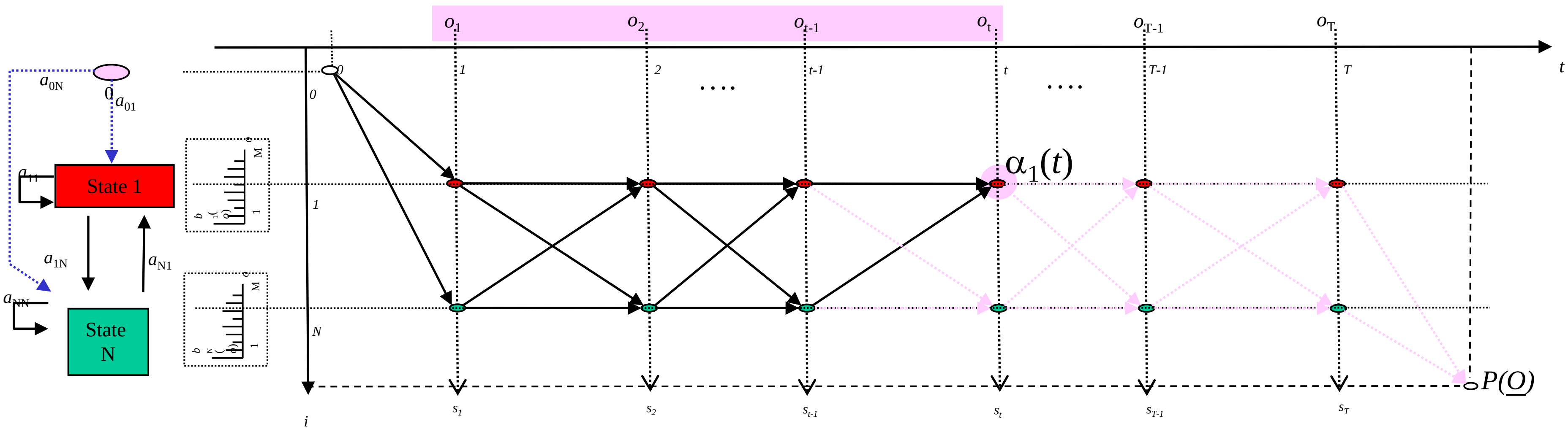
從(i=0，t=0)到(i=N+1，t=T+1)所有的路徑，
如圖所示總共有N^T條

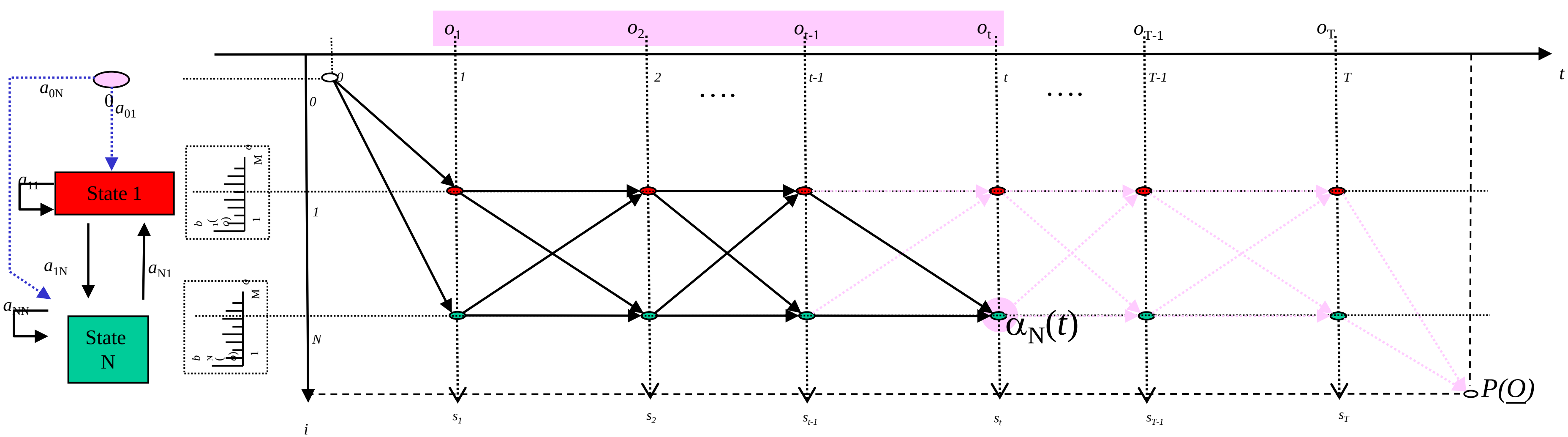


Ren-yuan Lyu
@ Chang Gung University

代表所有
從(i=0,t=0)到(i, t)所有的路徑，
如圖所示總共有N^{t-1}條

$\alpha_i(t)$



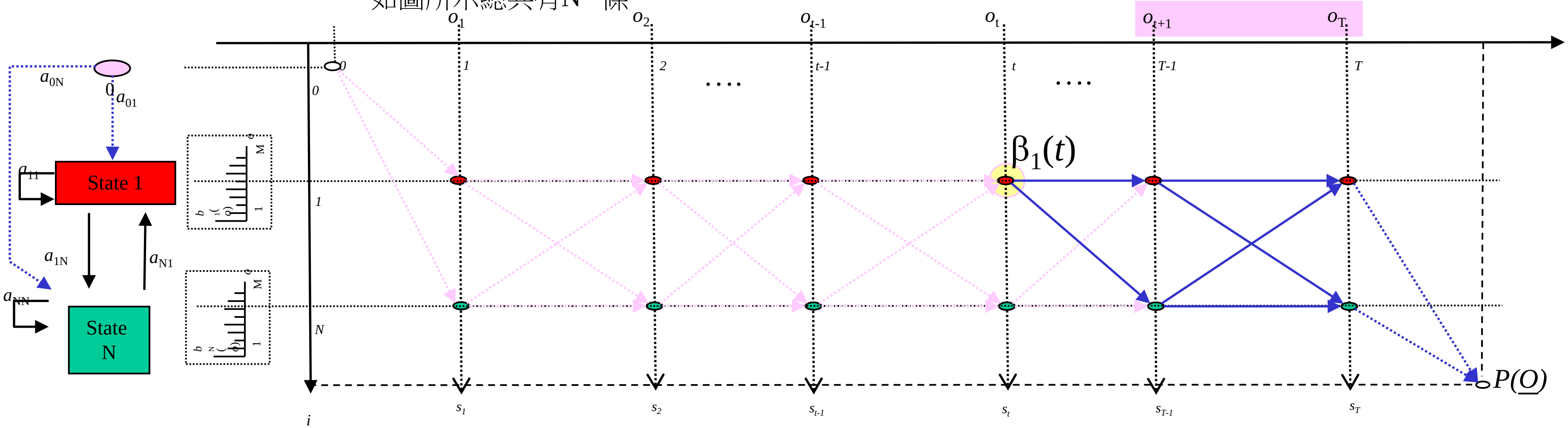


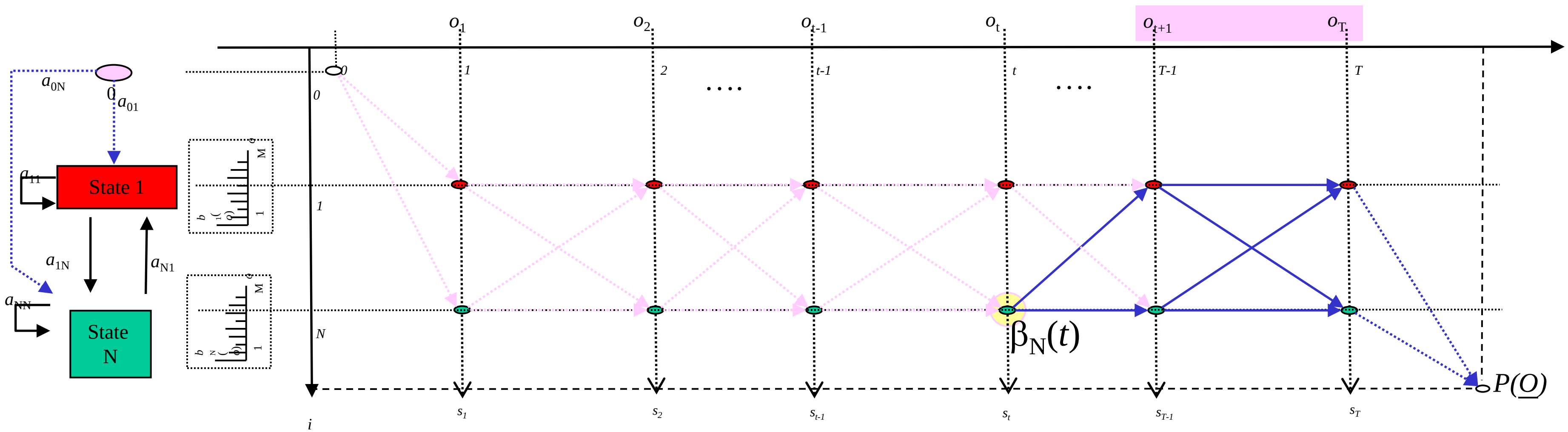
Ren-yuan Lyu
@ Chang Gung University

$\beta_i(t)$

代表所有

從(i,t)到(i=N+1,t=T+1)所有的路徑，
如圖所示總共有N^{T-t}條



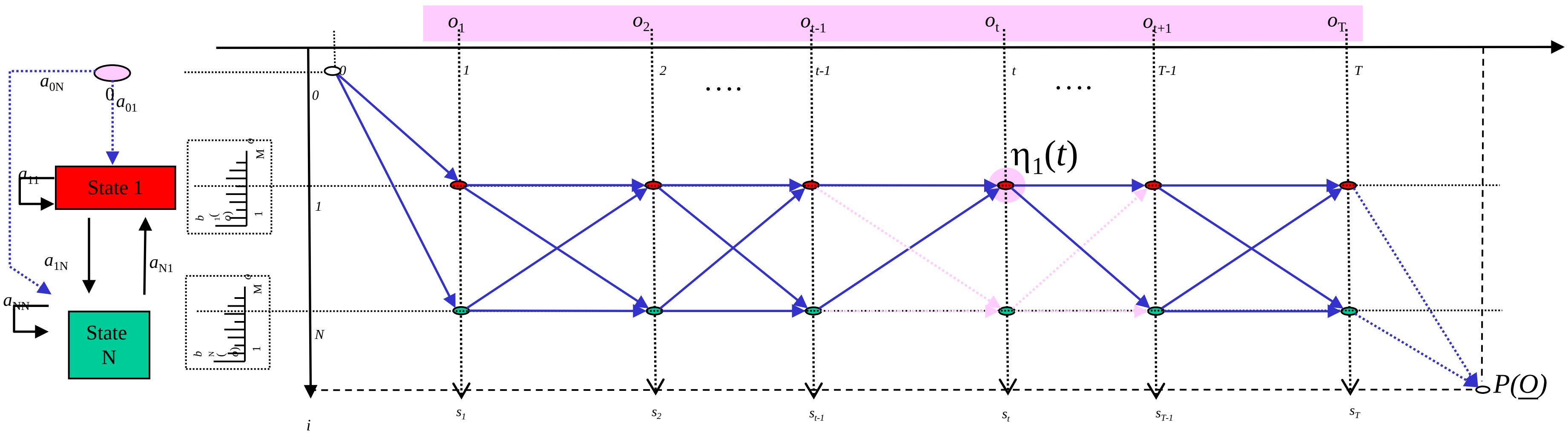


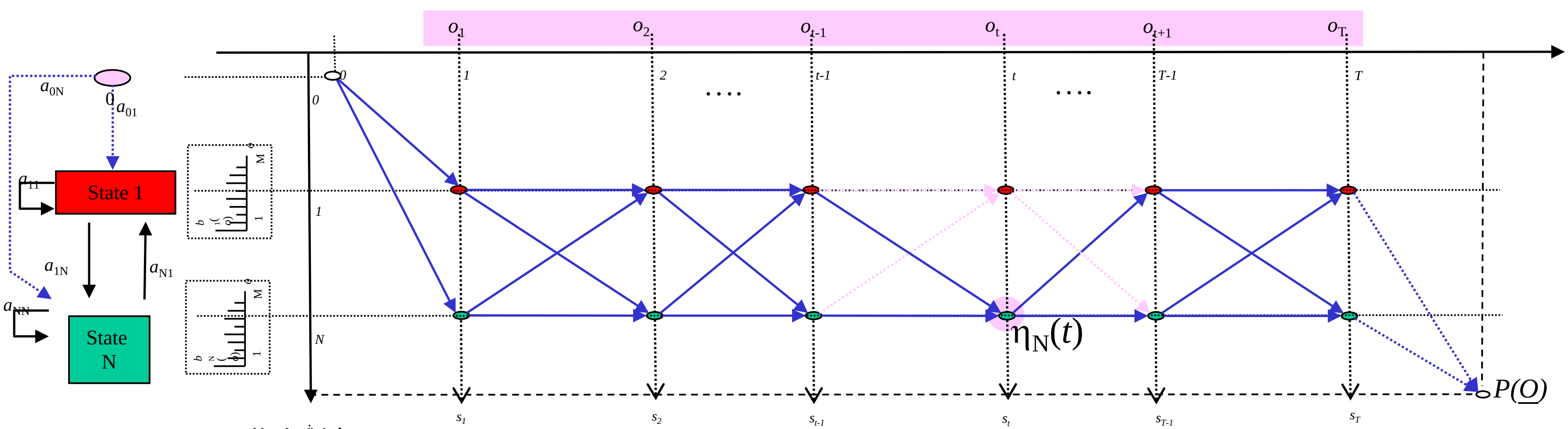
Ren-yuan Lyu
@ Chang Gung University

2013/10/29

$\gamma_i(t)$

$\eta_i(t)$





代表所有
從(i=0,t=0)到(I,t)

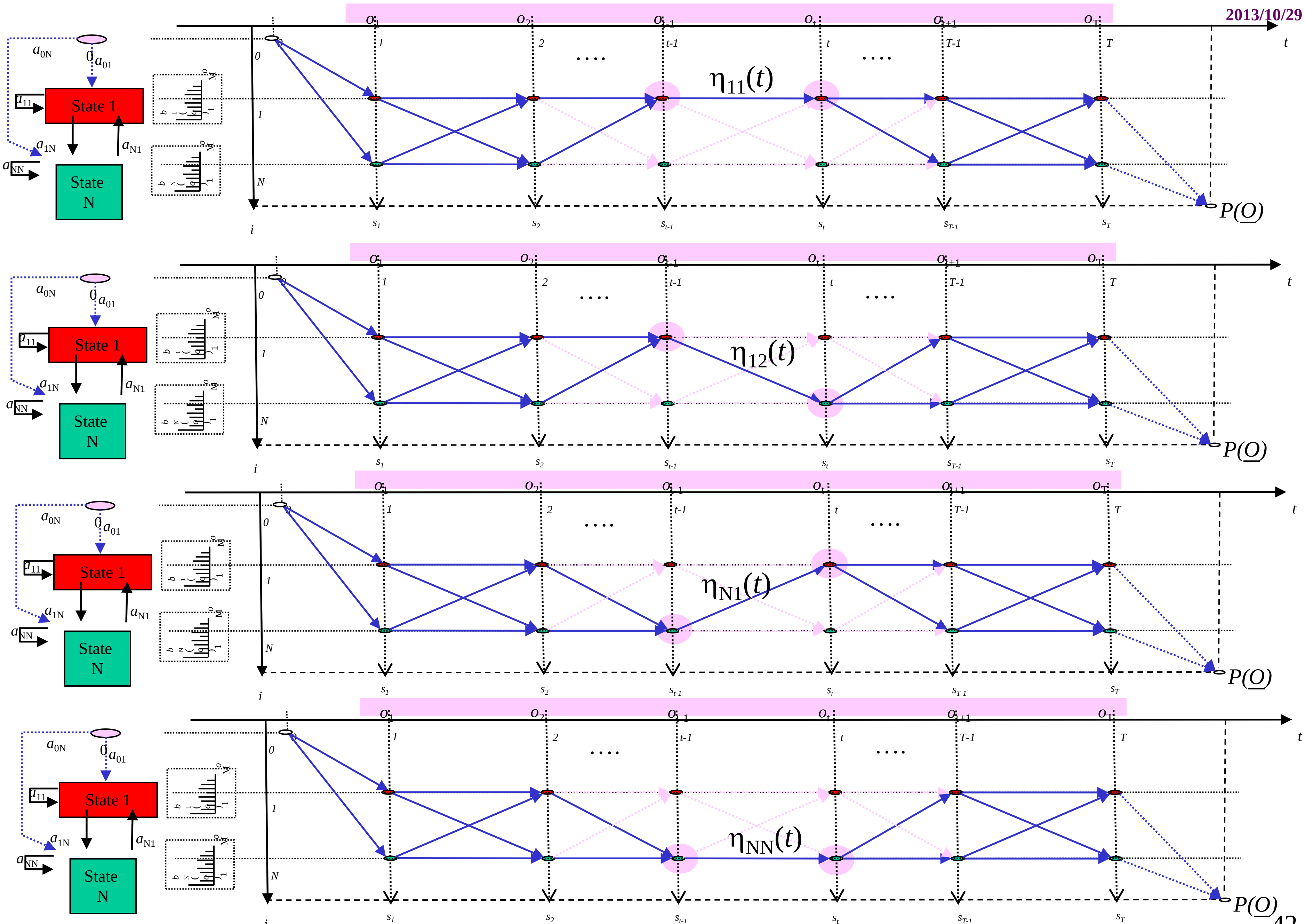
再從(i,t)到(i=N+1, t=T+1)的
所有的路徑，如圖所示總共有N^{T-1}條

這些路徑的共同特色為： 它們都經過(t,i)

$\eta_{ij}(t), \quad \gamma_{ij}(t)$

Ren-yuan Lyu
@ Chang Gung University

2013/10/29



Ren-yuan Lyu
@ Chang Gung University

Training Formula & Interpretations

$$\Gamma_i = \sum_{t=0}^{T-1} \gamma_i(t)$$ 走完全程後，(state=i)會出現的次數

$$\Gamma_{ij} = \sum_{t=1}^{T} \gamma_{ij}(t)$$ 走完全程後，(state=i)且(next state=j)會出現的次數

$$\Gamma_0 = \gamma_0(0)$$ 走完全程後，一開始(t=0時）(state=0)會出現的次數

$$\Gamma_{0j} = \gamma_{0j}(1)$$ 走完全程後，一開始(t=0時）(state=0)且(next state=j) 會出現的次數

$$\Gamma_j = \sum_{t=1}^{T} \gamma_j(t)$$ 走完全程後，(state=j)會出現的次數

$$\Delta_{jo} = \sum_{t=1}^{T} \gamma_j(t) \cdot \delta(o_t - o)$$ 走完全程後，(state=j)且(observation=o) 會出現的次數

$$\hat{\pi}_j = \hat{a}_{0j} |_{\underline{O}} = \frac{\Gamma_{0j}}{\Gamma_0}$$

$$\hat{a}_{ij} |_{\underline{O}} = \frac{\Gamma_{ij}}{\Gamma_i}$$

$$\hat{b}_j(o) |_{\underline{O}} = \frac{\Delta_{jo}}{\Gamma_j}$$

The Decoding Problem of HMM

- Given a HMM, with parameters {π, **A**, **B**}, and an observation sequence

$$o_1, o_2, o_3, \ldots, o_t, \ldots, o_T$$

What is the "best" or "most likely" state sequence

$$s_1^*, s_2^*, s_3^*, \ldots, s_t^*, \ldots, s_T^*$$

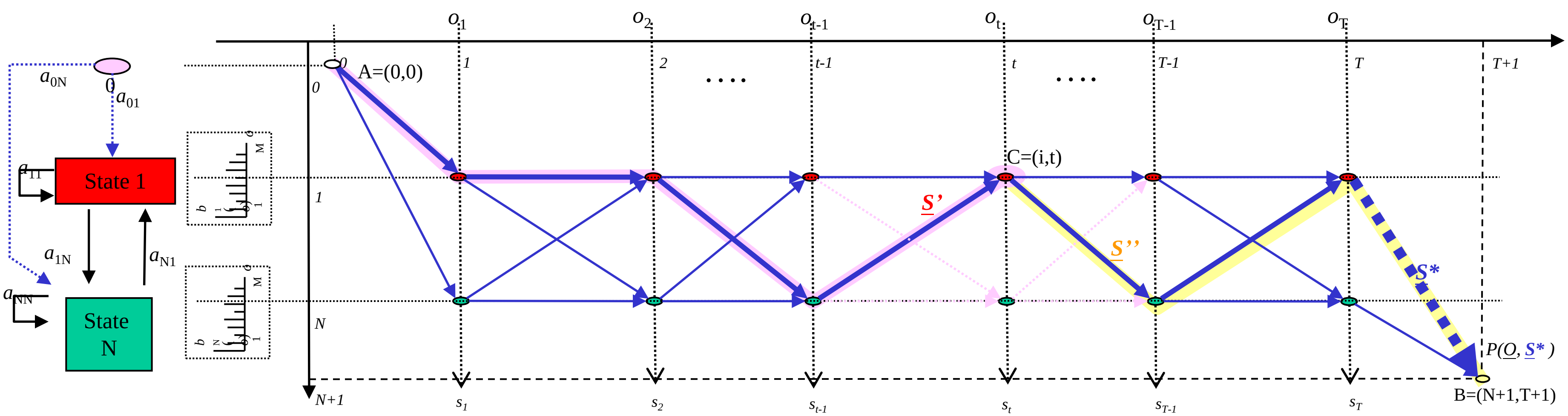
to help the HMMachine generate such an observation sequence? This problem can be reformulate as follows:

$$(s_1^*, s_2^*, s_3^*, \ldots, s_t^*, \ldots, s_T^*) = \underset{\forall (s_1, s_2, s_3, \ldots, s_t, \ldots, s_T)}{\boldsymbol{Argmax}} P(s_1, s_2, s_3, \ldots, s_t, \ldots, s_T \mid o_1, o_2, o_3, \ldots, o_t, \ldots, o_T) = ?$$

Ren-yuan Lyu
@ Chang Gung University
2013/10/29

Optimal principle: (The Dynamic Programming Principle)

If path ***S****** is the "***optimal***" **path** of all paths from A=(0,0) to B=(N+1,T+1), which passes C=(i, t),
Then the ***partial*** path ***S'*** (= ***S****[0..t]) of ***S**** from A=(0,0) to C=(i, t)
must also be the "***optimal***" path of all paths from A=(0,0) to C=(i, t),
And the ***partial*** path ***S''*** (= ***S****[t..T+1]) of ***S**** from C=(i,t) to B=(N+1, T+1)
must also be the "***optimal***" path of all paths from C=(i,t) to B=(N+1, T+1) .
e.g., in the following figure,
S*= (0,0)--(1,1)--(1,2)--(N,t-1)--(1,t)--(N,T-1)--(1,T)--(N+1)--T+1)
S' = (0,0)--(1,1)--(1,2)--(N,t-1)--(1,t)
S''= (1,t)--(N,T-1)--(1,T)--(N+1)--T+1)



If you want to find the optimal path from t=0 to t=T+1,
You should find the optimal partial path from t=0 to t for each possible i in {1,…N}
Because at time t, the optimal path may pass one of points in {(1,t),…,(N,t)}

i

Ren-yuan Lyu
@ Chang Gung University

$$t \in [1..T]$$
$$i \in \Omega_S = \{1..N\}$$

$$\begin{aligned}
\underline{S}'_i(t) &\equiv \text{the optimal partial path from } (i=0, t=0) \text{ to } (i,t) \\
&= \text{"}(0,0) \sim (s_1^*,1) \sim (s_2^*,2) \sim (s_{t-1}^*, t-1) \sim (i,t)\text{"} \\
&\equiv \text{the optimal partial state sequence from beginning to time } (t\text{-}1) \text{ and } (s_t = i) \\
&= [(s_0 \equiv 0), s_1^*, s_2^*, \ldots, s_{t-1}^*, (s_t = i)] \\
&= \underset{s_1, s_2, \ldots, s_{t-1}}{ArgMax}\, P\big(s_1, s_2, \ldots, s_{t-1}, (s_t = i), o_1, o_2, \ldots, o_t\big)
\end{aligned}$$

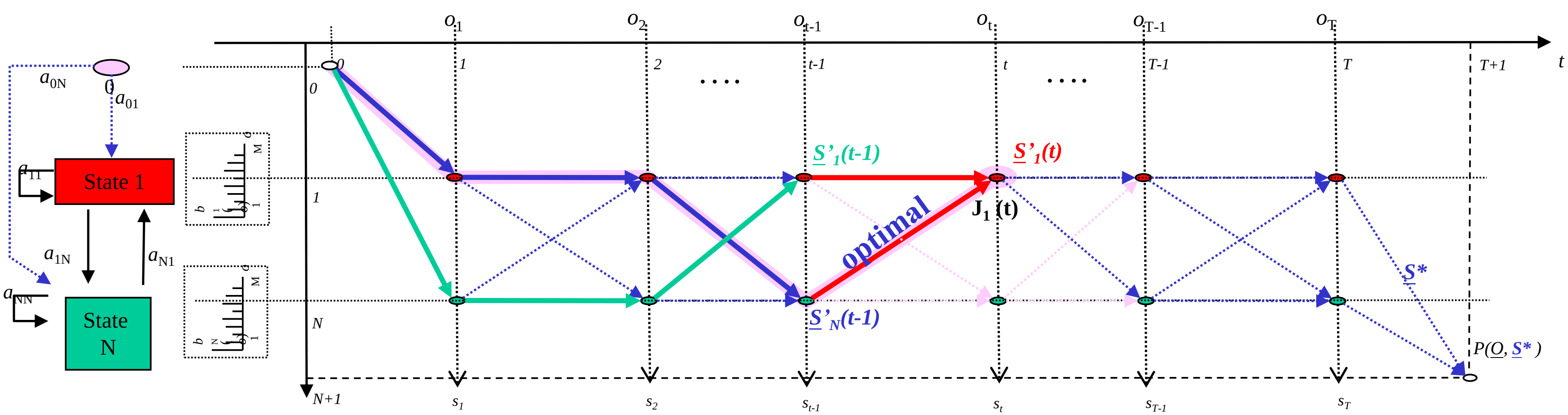
$s_\tau^* \equiv$ the optimal state at time τ

It can be shown by the optimal principle that

$$\underline{S}'_i(t) = \underset{j \in \Omega_S = \{1..N\}}{Opt} \left\{\underline{S}'_j(t-1) \sim (i,t)\right\}$$

And let's record the optimal argument as follows.

$$\begin{aligned}
J_i(t) &\equiv \text{the optimal pre-state to (i, t)} \\
&= \underset{j \in \Omega_S = \{1..N\}}{ArgOpt} \left\{\underline{S}'_j(t-1) \sim (i,t)\right\}
\end{aligned}$$



$$\underline{S}'_i(1) = (0,0) \sim (i,1) \qquad i \in \Omega_S = \{1..N\}$$

$$J_i(1) = 0$$

$$\underline{S}'_i(2) = \underset{j \in \Omega_S = \{1..N\}}{Opt} \left\{ \underline{S}'_j(1) \sim (i,2) \right\}$$

$$J_i(2) = \underset{j \in \Omega_S = \{1..N\}}{ArgOpt} \left\{ \underline{S}'_j(1) \sim (i,2) \right\}$$

$$\underline{S}'_i(t) = \underset{j \in \Omega_S = \{1..N\}}{Opt} \left\{ \underline{S}'_j(t-1) \sim (i,t) \right\}$$

$$J_i(t) = \underset{j \in \Omega_S = \{1..N\}}{ArgOpt} \left\{ \underline{S}'_j(t-1) \sim (i,t) \right\}$$

$$\underline{S}'_i(T) = \underset{j \in \Omega_S = \{1..N\}}{Opt} \left\{ \underline{S}'_j(T-1) \sim (i,T) \right\}$$

$$J_i(T) = \underset{j \in \Omega_S = \{1..N\}}{ArgOpt} \left\{ \underline{S}'_j(T-1) \sim (i,T) \right\}$$

$$\underline{S}'_{N+1}(T+1) = \underset{j \in \Omega_S = \{1..N\}}{optimal} \left\{ \underline{S}'_j(T) \sim (N+1, T+1) \right\}$$

$$J_{N+1}(T+1) = \underset{j \in \Omega_S = \{1..N\}}{ArgOpt} \left\{ \underline{S}'_j(T) \sim (N+1, T+1) \right\}$$

$$s^*(t) \equiv \text{the optimal state at time } t$$

$$s^*(T+1) \equiv N+1$$

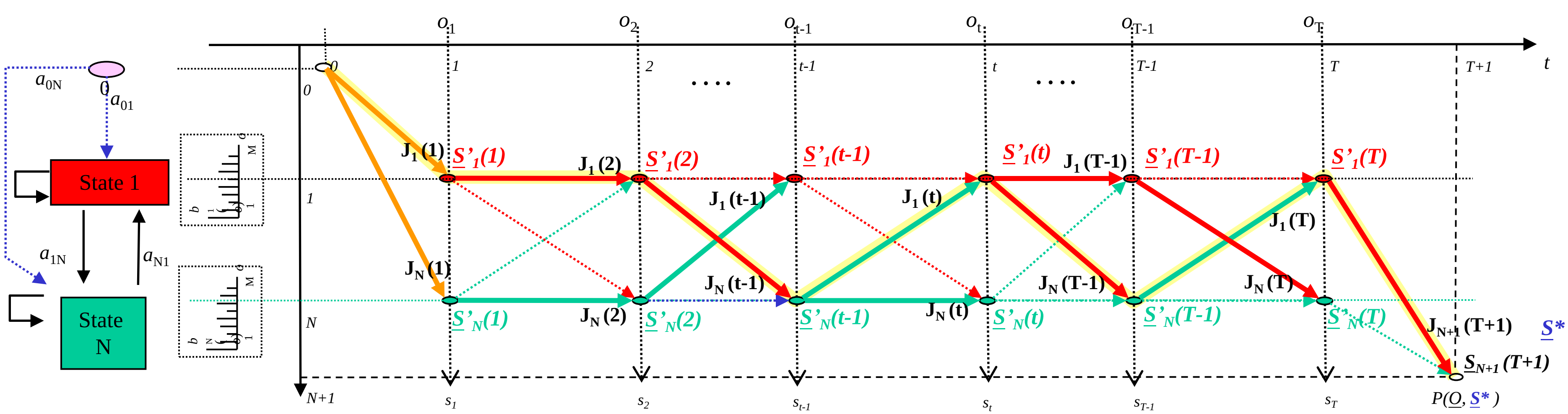
$$s^*(T) = J_{N+1}(T+1) = J_{s^*(T+1)}(T+1)$$

$$s^*(t) = J_{s^*(t+1)}(t+1)$$

$$s^*(1) = J_{s^*(2)}(2)$$

$$s^*(0) = J_{s^*(1)}(1) \equiv 0$$

$$\underline{S}^* = (s_0^* \equiv 0), s^*(1), s^*(2), \ldots, s^*(t), \ldots, s^*(T), (s^*(T+1) \equiv N+1)$$



Ren-yuan Lyu
@ Chang Gung University

An example application

GGGRFFGRFFRRGGGRRGGRGRGRGFRRGFGFRRGGGF

G: Green, down
R: Red, up
F: Flat, level

The observation sequence

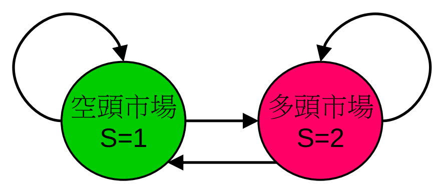
O(t) GGGRFFGRFFRRGGGRRGGRGRGRGFRRGFGFRRGGGF

A ***possible*** state sequence

S(t)

空頭市場 S=1 | 多頭市場 S=2 | 空頭市場 S=1 | 多頭市場 S=2 | 空頭市場 S=1 | 多頭市場 S=2 | 空頭市場 S=1

	空	多	
空	22	**3**	25
多	3	9	12
			37



	空	多	
G	15	**2**	
R	6	7	
F	5	3	
	26	12	38

Problems

(1) Provide an initial estimate of $\{\pi,\mathbf{A},\underline{\mathbf{B}}\}$

(2) $P(\underline{O})$ =?

(3) The optimal sequence $\underline{S}^*$=?

(4) Re-estimate $\{\pi,\mathbf{A},\underline{\mathbf{B}}\}$ as $\{\pi',\mathbf{A}',\underline{\mathbf{B}'}\}$ such that $P(\underline{O}|\ \pi',\mathbf{A}',\underline{\mathbf{B}'}) > P(\underline{O}|\ \pi,\mathbf{A},\underline{\mathbf{B}})$