CSCI933 Machine Learning: Algorithms and Applications

Central China Normal University Wollongong Joint Institute

Graphical Models

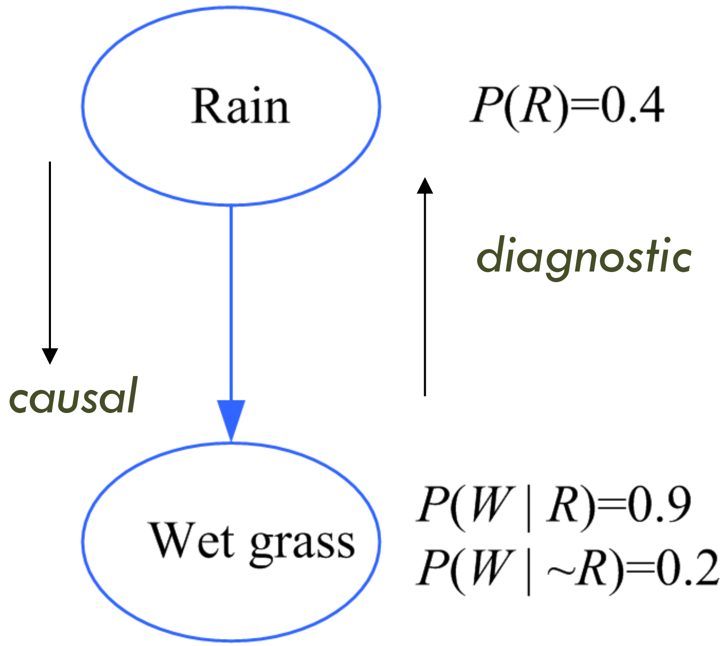
Outline

- Graphical Models
- Hidden Markov Models

Graphical Models

- Aka Bayesian networks, probabilistic networks
- Nodes are hypotheses (random vars) and the probabilities corresponds to our belief in the truth of the hypothesis
- Arcs are direct influences between hypotheses
- The structure is represented as a directed acyclic graph (DAG)
- The parameters are the conditional probabilities in the arcs (Pearl, 1988, 2000; Jensen, 1996; Lauritzen, 1996)

Causes and Bayes' Rule



Diagnostic inference:
Knowing that the grass is wet, what is the probability that rain is the cause?

$$P(R|W) = \frac{P(W|R)P(R)}{P(W)}$$

$$= \frac{P(W|R)P(R)}{P(W|R)P(R) + P(W|\sim R)P(\sim R)}$$

$$= \frac{0.9 \times 0.4}{0.9 \times 0.4 + 0.2 \times 0.6} = 0.75$$

Conditional Independence

- X and Y are independent if

$$P(X,Y)=P(X)P(Y)$$

- X and Y are conditionally independent given Z if

$$P(X,Y|Z)=P(X|Z)P(Y|Z)$$

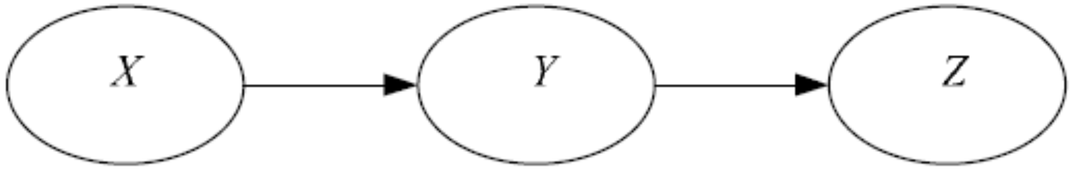
or

$$P(X|Y,Z)=P(X|Z)$$

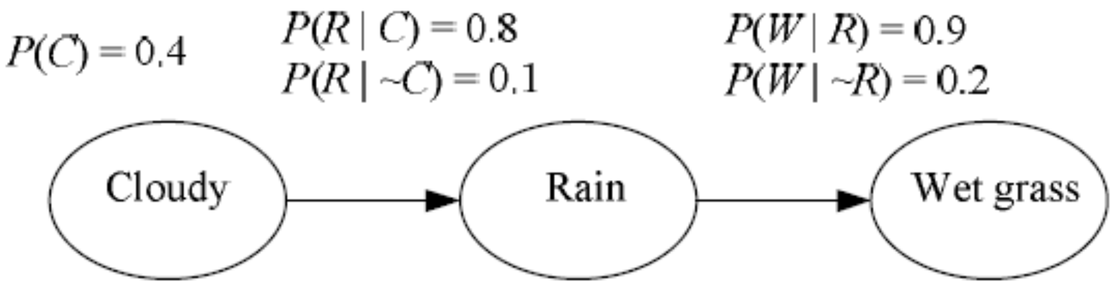
- Three canonical cases: Head-to-tail, Tail-to-tail, head-to-head

Case 1: Head-to-Head

- $P(X,Y,Z)=P(X)P(Y|X)P(Z|Y)$



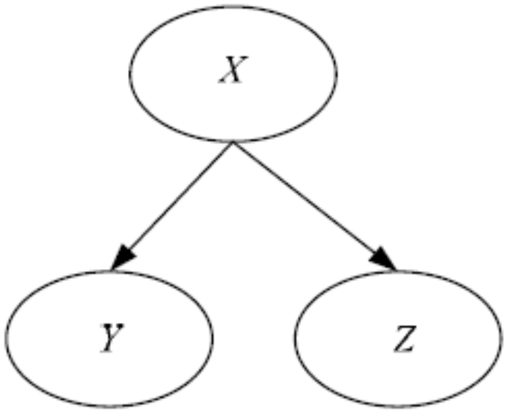
(a) Model

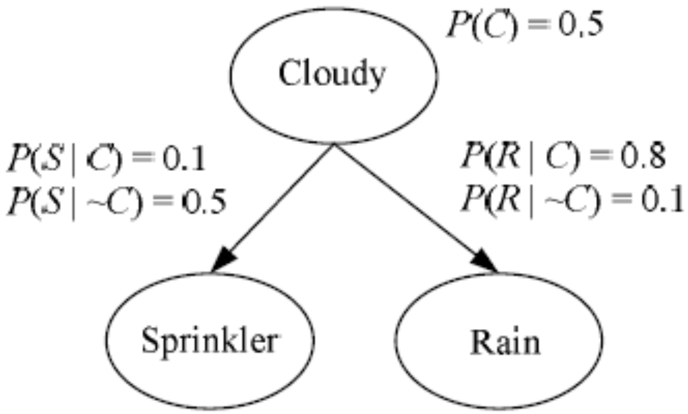


- $P(W|C)=P(W|R)P(R|C)+P(W|\sim R)P(\sim R|C)$

Case 2: Tail-to-Tail

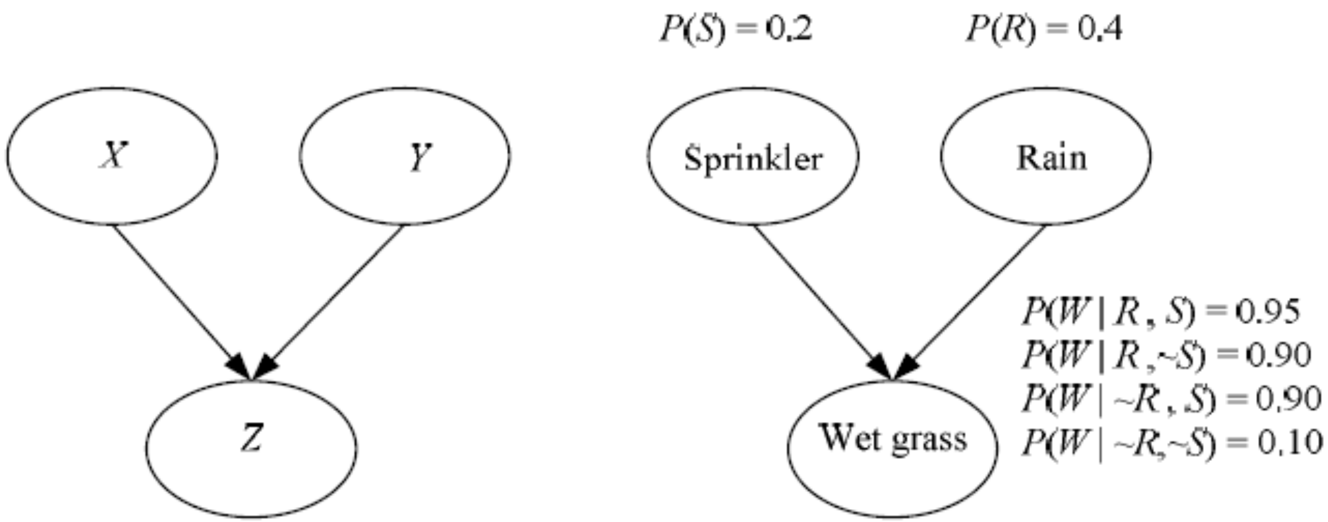
- $P(X,Y,Z)=P(X)P(Y|X)P(Z|X)$



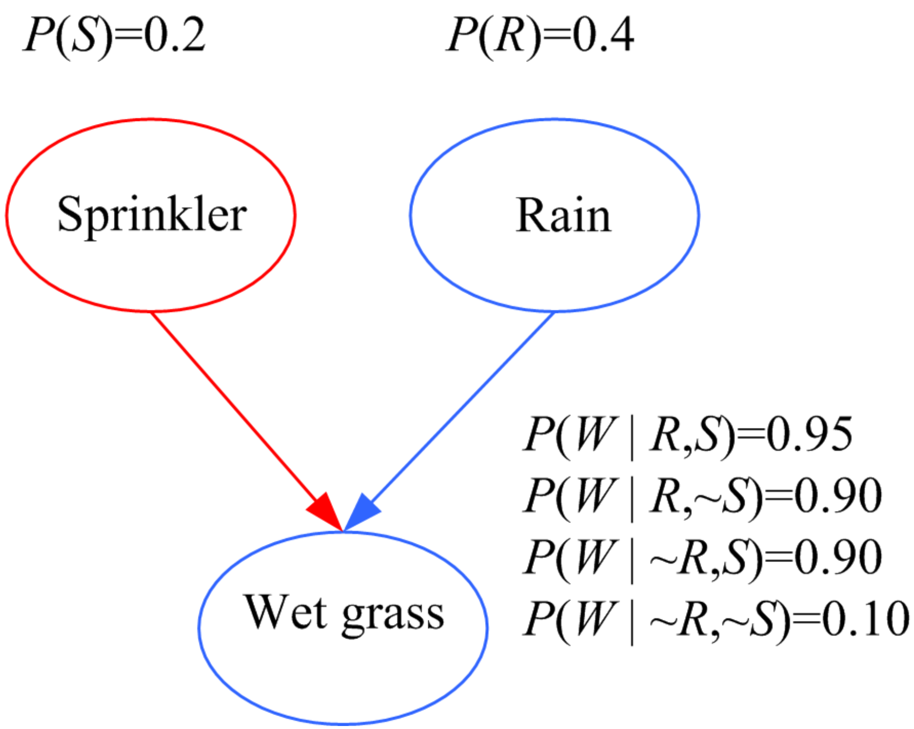


Case 3: Head-to-Head

- $P(X,Y,Z)=P(X)P(Y)P(Z|X,Y)$



Causal vs Diagnostic Inference



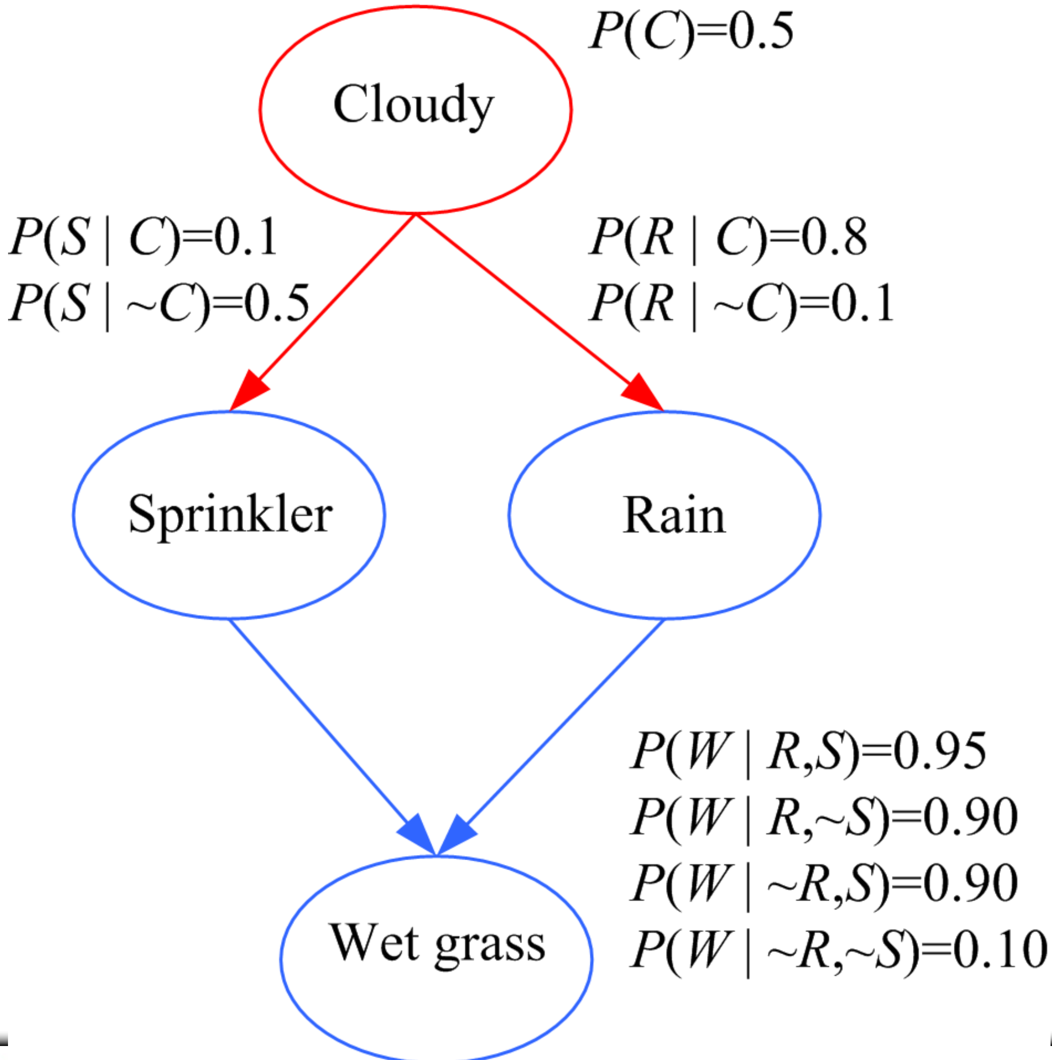
Causal inference: If the sprinkler is on, what is the probability that the grass is wet?

$$P(W|S) = P(W|R,S)\,P(R|S) + P(W|\sim R,S)\,P(\sim R|S)$$
$$= P(W|R,S)\,P(R) + P(W|\sim R,S)\,P(\sim R)$$
$$= 0.95\ 0.4 + 0.9\ 0.6 = 0.92$$

Diagnostic inference: If the grass is wet, what is the probability that the sprinkler is on? $P(S|W) = 0.35 > 0.2\ P(S)$

$P(S|R,W) = 0.21$ *Explaining away: Knowing that it has rained decreases the probability that the sprinkler is on.*

Causes



Causal inference:

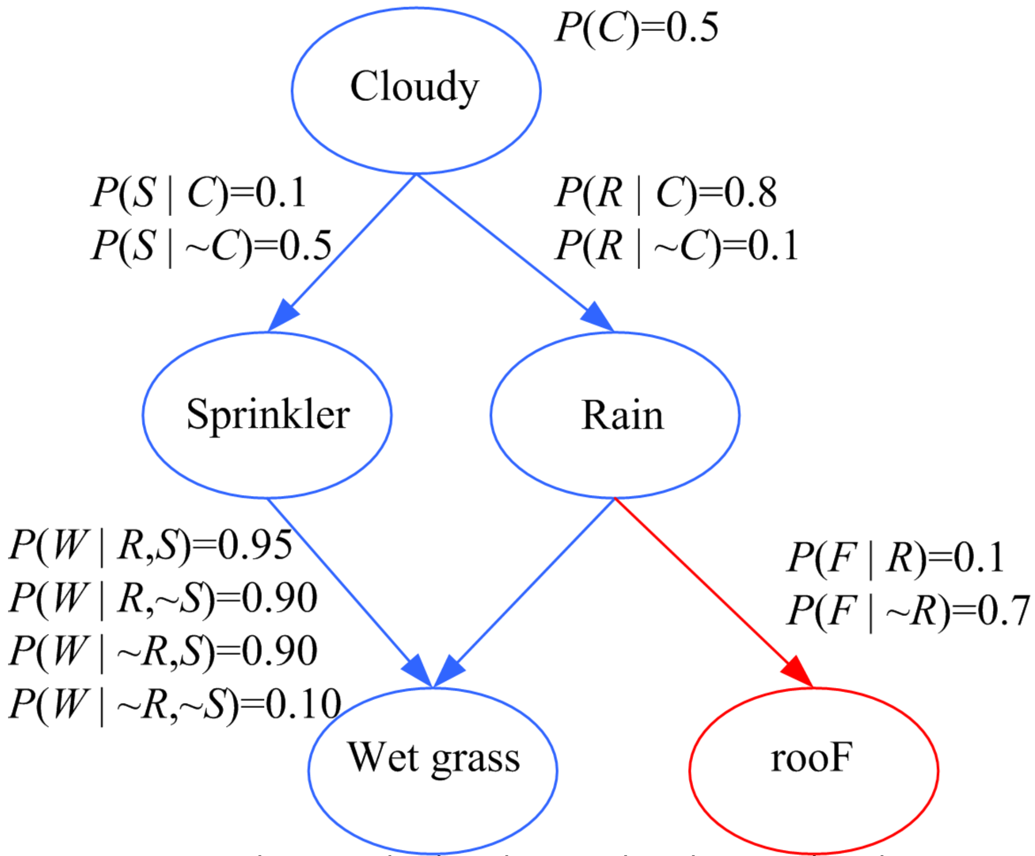
$$
\begin{aligned}
P(W|C) = \; & P(W|R,S)\, P(R,S|C) + \\
& P(W|\sim R,S)\, P(\sim R,S|C) + \\
& P(W|R,\sim S)\, P(R,\sim S|C) + \\
& P(W|\sim R,\sim S)\, P(\sim R,\sim S|C)
\end{aligned}
$$

and use the fact that

$$P(R,S|C) = P(R|C)\, P(S|C)$$

Diagnostic: $P(C|W) = ?$

Exploiting the Local Structure



$P(F \mid C) = ?$

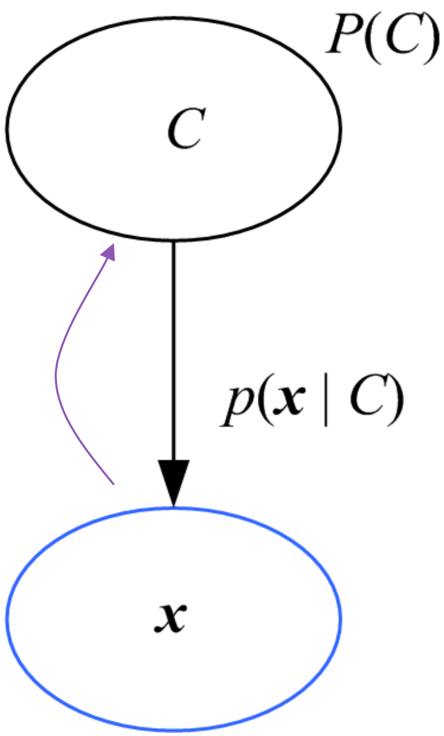
$$P(C,S,R,W,F)=P(C)P(S|C)P(R|C)P(W|S,R)P(F|R)$$

$$P(X_1,\ldots X_d)=\prod_{i=1}^{d}P(X_i \mid \text{parents}(X_i))$$

Classification

diagnostic

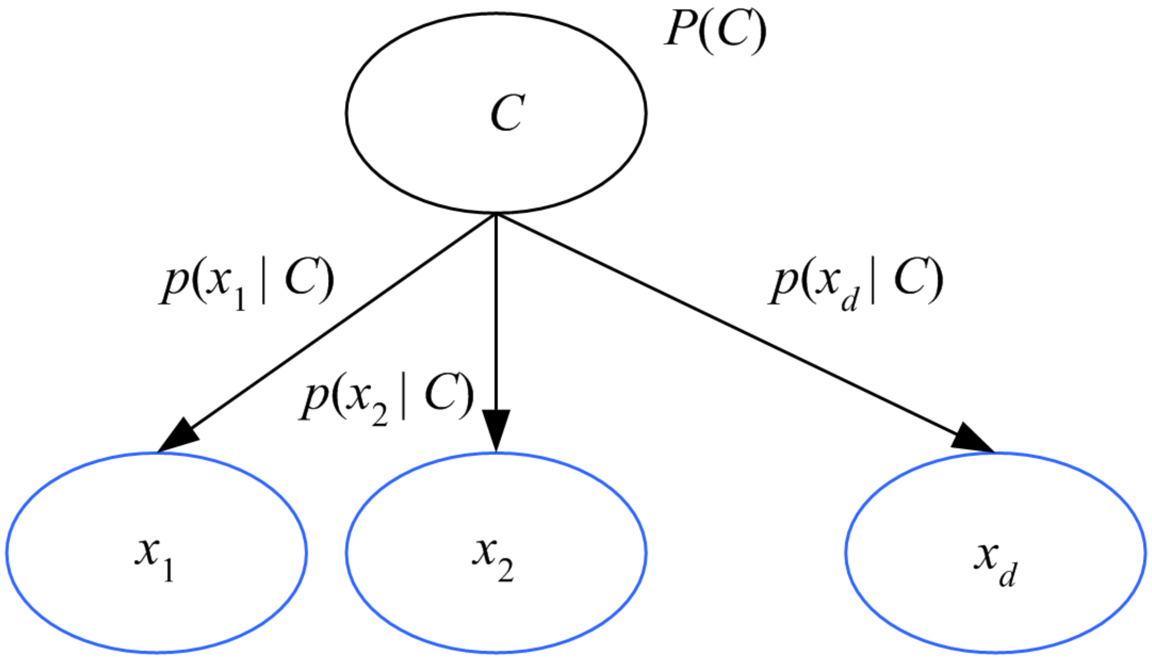
$P(C|x)$



Bayes' rule inverts the arc:

$$P(C|\mathbf{x}) = \frac{p(\mathbf{x}|C)P(C)}{p(\mathbf{x})}$$

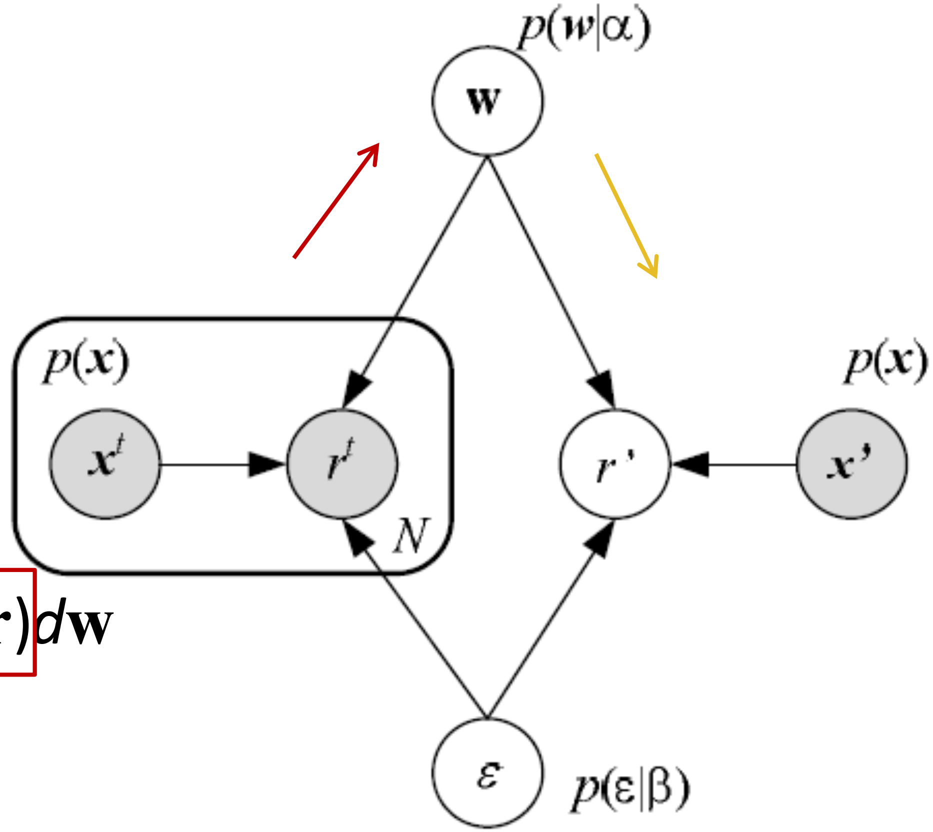
Naive Bayes' Classifier



Given C, x_j are independent:

$$p(\mathbf{x} \mid C) = p(x_1 \mid C)\, p(x_2 \mid C) \ldots p(x_d \mid C)$$

Linear Regression

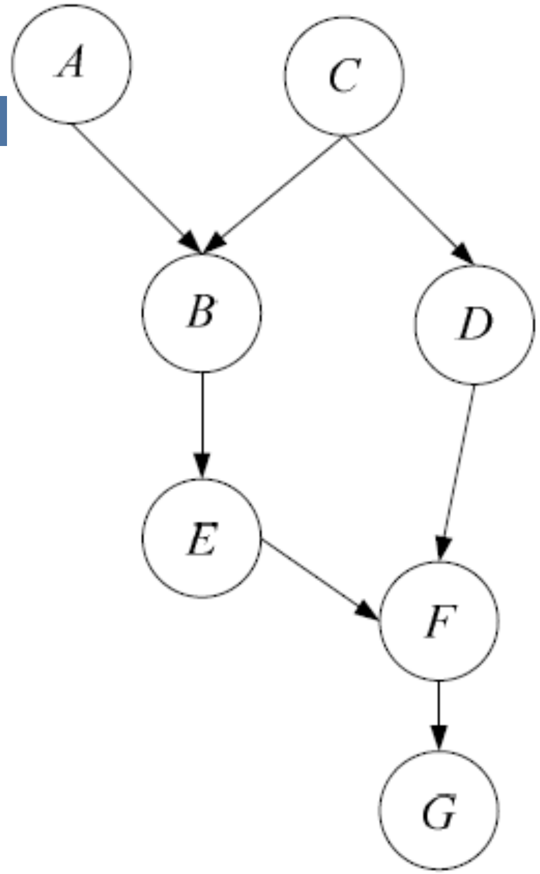


$$p(r'|\mathbf{x}',\mathbf{r},\mathbf{X}) = \int \boxed{p(r'|\mathbf{x}',\mathbf{w})}\,\boxed{p(\mathbf{w}|\mathbf{X},\mathbf{r})}\,d\mathbf{w}$$

$$= \int p(r'|\mathbf{x}',\mathbf{w}) \frac{p(\mathbf{r}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{r})} d\mathbf{w}$$

$$\propto \int p(r'|\mathbf{x}',\mathbf{w}) \prod_t p(r^t|\mathbf{x}^t,\mathbf{w})p(\mathbf{w})d\mathbf{w}$$

d-Separation



- A path from node *A* to node *B* is blocked if
  a) The directions of edges on the path meet head-to-tail (case 1) or tail-to-tail (case 2) and the node is in C, or
  b) The directions of edges meet head-to-head (case 3) and neither that node nor any of its descendants is in C.
- If all paths are blocked, *A* and *B* are d-separated (conditionally independent) given C.

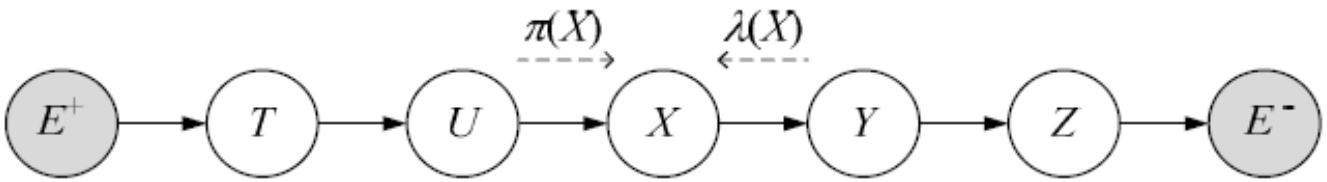
BCDF is blocked given C.

BEFG is blocked by *F*.

BEFD is blocked unless *F* (or G) is given.

Belief Propagation (Pearl, 1988)

- Chain:



$$P(X \mid E) = \frac{P(E \mid X)P(X)}{P(E)} = \frac{P(E^+, E^- \mid X)P(X)}{P(E)}$$

$$= \frac{P(E^+ \mid X)P(E^- \mid X)P(X)}{P(E)} = \alpha\pi(X)\lambda(X)$$

$$\pi(X) = \sum_U P(X \mid U)\pi(U)$$

$$\lambda(X) = \sum_Y P(Y \mid X)\lambda(Y)$$

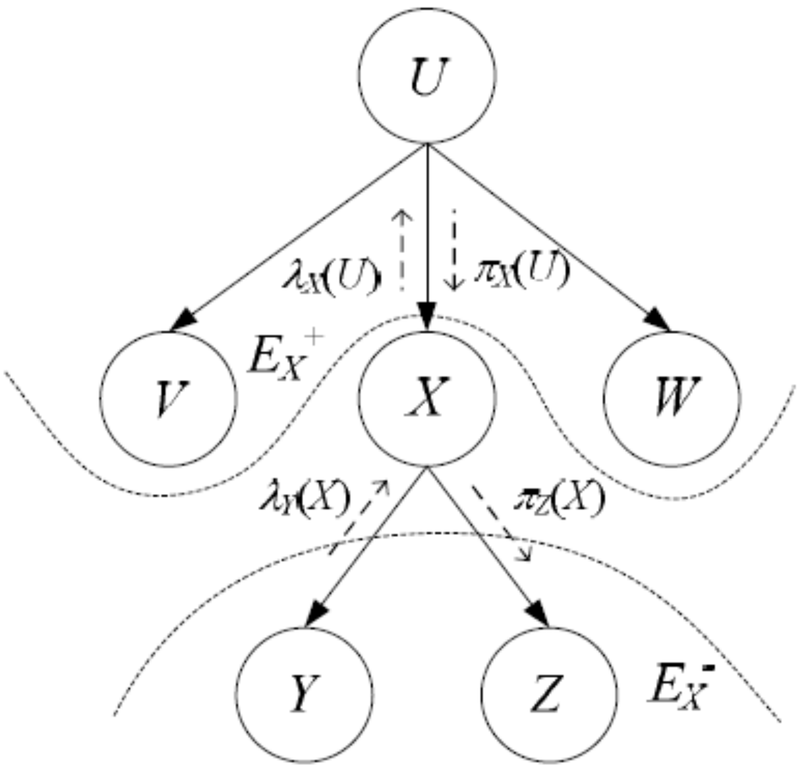
Trees

$$\lambda(X) = P(E_X^- \mid X) = \lambda_Y(X)\lambda_Z(X)$$

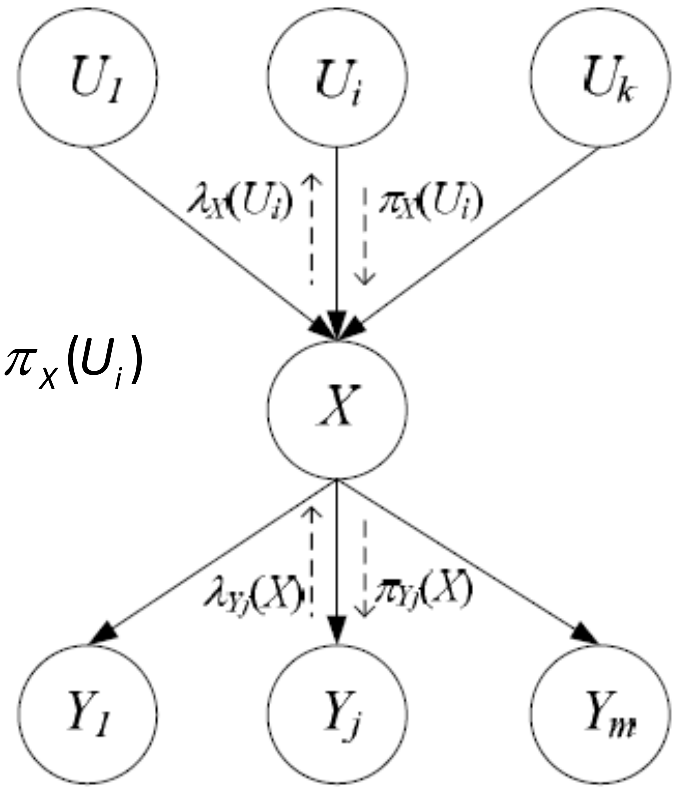
$$\lambda_X(U) = \sum_X \lambda(X)P(X \mid U)$$

$$\pi(X) = P(X \mid E_X^+) = \sum_U P(X \mid U)\pi_X(U)$$

$$\pi_y(X) = \alpha\lambda_Z(X)\pi(X)$$



Polytrees



$$\pi(X) = P(X \mid E_X^+) = \sum_{U_1}\sum_{U_2}\cdots\sum_{U_k} P(X \mid U_1, U_2, \cdots, U_k)\prod_{i=1}^{k}\pi_X(U_i)$$

$$\pi_{y_j}(X) = \alpha\prod_{s\neq j}\lambda_{Y_s}(X)\pi(X)$$

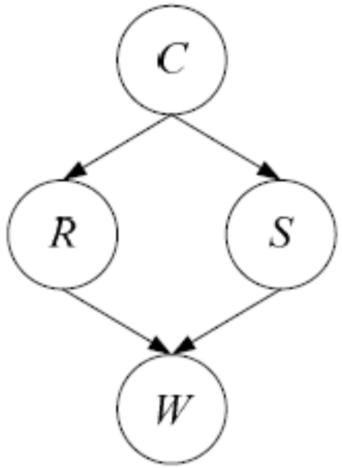
$$\lambda_X(U_i) = \beta\sum_X\lambda(X)\sum_{U_{r\neq i}} P(X \mid U_1, U_2, \cdots, U_k)\prod_{r\neq i}\pi_X(U_r)$$

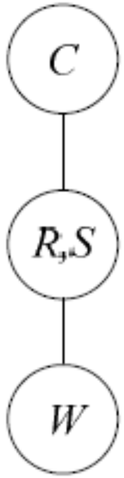
$$\lambda(X) = \prod_{j=1}^{m}\lambda_{Y_j}(X)$$

How can we model $P(X \mid U_1, U_2, \ldots, U_k)$ cheaply?

Junction Trees

- If X does not separate E^+ and E^-, we convert it into a junction tree and then apply the polytree algorithm





Tree of moralized, clique nodes

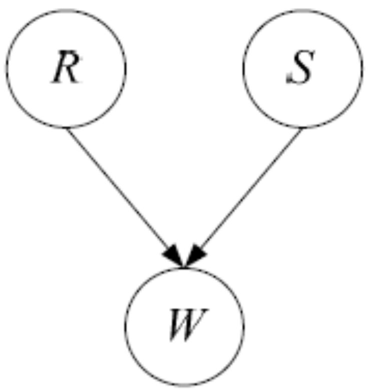
Undirected Graphs: Markov Random Fields

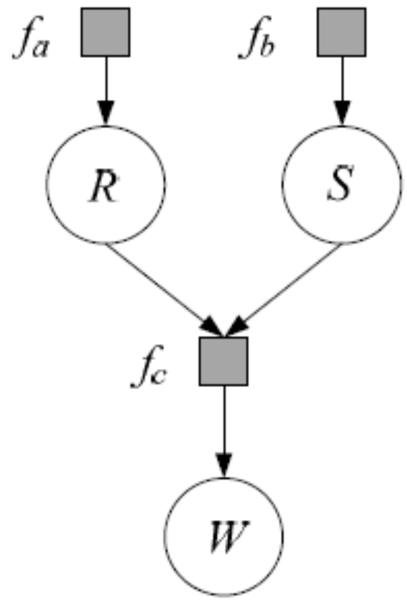
- In a Markov random field, dependencies are symmetric, for example, pixels in an image
- In an undirected graph, *A* and *B* are independent if removing C makes them unconnected.
- Potential function $\psi_c(X_c)$ shows how favorable is the particular configuration *X* over the clique C
- The joint is defined in terms of the clique potentials

$$p(X) = \frac{1}{Z}\prod_{C}\psi_C(X_C) \text{ where normalizer } Z = \sum_{X}\prod_{C}\psi_C(X_C)$$

Factor Graphs

- Define new factor nodes and write the joint in terms of them



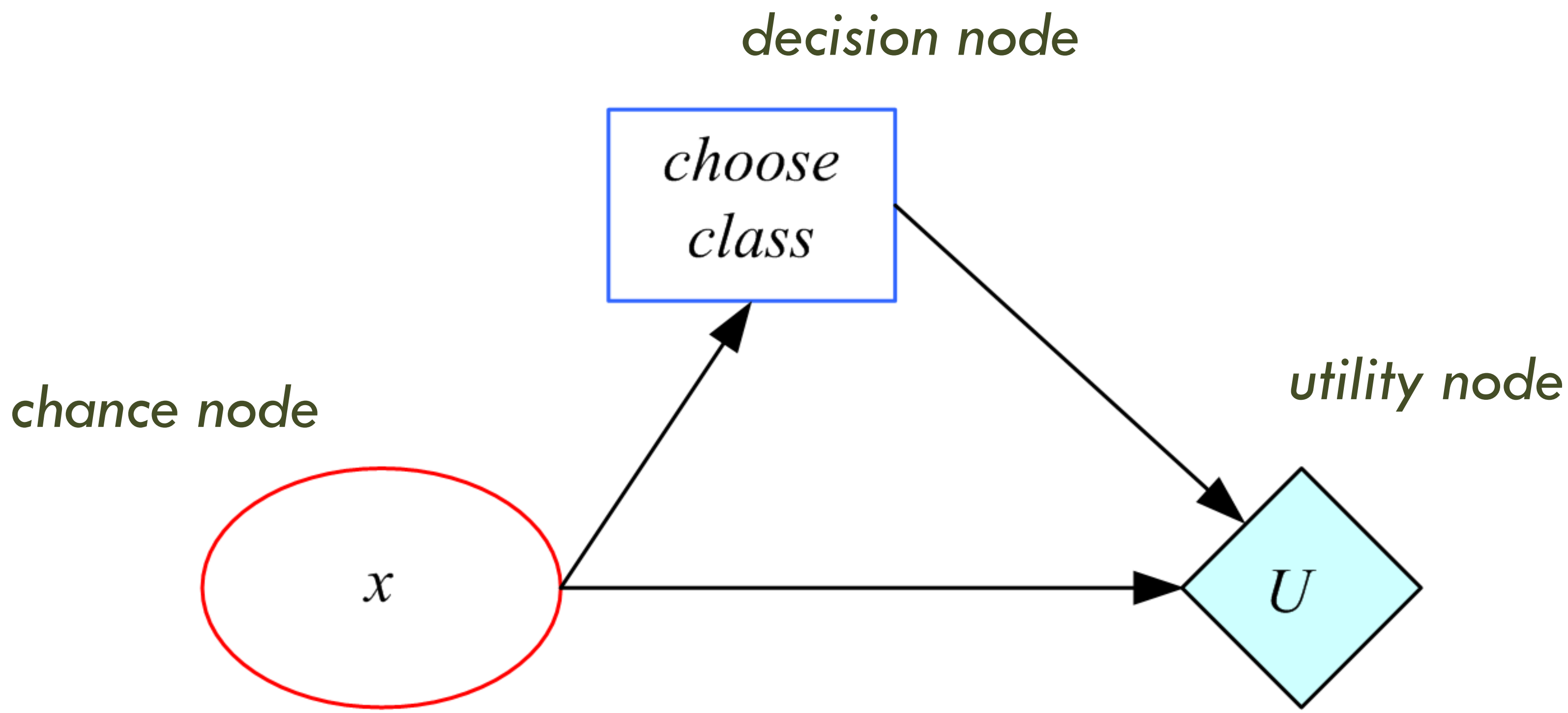


$$p(X) = \frac{1}{Z} \prod_{S} f_S(X_S)$$

Learning a Graphical Model

- Learning the conditional probabilities, either as tables (for discrete case with small number of parents), or as parametric functions
- Learning the structure of the graph: Doing a state-space search over a score function that uses both goodness of fit to data and some measure of complexity

Influence Diagrams



Outline

- Graphical Models
- 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: $S_1, S_2, \ldots, S_N$ State at "time" t, $q_t = S_i$
- First-order Markov

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

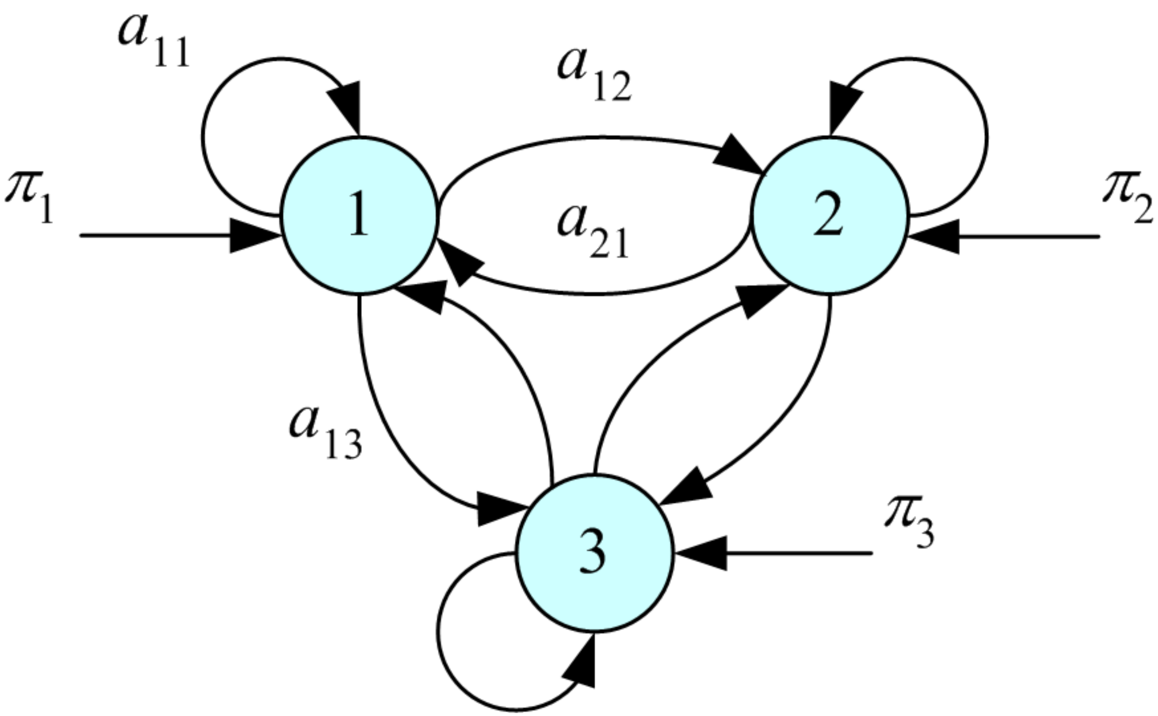
- Transition probabilities

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

- Initial probabilities

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

Stochastic Automaton



Example: Balls and Urns

- Three urns each full of balls of one color

  S_1: red, S_2: blue, S_3: green

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

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

$$P(O \mid \mathbf{A}, \Pi) = P(S_1) \cdot P(S_1 \mid S_1) \cdot P(S_3 \mid S_1) \cdot P(S_3 \mid S_3)$$

$$= \pi_1 \cdot a_{11} \cdot a_{13} \cdot a_{33}$$

$$= 0.5 \cdot 0.4 \cdot 0.3 \cdot 0.8 = 0.048$$

Balls and Urns: Learning

- Given K example sequences of length T

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

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

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

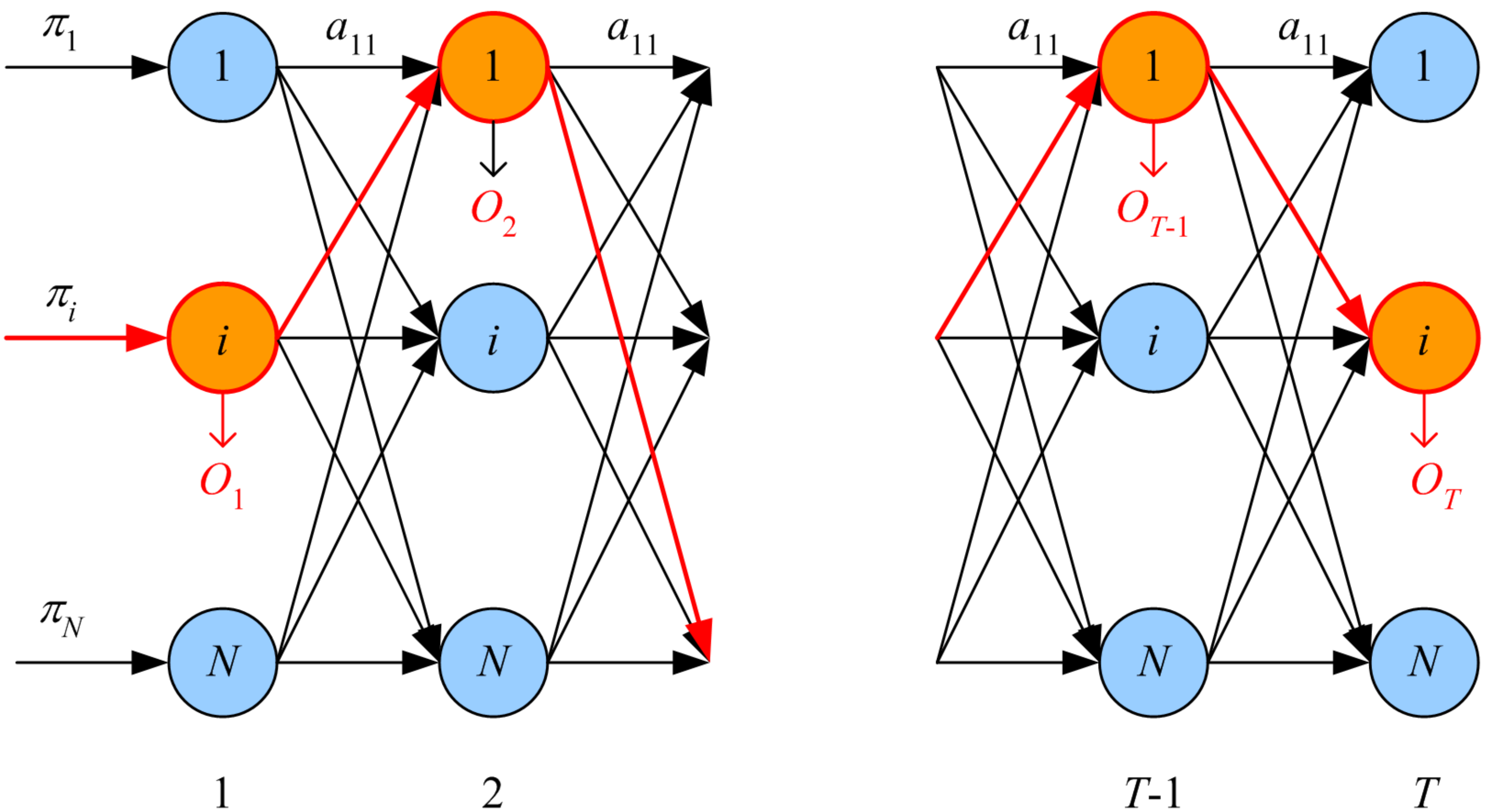
Hidden Markov Models

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

$$b_i(m) \equiv P(O_t = v_m \mid q_t = S_i)$$

- 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
- $\mathbf{B} = b_i(m)$: 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 $P\,(O \mid \lambda)$
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)

Evaluation

- Forward variable:

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

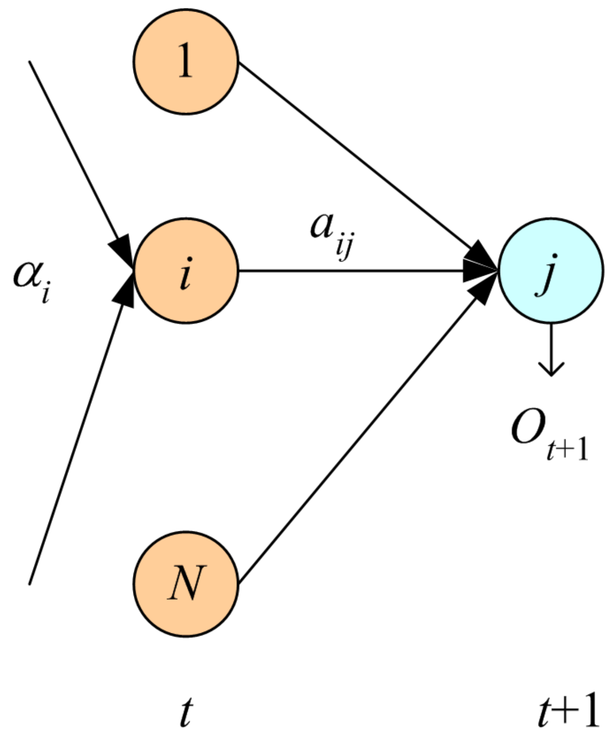
Initialization:

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

Recursion:

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

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



Backward variable:

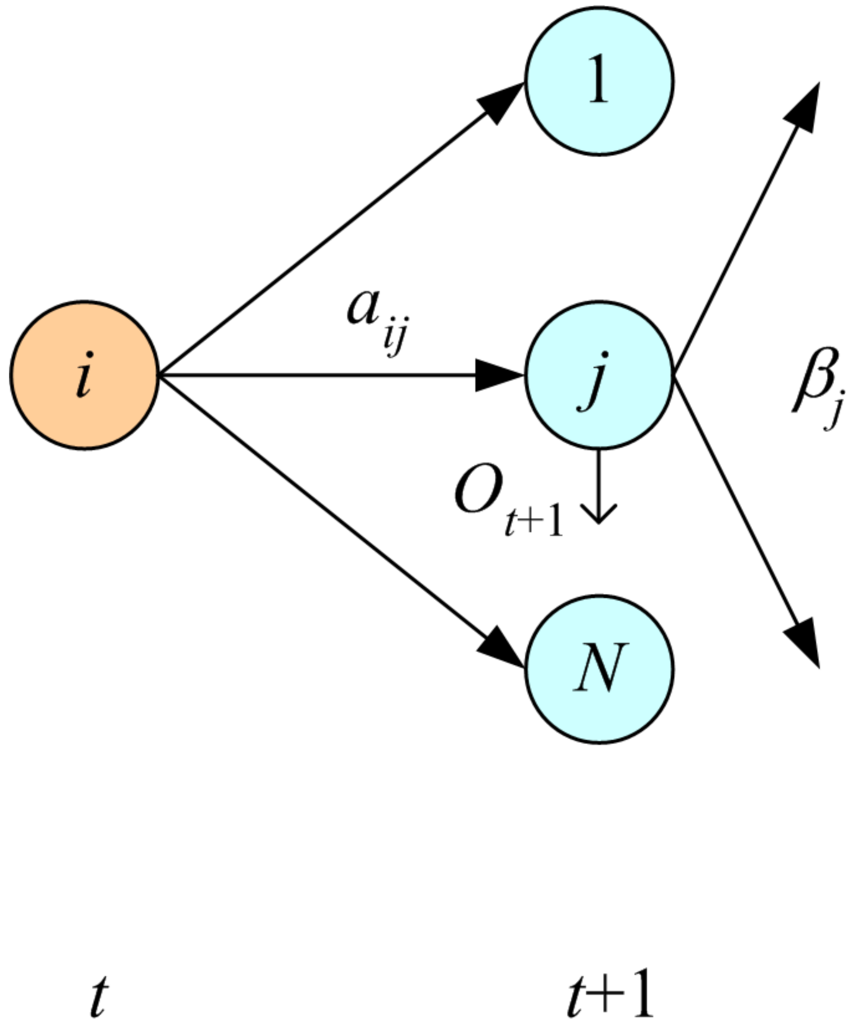
$$\beta_t(i) \equiv P(O_{t+1} \cdots O_T \mid q_t = S_i, \lambda)$$

Initialization:

$$\beta_T(i) = 1$$

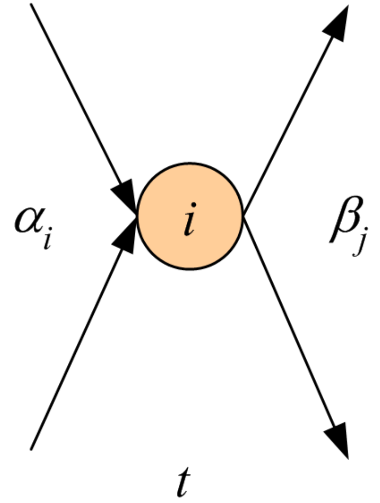
Recursion:

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



Finding the State Sequence

$$\gamma_t(i) \equiv P(q_t = S_i | O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$



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

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

No!

Viterbi's Algorithm

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

- Initialization:

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

- Recursion:

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

- Termination:

  $p^* = \max_i \delta_T(i)$, $q_T^* = \operatorname{argmax}_i \delta_T(i)$

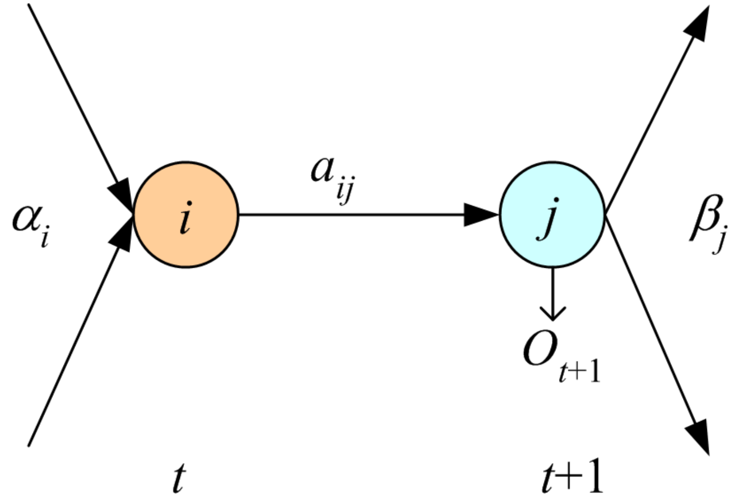
- Path backtracking:

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

Learning

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

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



Baum - Welch algorithm (EM):

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

Baum-Welch (EM)

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

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

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

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

Continuous Observations

- Discrete:

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

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

$$P(O_t \mid q_t = S_j, \lambda) = \sum_{l=1}^{L} P(\mathcal{G}_{jl}) p(O_t \mid q_t = S_j, \mathcal{G}_l, \lambda)$$

$$\sim \mathcal{N}(\mu_l, \Sigma_l)$$

- Continuous:

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

Use EM to learn parameters, e.g.,

$$\hat{\mu}_j = \frac{\sum_t \gamma_t(j) O_t}{\sum_t \gamma_t(j)}$$

HMM with Input

- Input-dependent observations:

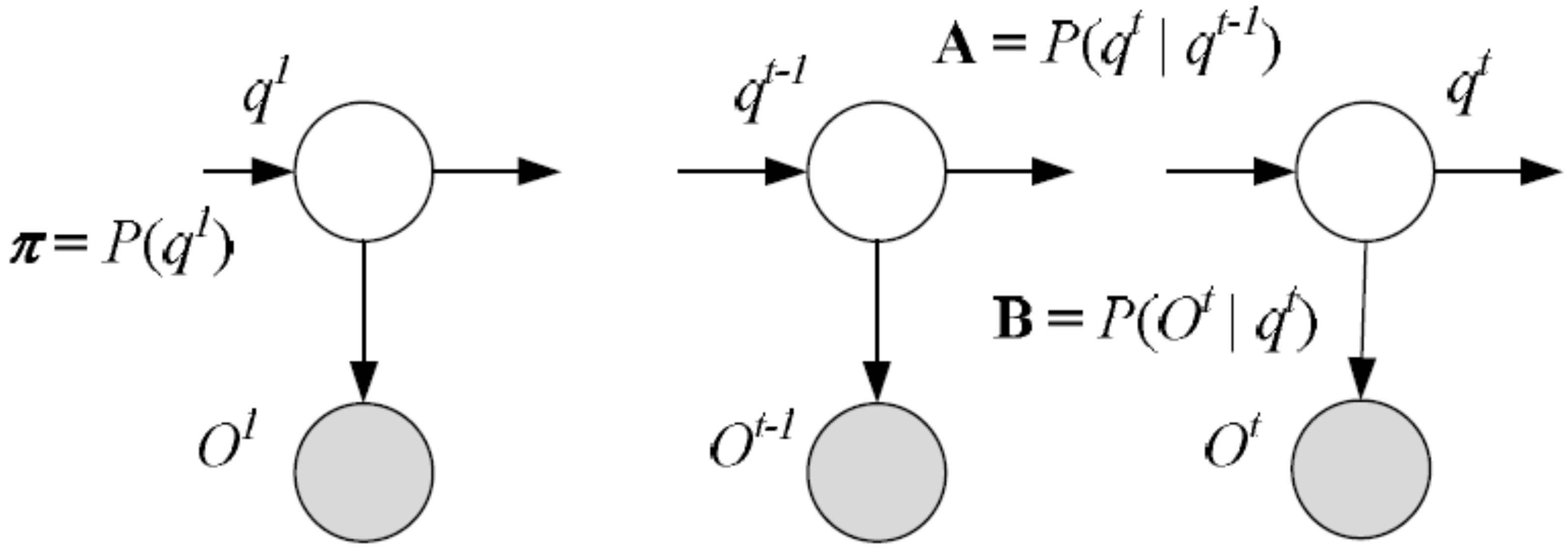
$$P(O_t \mid q_t = S_j, x^t, \lambda) \sim \mathcal{N}\left(g_j(x^t \mid \theta_j), \sigma_j^2\right)$$

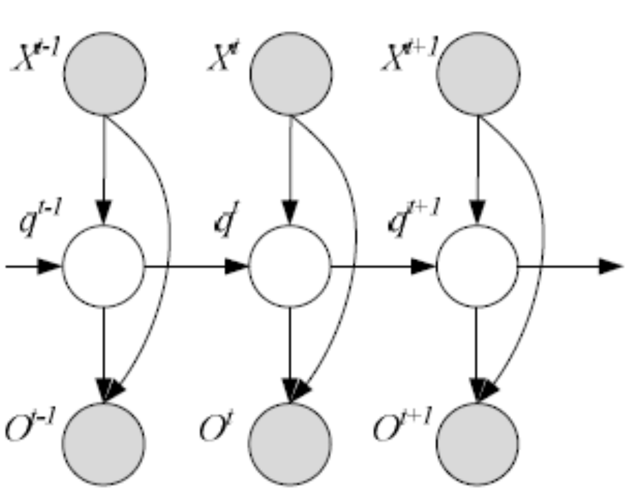
- Input-dependent transitions (Meila and Jordan, 1996; Bengio and Frasconi, 1996):

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

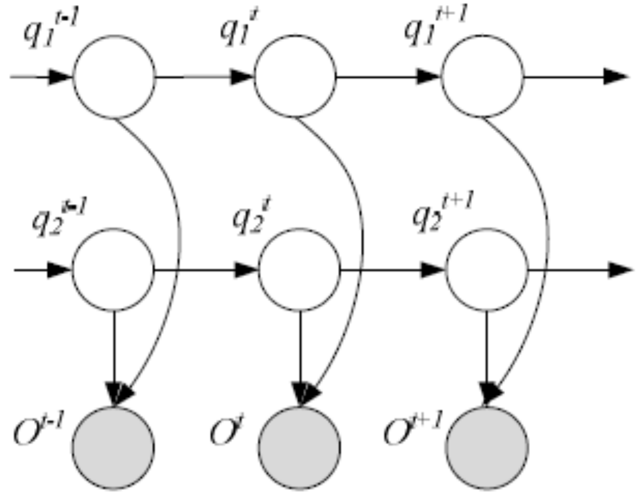
- Time-delay input: $\mathbf{x}^t = \mathbf{f}(O_{t-\tau}, \dots, O_{t-1})$

HMM as a Graphical Model

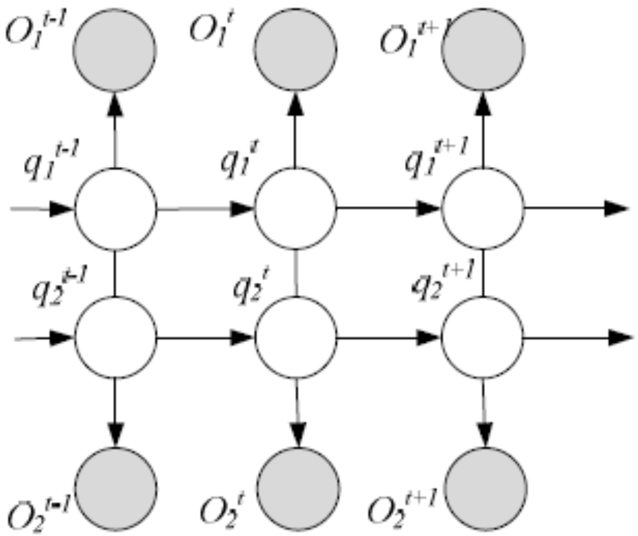




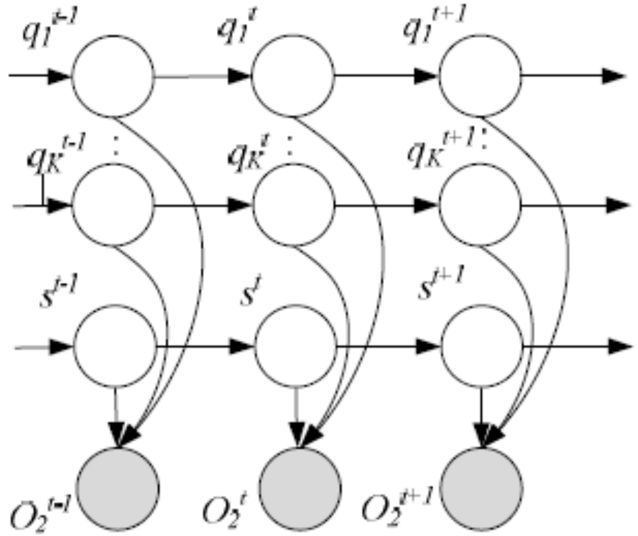
(a) Input-output HMM



(b) Factorial HMM



(c) Coupled HMM

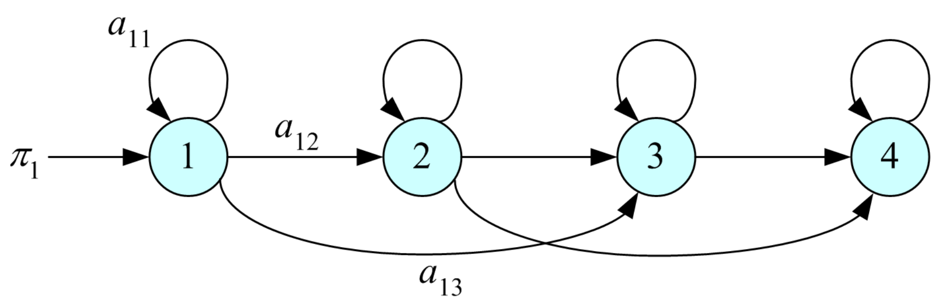


(d) Switching HMM

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

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

References

- Pattern Recognition and Machine Learning. 2006, *M.Bishop*.
- Statistical Pattern Recognition. 3rd, 2011.
- Introduction to Machine Learning. 3rd, 2014.