



Lecture Slides for

INTRODUCTION TO

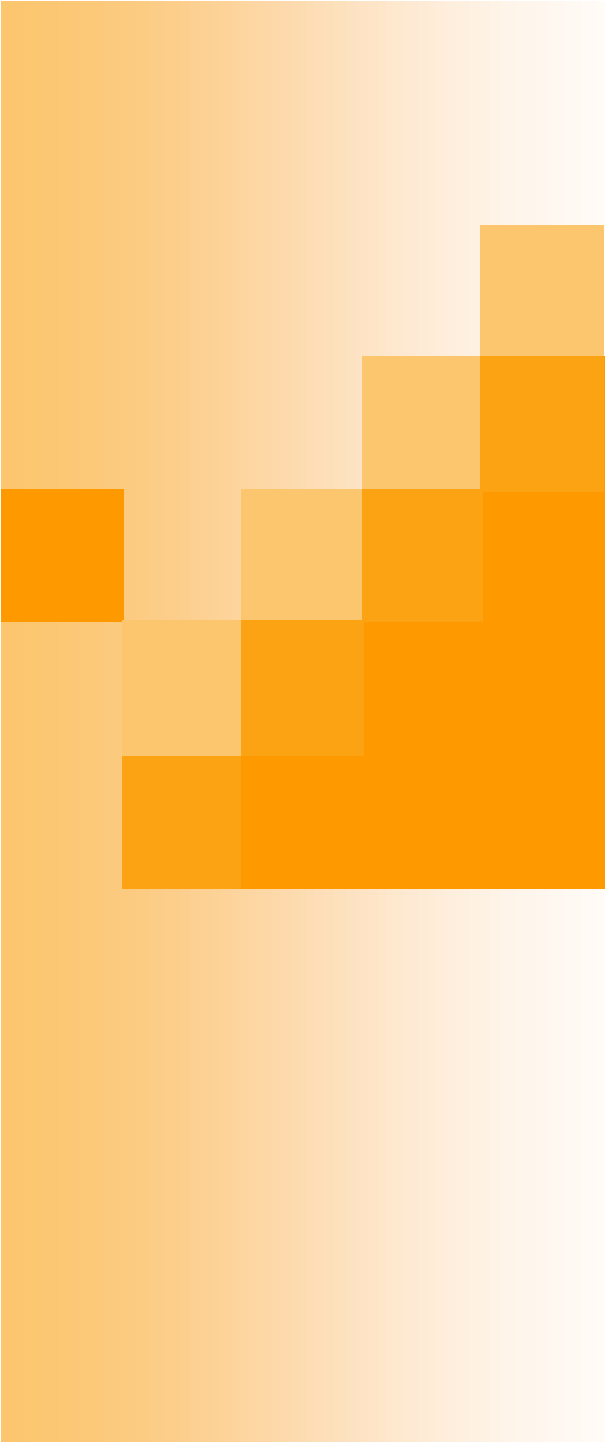
Machine Learning

ETHEM ALPAYDIN

© The MIT Press, 2004

alpaydin@boun.edu.tr

<http://www.cmpe.boun.edu.tr/~ethem/i2ml>



CHAPTER 13:

*Hidden Markov
Models*



Introduction

- Modeling dependencies in input; no longer iid
- Sequences:
 - Temporal: In speech; phonemes in a word (dictionary), words in a sentence (syntax, semantics of the language).
In handwriting, pen movements
 - Spatial: In a DNA sequence; base pairs



Discrete Markov Process

- N states: $S_1, S_2, \dots, S_N$ State at “time” t , $q_t = S_i$
- First-order Markov

$$P(q_{t+1} = S_j | q_t = S_i, q_{t-1} = S_k, \dots) = P(q_{t+1} = S_j | q_t = S_i)$$

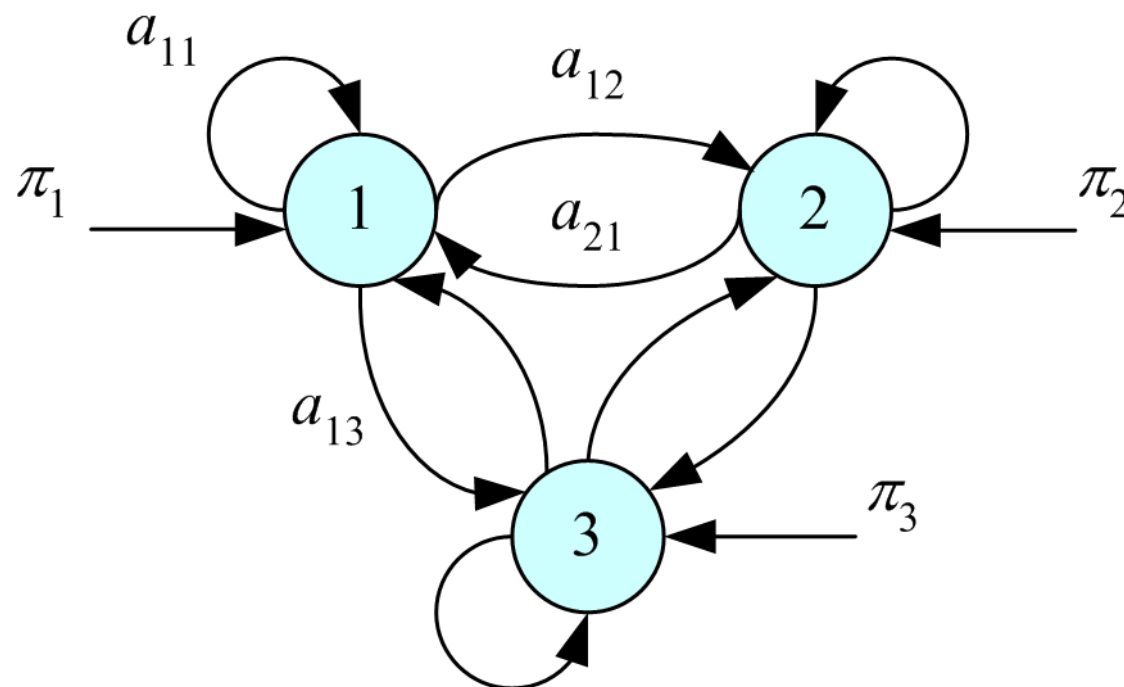
- Transition probabilities

$$a_{ij} \equiv P(q_{t+1} = S_j | q_t = S_i) \quad a_{ij} \geq 0 \text{ and } \sum_{j=1}^N a_{ij} = 1$$

- Initial probabilities

$$\pi_i \equiv P(q_1 = S_i) \quad \sum_{i=1}^N \pi_i = 1$$

Stochastic Automaton



$$P(O = Q | \mathbf{A}, \mathbf{\Pi}) = P(q_1) \prod_{t=2}^T P(q_t | q_{t-1}) = \pi_{q_1} a_{q_1 q_2} \cdots a_{q_{T-1} q_T}$$



Example: Balls and Urns

- Three urns each full of balls of one color

S_1 : red, S_2 = blue, S_3 : green

$$\mathbf{\Pi} = [0.5, 0.2, 0.3]^T \quad \mathbf{A} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

$$O = \{S_1, S_1, S_3, S_3\}$$

$$\begin{aligned} P(O|\mathbf{A}, \mathbf{\Pi}) &= P(S_1) \cdot P(S_1|S_1) \cdot P(S_3|S_1) \cdot P(S_3|S_3) \\ &= \pi_1 \cdot a_{11} \cdot a_{13} \cdot a_{33} \\ &= 0.5 \cdot 0.4 \cdot 0.3 \cdot 0.8 = 0.048 \end{aligned}$$



Balls and Urns: Learning

- Given K example sequences of length T

$$\hat{\pi}_i = \frac{\#\{\text{sequences starting with } S_i\}}{\#\{\text{number of sequences}\}} = \frac{\sum_k 1(q_1^k = S_i)}{K}$$

$$\begin{aligned}\hat{a}_{ij} &= \frac{\#\{\text{transitions from } S_i \text{ to } S_j\}}{\#\{\text{transitions from } S_i\}} \\ &= \frac{\sum_k \sum_{t=1}^{T-1} 1(q_t^k = S_i \text{ and } q_{t+1}^k = S_j)}{\sum_k \sum_{t=1}^{T-1} 1(q_t^k = S_i)}\end{aligned}$$



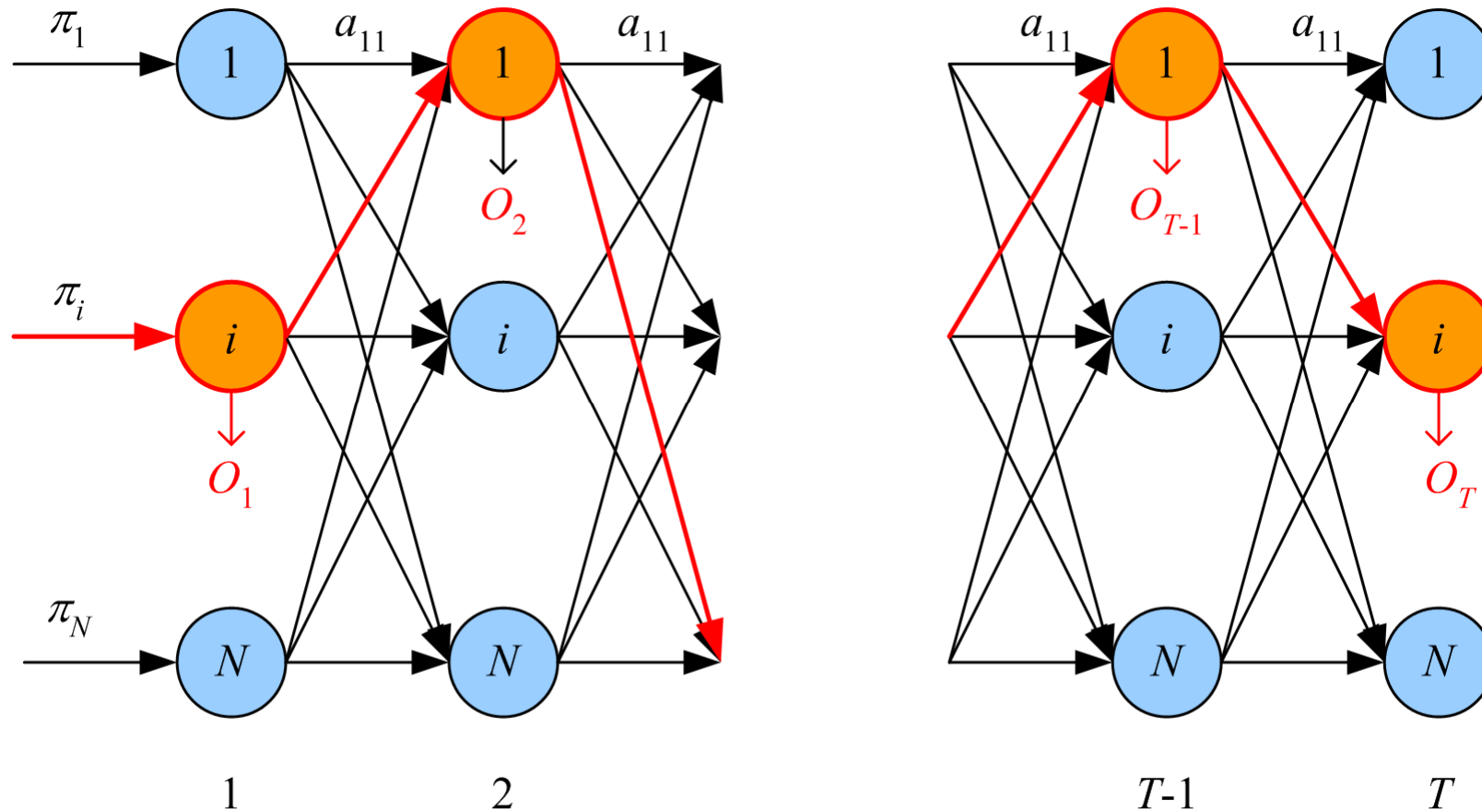
Hidden Markov Models

- States are not observable
- Discrete observations $\{v_1, v_2, \dots, v_M\}$ are recorded; a probabilistic function of the state
- Emission probabilities

$$b_j(m) \equiv P(O_t = v_m | q_t = S_j)$$

- Example: In each urn, there are balls of different colors, but with different probabilities.
- For each observation sequence, there are multiple state sequences

HMM Unfolded in Time





Elements of an HMM

- N : Number of states
- M : Number of observation symbols
- $\mathbf{A} = [a_{ij}]$: N by N state transition probability matrix
- $\mathbf{B} = b_j(m)$: N by M observation probability matrix
- $\mathbf{\Pi} = [\pi_i]$: N by 1 initial state probability vector

$\lambda = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$, parameter set of HMM



Three Basic Problems of HMMs

1. **Evaluation:** Given λ , and O , calculate $P(O | \lambda)$
2. **State sequence:** Given λ , and O , find Q^* such that
$$P(Q^* | O, \lambda) = \max_Q P(Q | O, \lambda)$$
3. **Learning:** Given $\mathcal{X} = \{O^k\}_k$, find λ^* such that
$$P(\mathcal{X} | \lambda^*) = \max_{\lambda} P(\mathcal{X} | \lambda)$$

(Rabiner, 1989)

Evaluation

- Forward variable:

$$\alpha_t(i) \equiv P(O_1 \cdots O_t, q_t = S_i | \lambda)$$

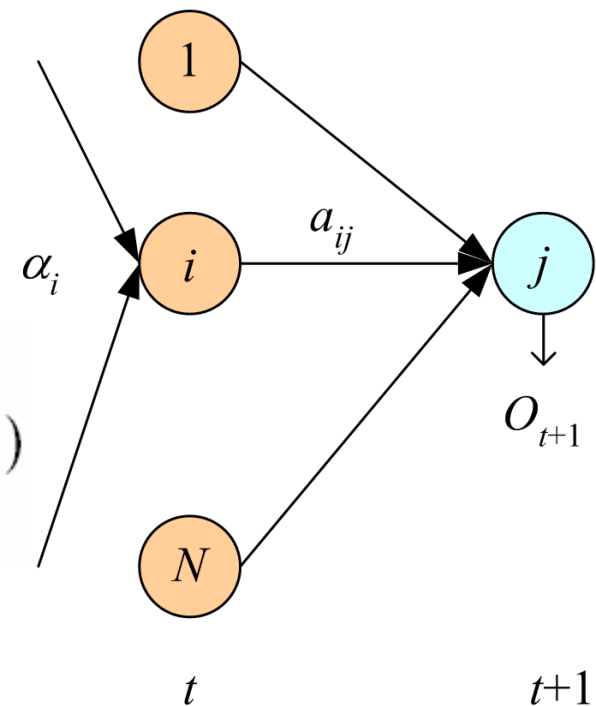
- Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1)$$

- Recursion:

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

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i)$$



- Backward variable:

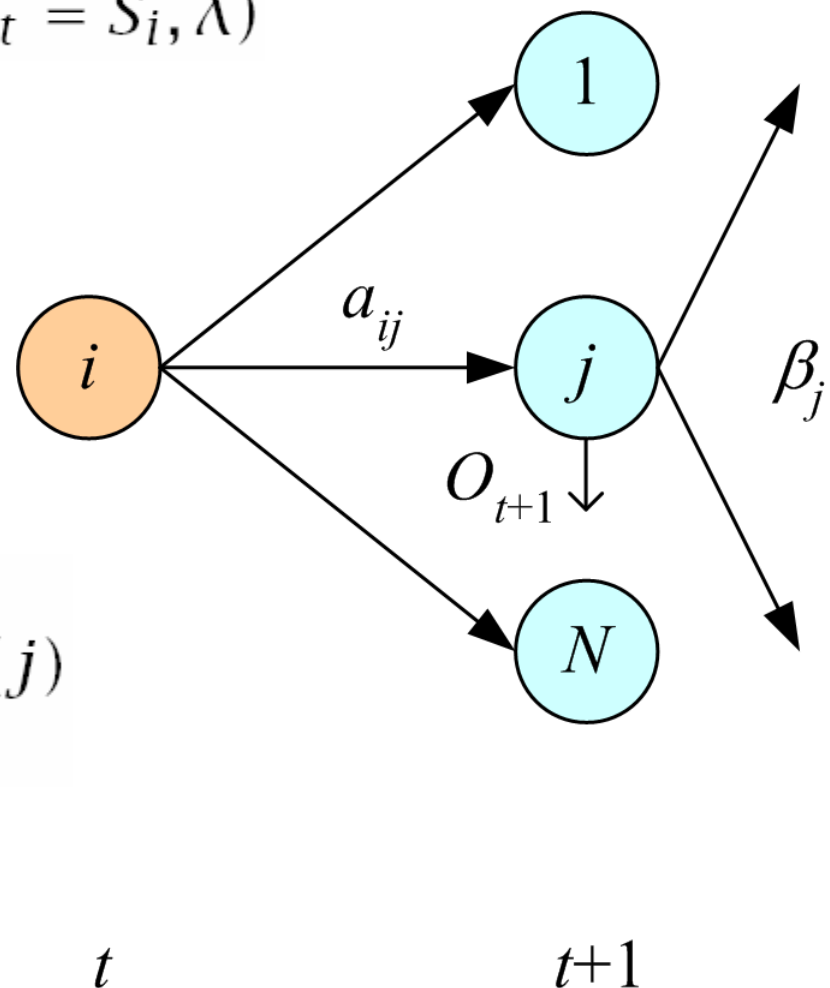
$$\beta_t(i) \equiv P(O_{t+1} \cdots O_T | q_t = S_i, \lambda)$$

- Initialization:

$$\beta_T(i) = 1$$

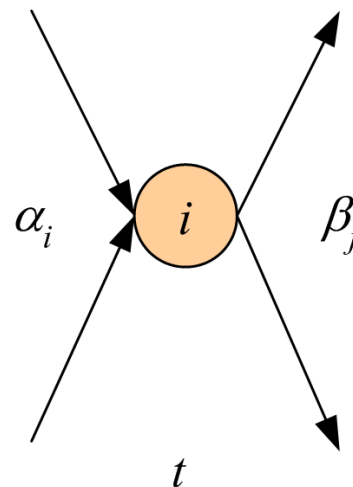
- Recursion:

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



Finding the State Sequence

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)}$$



Choose the state that has the highest probability,
for each time step:

$$q_t^* = \arg \max_i \gamma_t(i)$$

No!



Viterbi's Algorithm

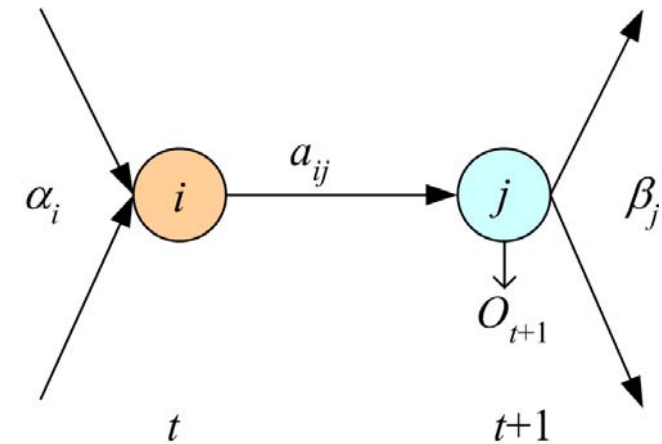
$$\delta_t(i) \equiv \max_{q_1 q_2 \cdots q_{t-1}} p(q_1 q_2 \cdots q_{t-1}, q_t = S_i, O_1 \cdots O_t | \lambda)$$

1. Initialization: $\delta_1(i) = \pi_i b_i(O_1) \quad \psi_1(i) = 0$
2. Recursion:
$$\delta_t(j) = \max_i \delta_{t-1}(i) a_{ij} \cdot b_j(O_t)$$
$$\psi_t(j) = \arg \max_i \delta_{t-1}(i) a_{ij}$$
3. Termination: $p^* = \max_i \delta_T(i) \quad q_T^* = \arg \max_i \delta_T(i)$
4. Path backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1$$

Learning

$$\xi_t(i, j) \equiv P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$$



$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_k \sum_l \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(l)}$$

Baum-Welch algorithm (EM)

$$z_i^t = \begin{cases} 1 & \text{if } q_t = S_i \\ 0 & \text{otherwise} \end{cases} \quad z_{ij}^t = \begin{cases} 1 & \text{if } q_t = S_i \text{ and } q_{t+1} = S_j \\ 0 & \text{otherwise} \end{cases}$$

Baum-Welch (EM)

■ E-step:

$$E[z_i^t] = \gamma_t(i)$$
$$E[z_{ij}^t] = \xi_t(i, j)$$

■ M-step:

$$\hat{a}_{ij} = \frac{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \xi_t^k(i, j)}{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \gamma_t^k(i)}$$
$$\hat{b}_j(m) = \frac{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \gamma_t^k(j) 1(O_t^k = v_m)}{\sum_{k=1}^K \sum_{t=1}^{T_k-1} \gamma_t^k(j)}$$
$$\hat{\pi}_i = \frac{\sum_{k=1}^K \gamma_1^k(i)}{K}$$

Continuous Observations

- Discrete:
$$P(O_t | q_t = S_j, \lambda) = \prod_{m=1}^M b_j(m)^{r_m^t} \quad r_m^t = \begin{cases} 1 & \text{if } O_t = v_m \\ 0 & \text{otherwise} \end{cases}$$

- Gaussian mixture (Discretize using k -means):

$$p(O_t | q_t = S_j, \lambda) = \sum_{l=1}^L P(G_{jl}) p(O_t | q_t = S_j, G_l, \lambda) \\ \sim \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)$$

- Continuous:

$$p(O_t | q_t = S_j, \lambda) \sim \mathcal{N}(\mu_j, \sigma_j^2)$$

Use EM to learn parameters, e.g., $\hat{\mu}_j = \frac{\sum_t y_t(j) O_t}{\sum_t y_t(j)}$



HMM with Input

- Input-dependent observations:

$$p(O_t | q_t = S_j, \mathbf{x}_t, \lambda) \sim \mathcal{N}(g_j(\mathbf{x}_t | \theta_j), \sigma_j^2)$$

- Input-dependent transitions (Meila and Jordan, 1996; Bengio and Frasconi, 1996):

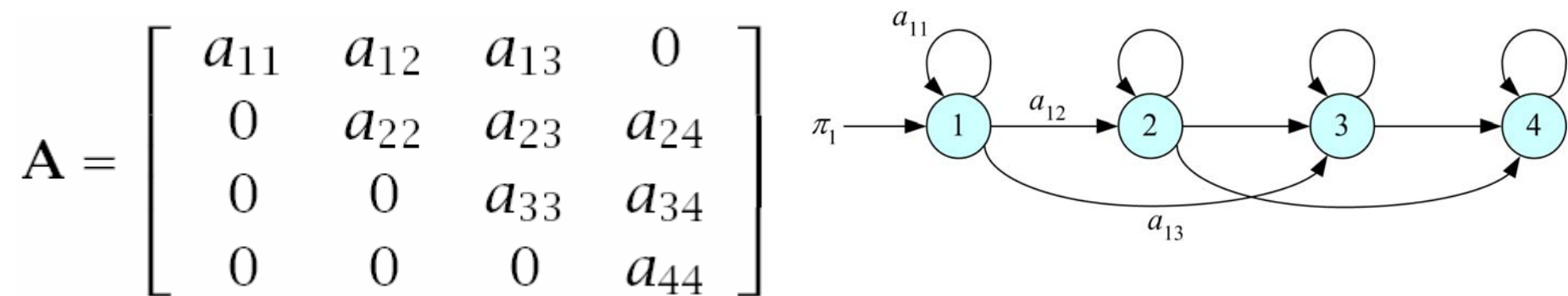
$$P(q_{t+1} = S_j | q_t = S_i, \mathbf{x}_t)$$

- Time-delay input:

$$\mathbf{x}_t = f(O_{t-\tau}, \dots, O_{t-1})$$

Model Selection in HMM

- Left-to-right HMMs:



- In classification, for each C_i , estimate $P(O | \lambda_i)$ by a separate HMM and use Bayes' rule

$$P(\lambda_i | O) = \frac{P(O | \lambda_i) P(\lambda_i)}{\sum_j P(O | \lambda_j) P(\lambda_j)}$$