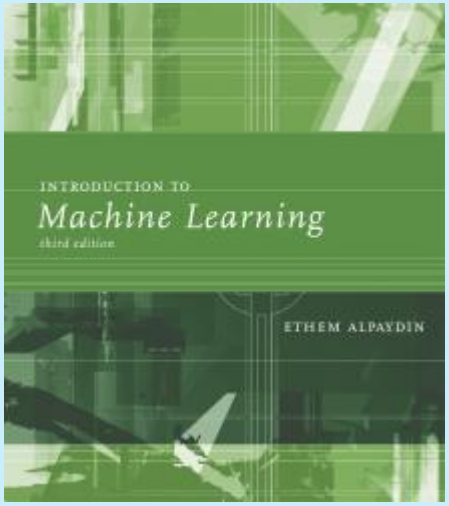


Lecture Slides for

INTRODUCTION TO MACHINE LEARNING

3RD EDITION

ETHEM ALPAYDIN

alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2ml3e

CHAPTER 15:

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: $S_1, S_2, ..., S_N$ State at "time" t, $q_t = S_i$
- First-order Markov

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

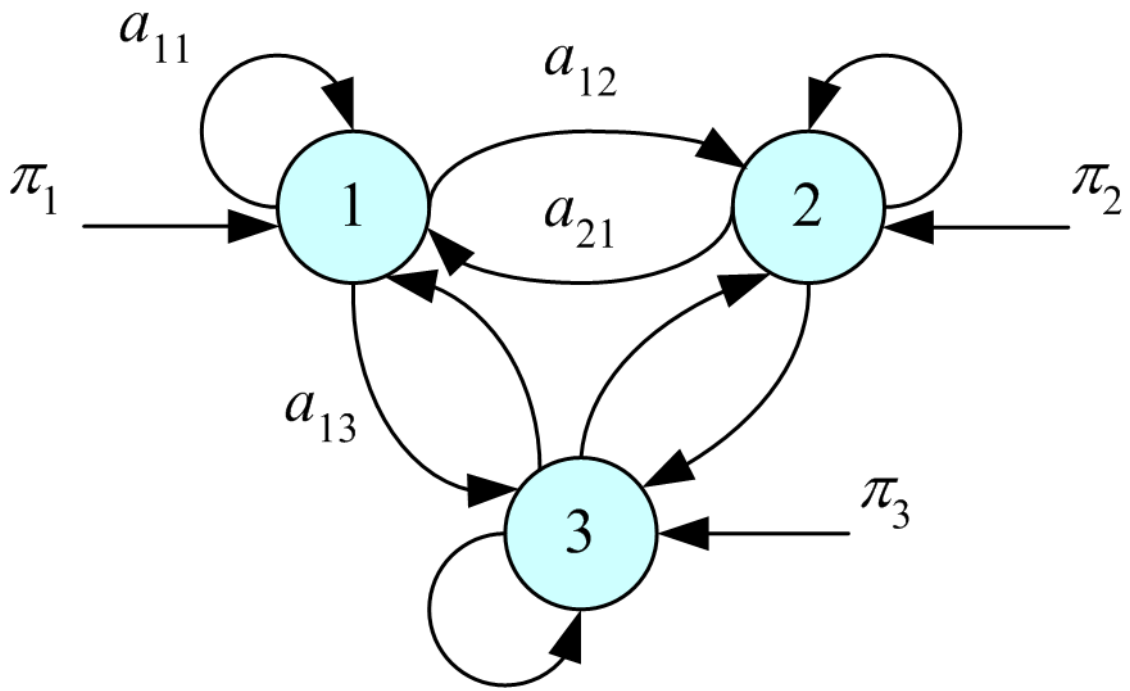
- Transition probabilities

$$a_{ij} \equiv P(q_{t+1}=S_j \mid q_t=S_i) \qquad a_{ij} \geq 0 \text{ and } \Sigma^N_{j=1}\, a_{ij}=1$$

- Initial probabilities

$$\pi_i \equiv P(q_1=S_i) \qquad \Sigma^N_{j=1}\, \pi_i=1$$

Stochastic Automaton



Example: Balls and Urns

- Three urns each full of balls of one color

  S_1: red, S_2: blue, S_3: green

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

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

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

Balls and Urns: Learning

- Given K example sequences of length T

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

$$\hat{a}_{ij} = \frac{\#\{\text{transitions from } S_i \text{ to } S_j\}}{\#\{\text{transitions from } S_i\}}$$

$$= \frac{\sum_k \sum_{t=1}^{T-1} 1\left(q_t^k = S_i \text{ and } q_{t+1}^k = S_j\right)}{\sum_k \sum_{t=1}^{T-1} 1\left(q_t^k = S_i\right)}$$

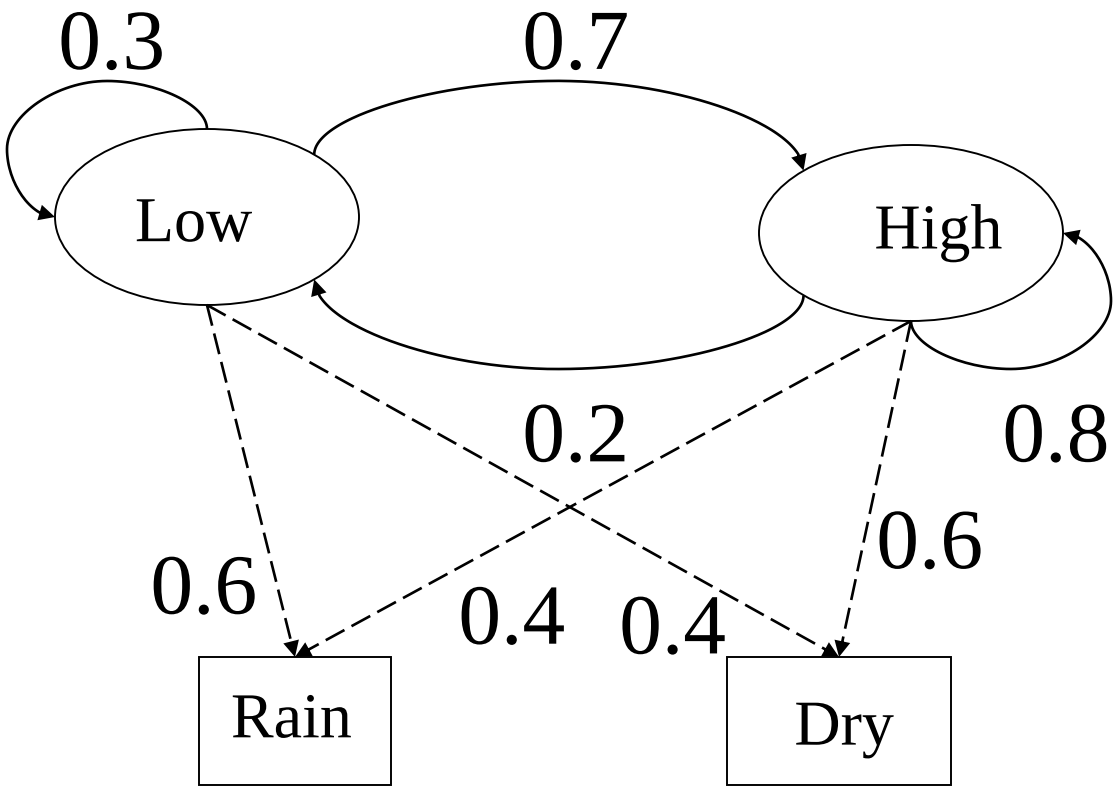
Hidden Markov Models

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

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

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

Example of Hidden Markov Model



Example of Hidden Markov Model

- Two states : ‘Low’ and ‘High’ atmospheric pressure.
- Two observations : ‘Rain’ and ‘Dry’.
- Transition probabilities: P(‘Low’|‘Low’)=0.3 , P(‘High’|‘Low’)=0.7 , P(‘Low’|‘High’)=0.2, P(‘High’|‘High’)=0.8
- Observation probabilities : P(‘Rain’|‘Low’)=0.6 , P(‘Dry’|‘Low’)=0.4 , P(‘Rain’|‘High’)=0.4 , P(‘Dry’|‘High’)=0.6 .
- Initial probabilities: say P(‘Low’)=0.4 , P(‘High’)=0.6 .

Calculation of observation sequence probability

- Suppose we want to calculate a probability of a sequence of observations in our example, {'Dry','Rain'}.
- Consider all possible hidden state sequences:

$$P(\{\text{'Dry','Rain'}\}) = P(\{\text{'Dry','Rain'}\}, \{\text{'Low','Low'}\}) + P(\{\text{'Dry','Rain'}\}, \{\text{'Low','High'}\}) + P(\{\text{'Dry','Rain'}\}, \{\text{'High','Low'}\}) + P(\{\text{'Dry','Rain'}\}, \{\text{'High','High'}\})$$

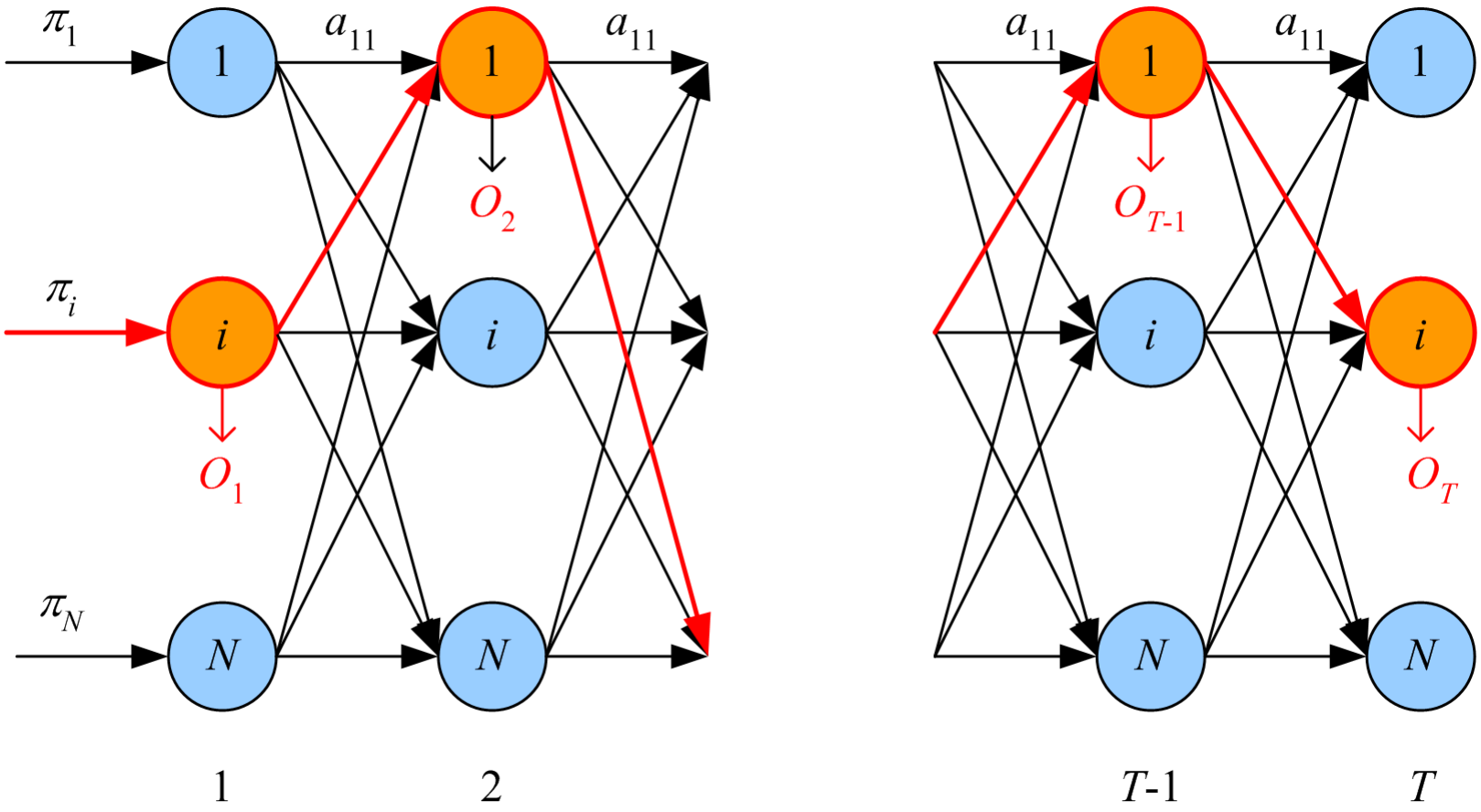
❖where first term is :

$$P(\{\text{'Dry','Rain'}\}, \{\text{'Low','Low'}\}) =$$

$$P(\{\text{'Dry','Rain'}\} \mid \{\text{'Low','Low'}\})\ P(\{\text{'Low','Low'}\}) =$$

$$P(\text{'Dry'}|\text{'Low'})P(\text{'Rain'}|\text{'Low'})\ P(\text{'Low'})P(\text{'Low'}|\text{'Low}) = 0.4\times0.6\times0.4\times0.3=0.0288$$

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

observation sequence $O = \{O_1 \quad O_2 \quad \dots \quad O_T\}$

the state sequence $Q = \{q_1 \quad q_2 \quad \dots \quad q_T\}$

1. Evaluation: Given λ, and O, calculate $P(O \mid \lambda)$
2. State sequence: Given λ, and O, find Q^* such that

   $P(Q^* \mid O, \lambda) = \max_Q P(Q \mid O, \lambda)$
3. Learning: Given $X=\{O^k\}_k$, find λ^* such that

   $P(X \mid \lambda^*)=\max_\lambda P(X \mid \lambda)$

(Rabiner, 1989)

1-Evaluation

Given observation sequence $O = \{O_1 \quad O_2 \quad \dots \quad O_T\}$

and the state sequence $Q = \{q_1 \quad q_2 \quad \dots \quad q_T\}$

$$P(O|Q,\lambda) = \prod_{t=1}^{T} P(O_t|q_t,\lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdots b_{q_T}(O_T)$$

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

$$\begin{aligned} P(O,Q|\lambda) &= P(q_1) \prod_{t=2}^{T} P(q_t|q_{t-1}) \prod_{t=1}^{T} P(O_t|q_t) \\ &= \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T) \end{aligned}$$

$$P(O|\lambda) = \sum_{\text{all possible } Q} P(O,Q|\lambda)$$

there are N^T possible Q!

Evaluation

- Forward variable:

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

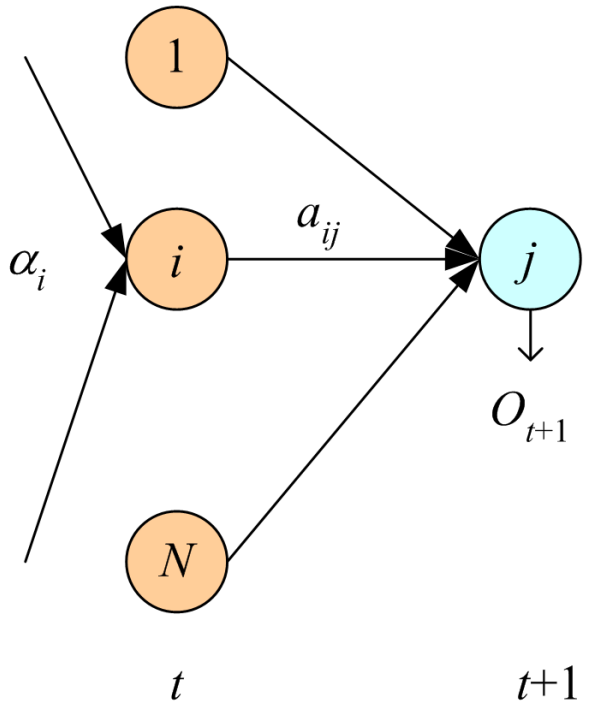
Initialization:

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

Recursion:

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

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



Forward

Backward variable:

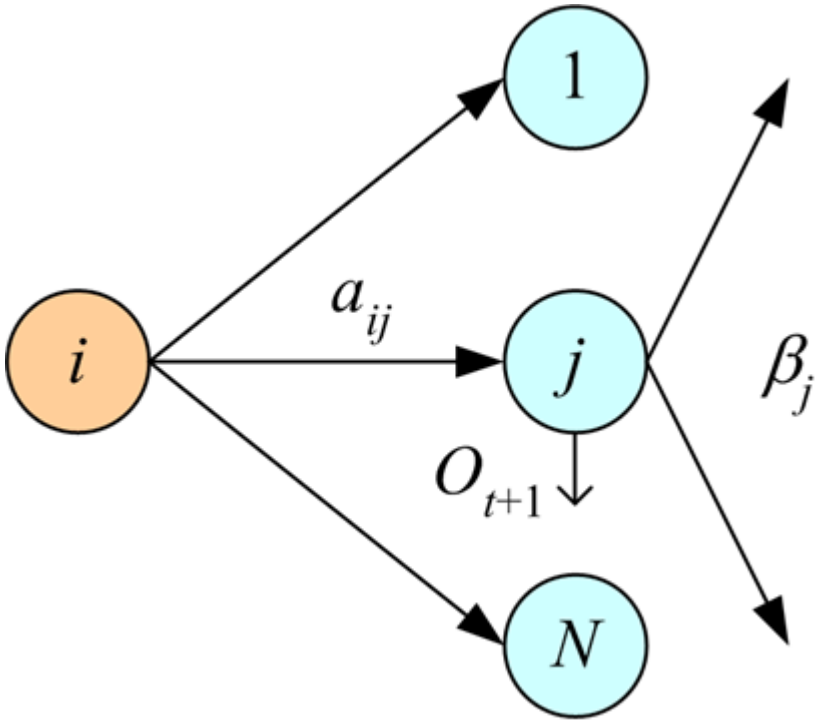
$$\beta_t(i) \equiv P(O_{t+1} \cdots O_T | q_t = S_i, \lambda)$$

Initialization:

$$\beta_T(i) = 1$$

Recursion:

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

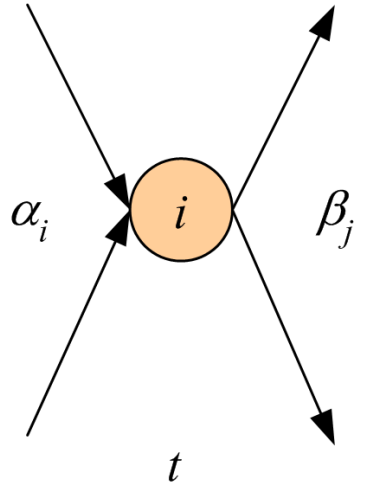


Backward

2-Finding the State Sequence

$$\gamma_t(i) \equiv P(q_t = S_i | O, \lambda)$$

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



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

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

Viterbi's Algorithm

$$\delta_t(i) \equiv \max_{q1q2\cdots qt\text{-}1} p(q_1 q_2 \cdots q_{t\text{-}1}, q_t = S_i, O_1 \cdots O_t \mid \lambda)$$

- Initialization:

  $\delta_1(i) = \pi_i b_i(O_1)$, $\psi_1(i) = 0$
- Recursion:

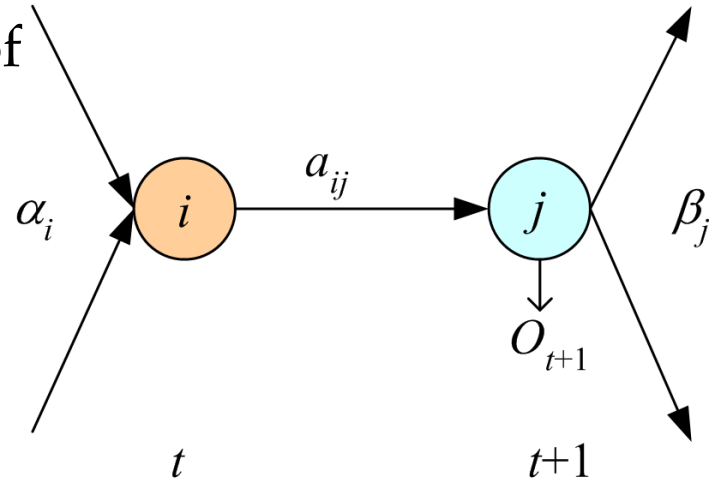
  $\delta_t(j) = \max_i \delta_{t\text{-}1}(i) a_{ij} b_j(O_t)$, $\psi_t(j) = \text{argmax}_i \delta_{t\text{-}1}(i) a_{ij}$
- Termination:

  $p^* = \max_i \delta_T(i)$, $q_T^* = \text{argmax}_i \delta_T(i)$
- Path backtracking:

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

Learning

The approach is ML, and we would like to calculate λ^* that maximizes the likelihood of the sample of training sequences, $X = \{O^k\}$ $k = 1, \ldots, K$ namely, $P(X \mid \lambda)$.



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

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

$$\gamma_i(t) = \sum_{j=1}^{N} \xi_t(i,j)$$

Baum-Welch algorithm (EM):

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

Baum-Welch (EM)

$$\text{E}-\text{step: } E\left[z_i^t\right]=\gamma_t(i) \qquad E\left[z_{ij}^t\right]=\xi_t(i,j)$$

$$\text{M}-\text{step:}$$

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

$$\hat{b}_j(m)=\frac{\sum_{k=1}^{K}\sum_{t=1}^{T_k}\gamma_t^k(j)1\left(O_t^k=v_m\right)}{\sum_{k=1}^{K}\sum_{t=1}^{T_k}\gamma_t^k(i)}$$

* Continuous Observations

- Discrete:

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

- Gaussian mixture (Discretize using *k*-means):

$$P\left(O_t|q_t = S_j, \lambda\right) = \sum_{l=1}^{L} P\left(\mathcal{G}_{jl}\right) p\left(O_t|q_t = S_j, \mathcal{G}_l, \lambda\right) \sim \mathcal{N}\left(\mu_l, \Sigma_l\right)$$

- Continuous: $P\left(O_t|q_t = S_j, \lambda\right) \sim \mathcal{N}\left(\mu_j, \sigma_j^2\right)$

Use EM to learn parameters, e.g., $\hat{\mu}_j = \frac{\sum_t \gamma_t\left(j\right) O_t}{\sum_t \gamma_t\left(j\right)}$, $\hat{\sigma}_j^2 = \frac{\sum_t \gamma_t\left(j\right)\left(O_t - \hat{\mu}_j\right)^2}{\sum_t \gamma_t\left(j\right)}$

* HMM with Input

- Input-dependent observations:

$$P\left(O_t|q_t=S_j,x^t,\lambda\right)\sim\mathcal{N}\left(g_j\left(x^t|\theta_j\right),\sigma_j^2\right)$$

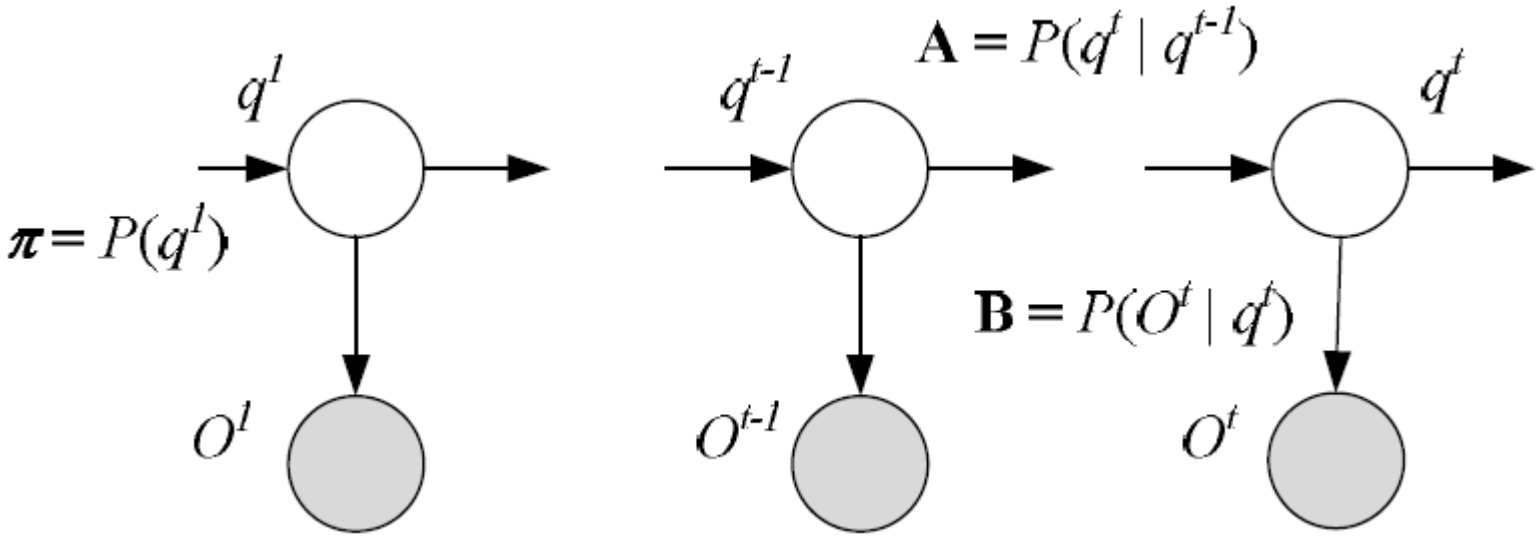
- Input-dependent transitions (Meila and Jordan, 1996; Bengio and Frasconi, 1996):

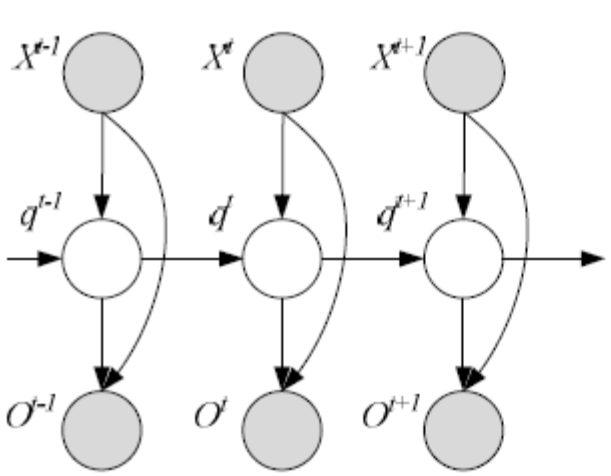
$$P\left(q_{t+1}=S_j|q_t=S_i,x^t\right)$$

- Time-delay input:

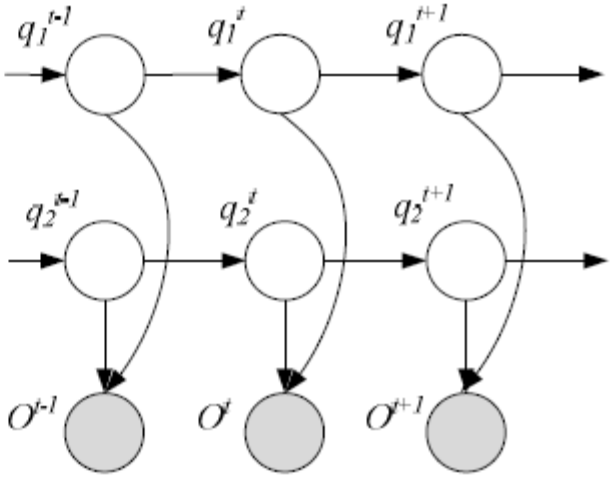
$$\mathbf{x}^t=\mathbf{f}\left(O_{t-\tau},...,O_{t-1}\right)$$

* HMM as a Graphical Model

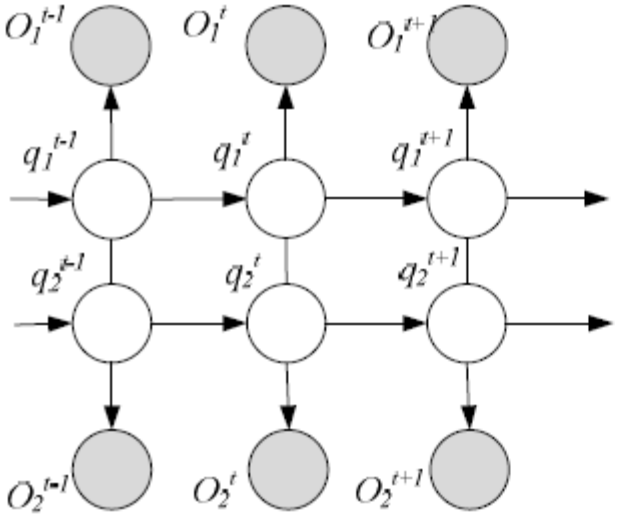




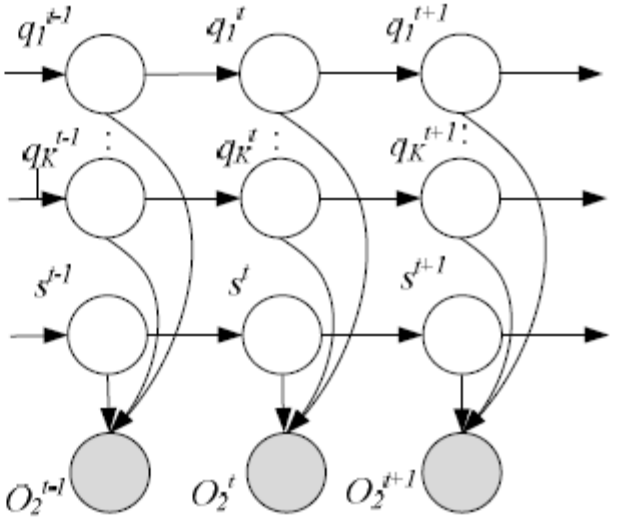
(a) Input-output HMM



(b) Factorial HMM



(c) Coupled HMM

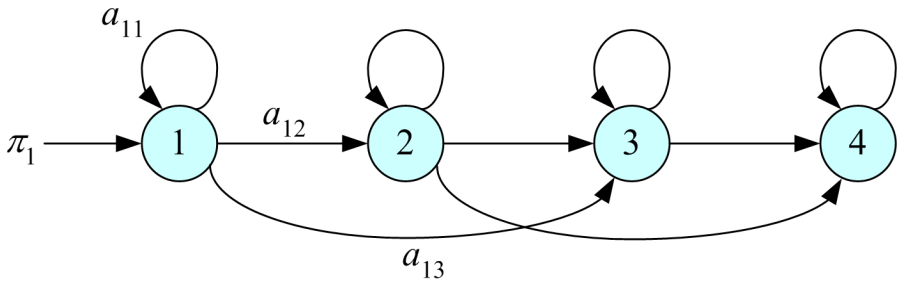


(d) Switching HMM

* Model Selection in HMM

- Left-to-right HMMs:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$



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

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