Hidden Markov Models (HMM) continued

ECE/CS 498 DS U/G

Lecture 15

Ravi K. Iyer

Dept. of Electrical and Computer Engineering

University of Illinois at Urbana Champaign

Announcements

- MP2 Checkpoint 3 due on Wednesday, Mar 27
- MP3 will be released on Friday, Mar 29
- ICA 4 on HMMs today
- Graduate project details released (check Piazza, course website)
    - Two proposal ideas due on Friday, Mar 29
    - Encouraged to start early and share ideas before the deadline

Markov Models vs HMM

A **Markov Model** can be specified by the following components.

Component	Explanation
$x = \{1, 2, \dots N\}; x_t \in x$	A set of N **states** that can be observed directly
$A = \begin{bmatrix} a_{11} & \cdots & \cdots & \cdots & a_{N1} \\ \vdots & \ddots & \cdots & \cdots & \vdots \\ a_{1j} & \cdots & a_{ij} & \cdots & a_{Nj} \\ \vdots & \cdots & \cdots & \ddots & \cdots \\ a_{1N} & \cdots & \cdots & \cdots & a_{NN} \end{bmatrix}$	A **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, $s.t.$ $\sum_{j=1}^{N} a_{ij} = 1 \; \forall i$
$\pi = \pi_1, \pi_2, \dots \pi_N$	An **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{N} \pi_i = 1$

A **Markov Model** embodies the Markov Assumption:

$$P(x_{t+1} | x_0, \dots x_t) = P(x_{t+1} | x_t)$$

Markov Models vs HMM

A **Hidden Markov Model (HMM)** can be specified by the following components.

Component	Explanation
$S = \{\sigma_1, \sigma_2, \dots \sigma_n\}; S_t \in S$	A set of N **states** that are hidden and cannot be directly observed
$A = \begin{bmatrix} a_{11} & \cdots & \cdots & \cdots & a_{N1} \\ \vdots & \ddots & \dots & \dots & \vdots \\ a_{1j} & \cdots & a_{ij} & \cdots & a_{Nj} \\ \vdots & \cdots & \dots & \ddots & \dots \\ a_{1N} & \cdots & \dots & \dots & a_{NN} \end{bmatrix}$	A **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, $s.t.$ $\sum_{j=1}^{N} a_{ij} = 1 \ \forall i$
$E = \{\epsilon_1, \epsilon_2, \dots \epsilon_M\}; E_t \in E$	A set of **observable events**
$O = E_1, E_2, \dots E_T$	A sequence of T observations
$B = \begin{bmatrix} b_{11} & \cdots & \cdots & \cdots & b_{M1} \\ \vdots & \ddots & \dots & \dots & \vdots \\ b_{1j} & \cdots & b_{ij} & \cdots & b_{Mj} \\ \vdots & \cdots & \dots & \ddots & \dots \\ b_{1N} & \cdots & \dots & \dots & b_{MN} \end{bmatrix}$	An **observation matrix** B. Each b_{ij} is referred to as an emission probability or observation likelihood. $i.e$ $b_{ij} = P(E = \epsilon_j \| S = \sigma_i)$
$\pi = \pi_1, \pi_2, \dots \pi_N$	An **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{N} \pi_i = 1$

A HMM embodies the **Markov Assumption**:

$$P(S_{t+1} | S_0, \dots S_t) = P(S_{t+1} | S_t)$$

A HMM also follows **Output Independence**:

$$P(E_t | S_0, \dots, S_t, \dots S_T, E_1, \dots, E_t, \dots E_T) = P(E_t | S_t)$$

Forwards Algorithm

1. Input: (A, B, π) and observed sequence $E_1, \dots, E_n$
2. $[\alpha_1, Z_1]$ = normalize(b_1 π)
3. **for** $t = 2{:}n$ **do**
   $[\alpha_t, Z_t]$ = normalize(b_t $(A^T \alpha_{t-1})$)
4. return $\alpha_1, \dots, \alpha_n$ and $\log\big(P(E_1, \dots, E_n)\big) = \sum_t \log(Z_t)$

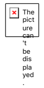
$$\alpha_t(j) = \frac{1}{Z_t} P(E_t | S_t = \sigma_j) \sum_{i=1}^{N} P(S_t = \sigma_j \mid S_{t-1} = \sigma_i)\, \alpha_{t-1}(i)$$

$$Z_t = \sum_{j=1}^{N} \alpha_t(j)$$

5. Subroutine: [v, Z] = normalize(u): $Z = \sum_j u_j$; $v_j = u_j / Z$;

NOTE: represents elementwise product (Hadamard product)

Backwards Algorithm

1. Input: (A, B, π) and observed sequence $E_1, \dots, E_n$
2. *2.* $\beta_n = 1$; // initialize $\beta_n(j)$ to 1 for all states σ_j
3. **for** $t = n - 1 : 1$ **do**
   β_{t-1} = $A(b_t$  $\beta_t)$
4. return $\beta_1, \dots, \beta_n$

Inference – using Forwards-Backwards expressions

$$P(S_t|E_1, E_2, \dots, E_n) = \frac{P(E_{t+1}, \dots, E_n\ |S_t)\ P(S_t|E_1, \dots, E_t)}{P(E_{t+1}, \dots, E_n|E_1, \dots, E_t)}$$

For $S_t = \sigma_j$ and $\gamma_t(j) = P\left(S_t = \sigma_j \middle| E_1, E_2, \dots, E_n\right)$, the above equation is:

$$P\left(S_t = \sigma_j \middle| E_1, E_2, \dots, E_n\right) = \frac{P(E_{t+1}, \dots, E_n\ |S_t = \sigma_j)\ P(S_t = \sigma_j|E_1, \dots, E_t)}{P(E_{t+1}, \dots, E_n|E_1, \dots, E_t)}$$

$$\gamma_t(j) = \frac{\beta_t(j)\alpha_t(j)}{P(E_{t+1}, \dots, E_n|E_1, \dots, E_t)} = \frac{\beta_t(j)\alpha_t(j)}{\sum_{i=1}^{N} \beta_t(j)\alpha_t(j)}$$

Theorem of total probability

$$\boxed{\gamma_t(j) \propto \beta_t(j)\alpha_t(j)}$$

Inference: Most likely state

- Forwards-backwards algorithm gives $P(S_t = \sigma_j | E_1, \dots, E_n)$ for all j
- Find the individually most likely state at time t given all observations

$$S_t^* = \underset{j \in \{1,..,N\}}{\operatorname{argmax}} \gamma_t(j)$$

HMM Security Example

- Suppose you are a security expert monitoring the NCSA system
- By monitoring the system events, you want to say whether the system is safe or not
  - System’s safety is a hidden state
  - Events are observed
  - Events are related to the safety of the system
- Is the system safe?
  - **HMM** to the rescue!

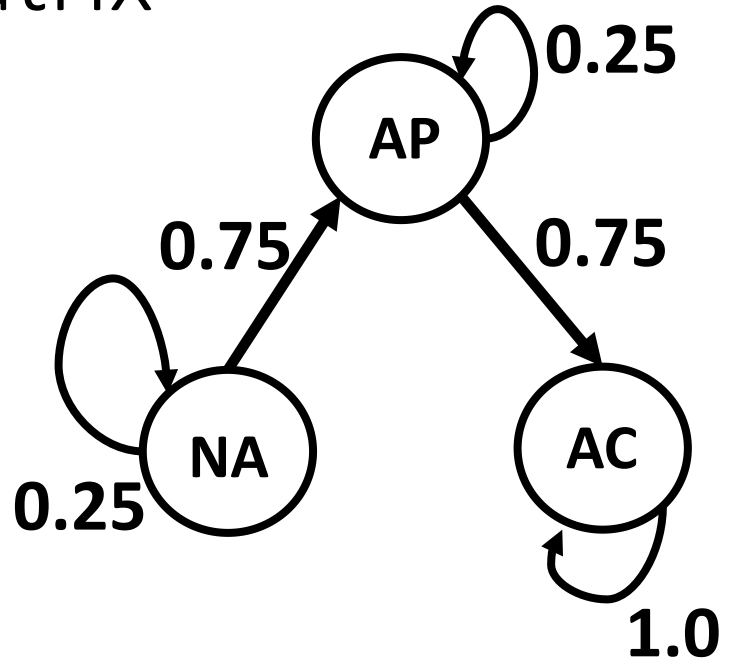
Security Example: Transition Matrix

Transition matrix (A)

The system has three distinct security states –

(a) No Attack **(NA)**,
(b) Attack in Progress **(AP),** and
(c) Attack Complete **(AC).**

- Every hour, the system is being attacked by attackers coordinating together around the world and trying to compromise the system.
- The system states always transition from **NA to AP** and **AP to AC.**
- An attacker is successful in changing the state of the system with probability of 0.75 and fails with a probability of 0.25.
- If the attack fails, the system stays in its current state.
- If the system state reaches **AC** the attack is complete, and the system stays in that state.



The picture can't be displayed.

	NA	**AP**	**AC**
NA			
AP			
AC			

Transition Probability Matrix

Security Example: Emission matrix and initial distribution

Observation matrix (B)

- Your monitoring system reports two types of events
  - Port Scan **(PS)**
  - Software Installation **(SI)**
- Monitors are always accurate and works. Attackers cannot compromise the monitors. Every hour, we get information from the monitors if the attackers are trying to do **PS or SI.**

Initial distribution (π)

- We have no idea about the initial state of the system.

$$\mathbf{B} = \begin{array}{c} \\ \textbf{NA} \\ \textbf{AP} \\ \textbf{AC} \end{array} \begin{array}{c} \begin{array}{cc} \textbf{PS} & \textbf{SI} \end{array} \\ \begin{pmatrix} 0.7 & 0.3 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{pmatrix} \end{array}$$

Observation Matrix

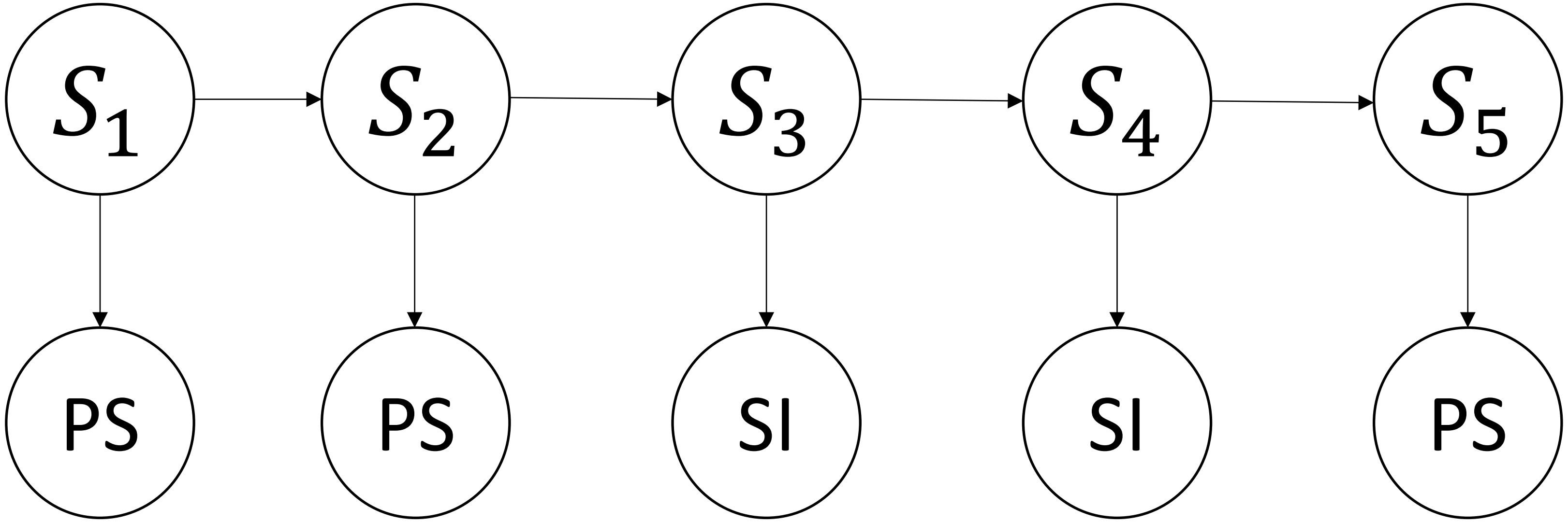
The picture can't be displayed.

NA
AP
AC

Initial state distribution/prior

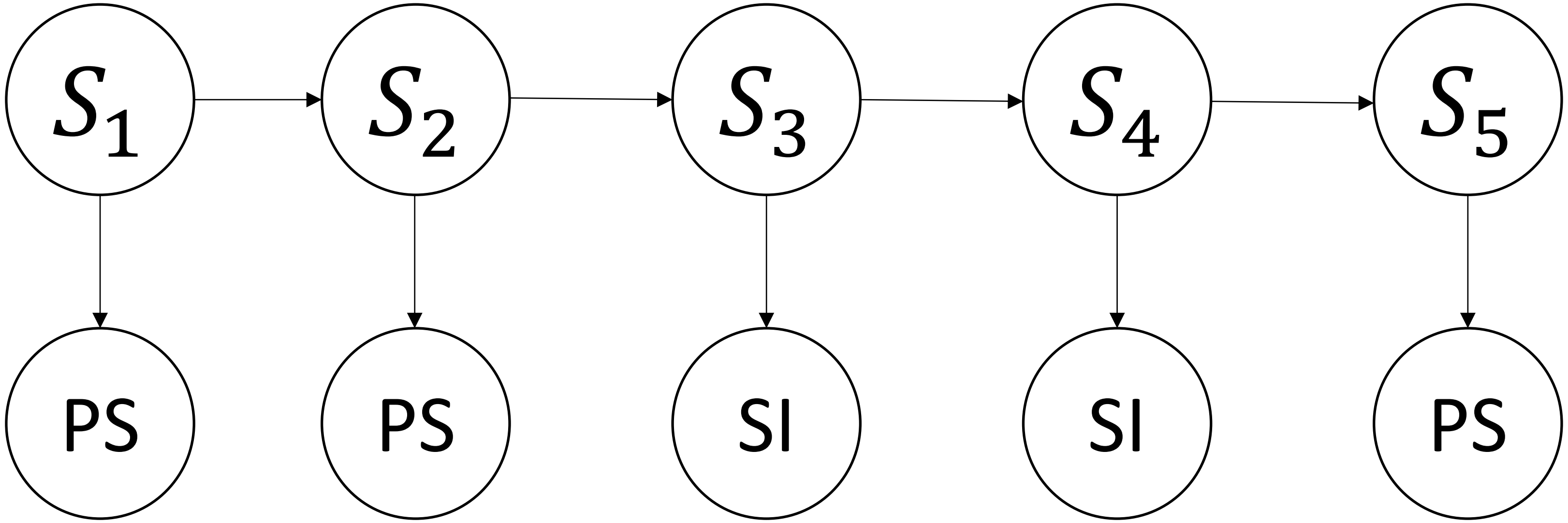
Security Example – Observed Sequence

Find $S_1, \ldots, S_5$ given the observed sequence PS, PS, SI, SI, SI.



Forward Algorithm

	α_1	α_2	α_3	α_4	α_5
NA	0.5	0.22	0.02	0.002	0.0003
AP	0.36	0.58	0.23	0.048	0.0087
AC	0.14	0.20	0.75	0.95	0.991



$$\alpha_3 \propto b_3 \odot (A^T \alpha_2)$$

$$= \begin{bmatrix} 0.3 \\ 0.5 \\ 0.8 \end{bmatrix} \odot \left(\begin{bmatrix} 0.25 & 0 & 0 \\ 0.75 & 0.25 & 0 \\ 0 & 0.75 & 1 \end{bmatrix} \begin{bmatrix} 0.22 \\ 0.58 \\ 0.20 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 0.3 \\ 0.5 \\ 0.8 \end{bmatrix} \odot \begin{bmatrix} 0.055 \\ 0.31 \\ 0.635 \end{bmatrix}$$

$$= \begin{bmatrix} 0.0165 \\ 0.155 \\ 0.508 \end{bmatrix}$$

Normalizing, we get:

$$\alpha_3 = \frac{1}{0.6795} \begin{bmatrix} 0.0165 \\ 0.155 \\ 0.508 \end{bmatrix}$$

$$= \begin{bmatrix} 0.02 \\ 0.23 \\ 0.75 \end{bmatrix}$$

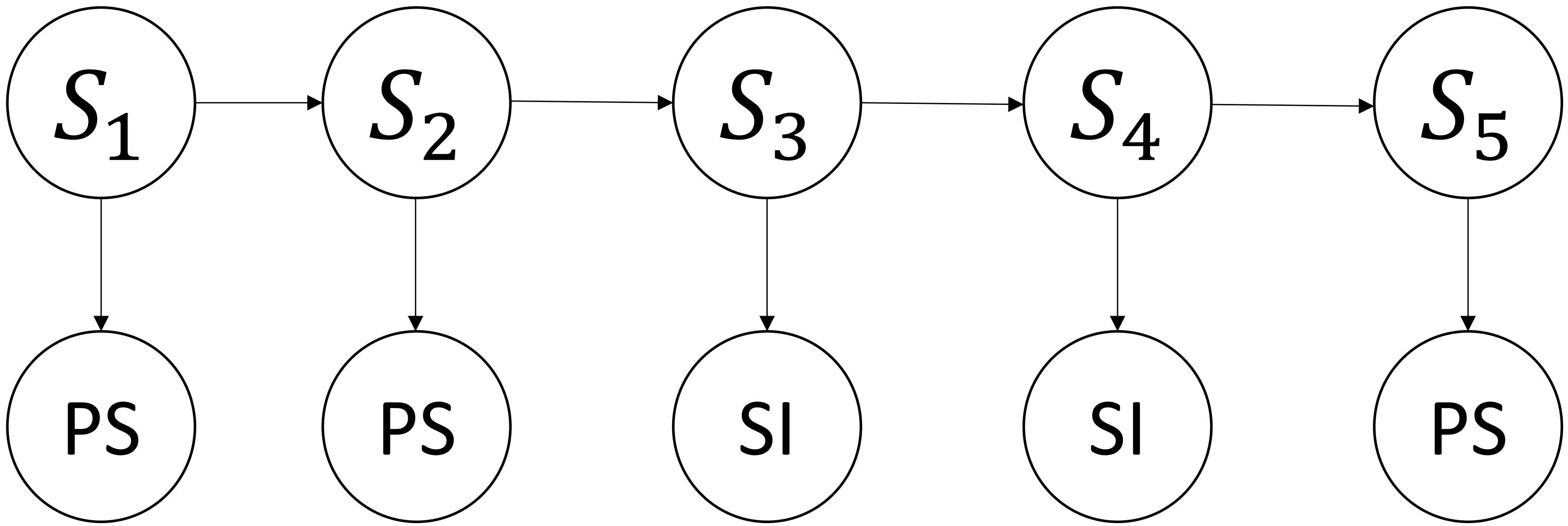
Forward Algorithm

$$\mathbf{B} = \begin{array}{c} \\ NA \\ AP \\ AC \end{array} \begin{array}{c} \begin{array}{cc} PS & SI \end{array} \\ \begin{pmatrix} 0.7 & 0.3 \\ 0.5 & 0.5 \\ 0.2 & 0.8 \end{pmatrix} \end{array} \qquad \boldsymbol{A} = \begin{array}{c} \\ NA \\ AP \\ AC \end{array} \begin{array}{c} \begin{array}{ccc} NA & AP & AC \end{array} \\ \begin{pmatrix} 0.25 & 0.75 & 0 \\ 0 & 0.25 & 0.75 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

States		$< \boldsymbol{PS} > \ (\boldsymbol{t = 1})$	Normalize
NA	$\alpha_1(NA)$	$P(NA)\times P(PS\|NA) = \frac{1}{3}\times 0.7 = 0.23$	$= \frac{0.23}{0.464} = 0.5$
AP	$\alpha_1(AP)$	$P(AP)\times P(PS\|AP) = \frac{1}{3}\times 0.5 = 0.167$	$= \frac{0.167}{0.464} = 0.36$
AC	$\alpha_1(AC)$	$P(AC)\times P(PS\|AC) = \frac{1}{3}\times 0.2 = 0.67$	$= \frac{0.067}{0.464} = 0.14$
		$< \boldsymbol{PS}, \boldsymbol{PS} > \ (\boldsymbol{t = 2})$	
NA	$\alpha_2(NA)$	$\left(\alpha_1(NA)\times P(NA\|NA) + \alpha_1(AP)\times P(NA\|AP) + \alpha_1(AC)P(NA\|AC)\right)\times P(\mathrm{PS}\|NA) =$ $(0.5\times 0.25 + 0.36\times 0 + 0.14\times 0)\times 0.7 = 0.0875$	$= \frac{0.0875}{0.402} = 0.22$
AP	$\alpha_2(AP)$	$\left(\alpha_1(NA)\times P(AP\|NA) + \alpha_1(AP)\times P(AP\|AP) + \alpha_1(AC)P(AP\|AC)\right)\times P(\mathrm{PS}\|AP) =$ $(0.5\times 0.75 + 0.36\times 0.25 + 0.14\times 0)\times 0.5 = 0.2325$	$= \frac{0.2325}{0.402} = 0.58$
AC	$\alpha_2(AC)$	$\left(\alpha_1(NA)\times P(AC\|NA) + \alpha_1(AP)\times P(AC\|AP) + \alpha_1(AC)P(AC\|AC)\right)\times P(\mathrm{PS}\|AC) =$ $(0.5\times 0 + 0.36\times 0.75 + 0.14\times 1)\times 0.2 = 0.082$	$= \frac{0.082}{0.402} = 0.20$

Backward Algorithm

	β_1	β_2	β_3	β_4	β_5
NA	0.11	0.27	0.31	0.45	1
AP	0.04	0.17	0.57	0.725	1
AC	0.03	0.13	0.64	0.8	1



$$\beta_3 = A(b_3 \odot \beta_4)$$

$$= \begin{bmatrix} 0.25 & 0.75 & 0 \\ 0 & 0.25 & 0.75 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 0.3 \\ 0.5 \\ 0.8 \end{bmatrix} \odot \begin{bmatrix} 0.45 \\ 0.725 \\ 0.8 \end{bmatrix} \right)$$

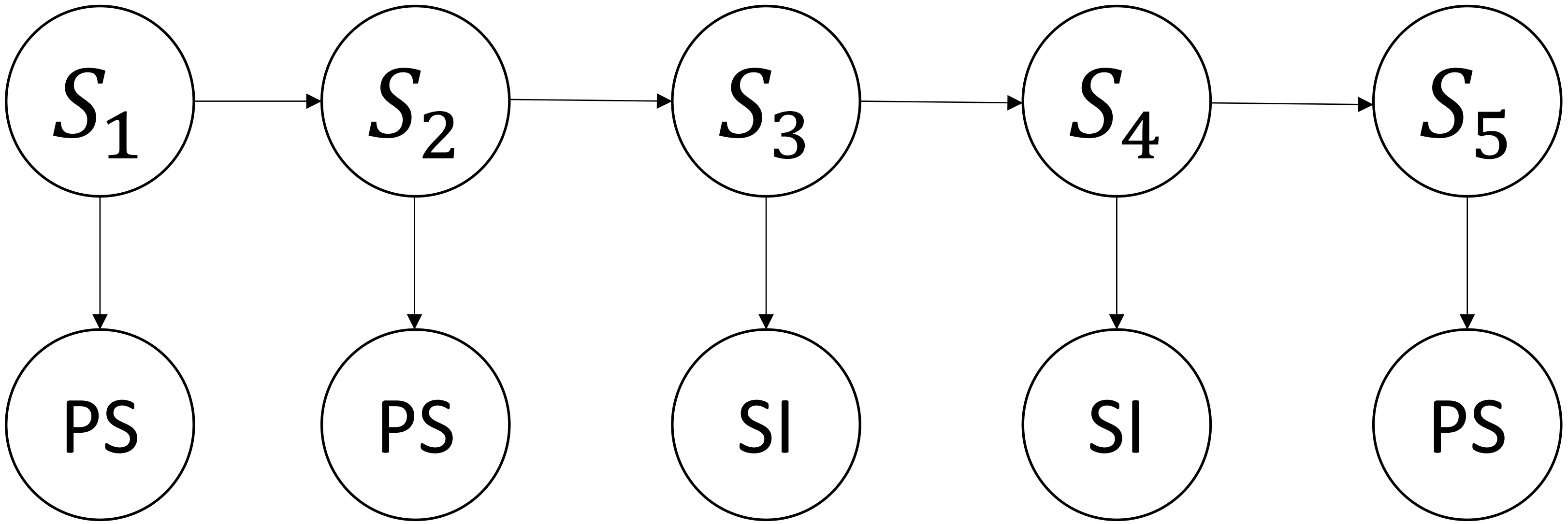
$$= \begin{bmatrix} 0.25 & 0.75 & 0 \\ 0 & 0.25 & 0.75 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.135 \\ 0.3625 \\ 0.64 \end{bmatrix}$$

$$= \begin{bmatrix} 0.31 \\ 0.57 \\ 0.64 \end{bmatrix}$$

Note that $\sum_j \beta_t(j)$ is not necessarily 1.

Gamma calculation (using forwards-backwards)

	γ_1	γ_2	γ_3	γ_4	γ_5
NA	**0.75**	0.32	0.01	0.001	0.0002
AP	0.2	**0.54**	0.21	0.044	0.0088
AC	0.05	0.14	**0.78**	**0.955**	**0.991**



argmax state: NA AP AC AC AC

$$\gamma_3 \propto \alpha_3 \odot \beta_3$$

$$= \begin{bmatrix} 0.02 \\ 0.23 \\ 0.75 \end{bmatrix} \odot \begin{bmatrix} 0.31 \\ 0.57 \\ 0.64 \end{bmatrix}$$

$$= \begin{bmatrix} 0.0062 \\ 0.1311 \\ 0.48 \end{bmatrix}$$

Normalizing, we get:

$$\gamma_3 = \frac{1}{0.6173} \begin{bmatrix} 0.0062 \\ 0.1311 \\ 0.48 \end{bmatrix}$$

$$= \begin{bmatrix} 0.01 \\ 0.21 \\ 0.78 \end{bmatrix}$$

$$S_3^* = \operatorname{argmax}\{\overset{\text{NA}}{0.01}, \overset{\text{AP}}{0.21}, \overset{\text{AC}}{0.78}\}$$

$$S_3^* = \text{AC}$$