Machine Learning for Data Science (CS4786) Lecture 21

Graphical Models

Course Webpage :

http://www.cs.cornell.edu/Courses/cs4786/2016sp/

Bayesian Networks

- Directed acyclic graph (DAG): $G = (V, E)$
- Joint distribution P_θ over $X_1, \ldots, X_n$ that factorizes over G:

$$P_\theta(X_1, \ldots, X_n) = \prod_{i=1}^{N} P_\theta(X_i | \text{Parent}(X_i))$$

- Hence Bayesian Networks are specified by G along with CPD's over the variables (given their parents)

Variable Elimination: Bayesian Network

Initialize List with conditional probability distributions

Pick an order of elimination I for remaining variables

For each $X_i \in I$

Find distributions in List containing variable X_i and remove them

Define new distribution as the sum (over values of X_i) of the product of these distributions

Place the new distribution on List

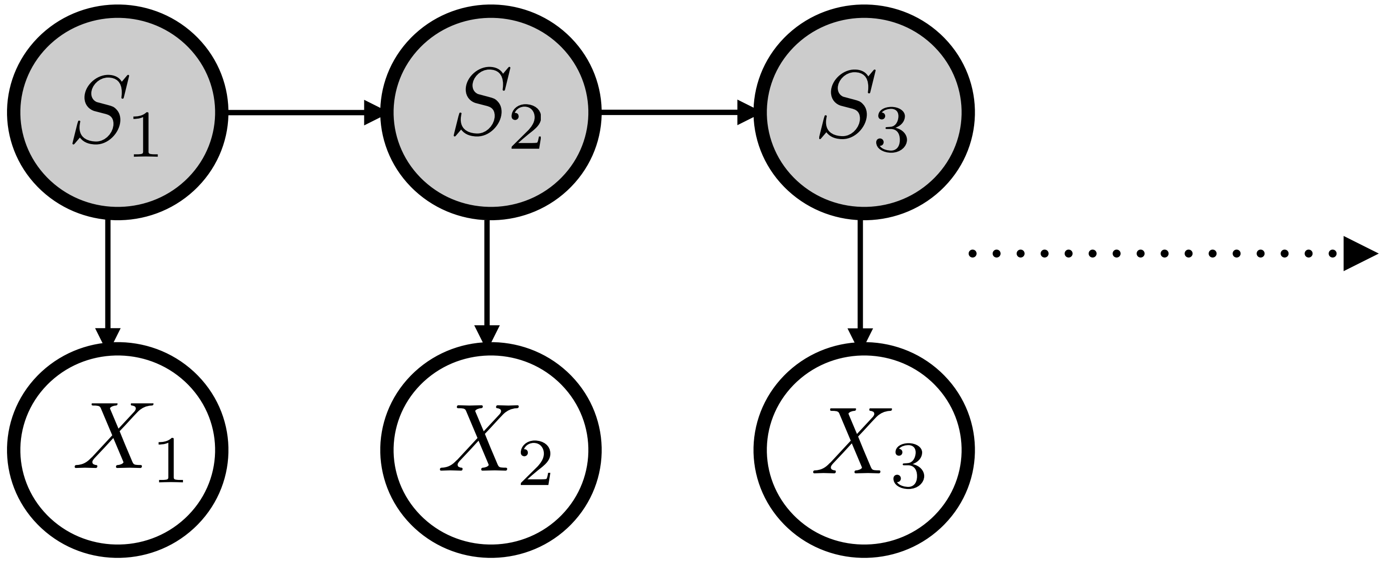
End

Return List

MESSAGE PASSING

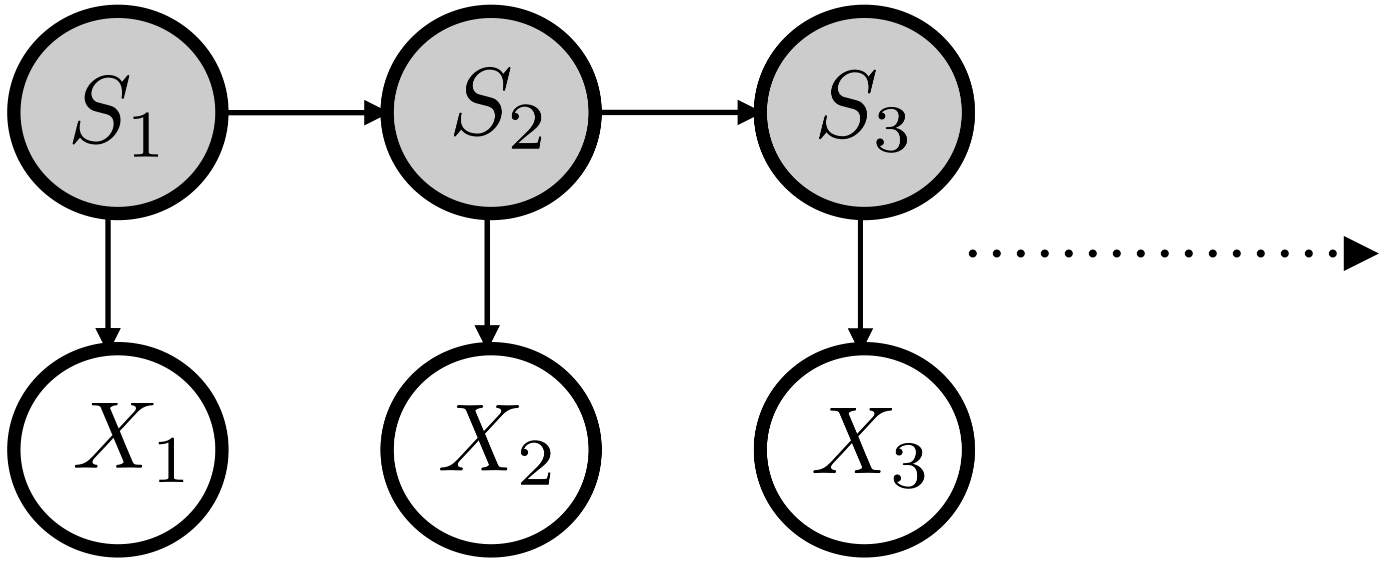
- Often we need more than one marginal computation
- Over variables we need marginals for, there are many common distributions/potentials in the list
- Can we exploit structure and compute these intermediate terms that can be reused?

Example: Hidden Markov Model



$$P(S_1, \ldots, S_n | X_1, \ldots, X_n) = \frac{P(S_1, \ldots, S_n, X_1, \ldots, X_n)}{P(X_1, \ldots, X_n)}$$

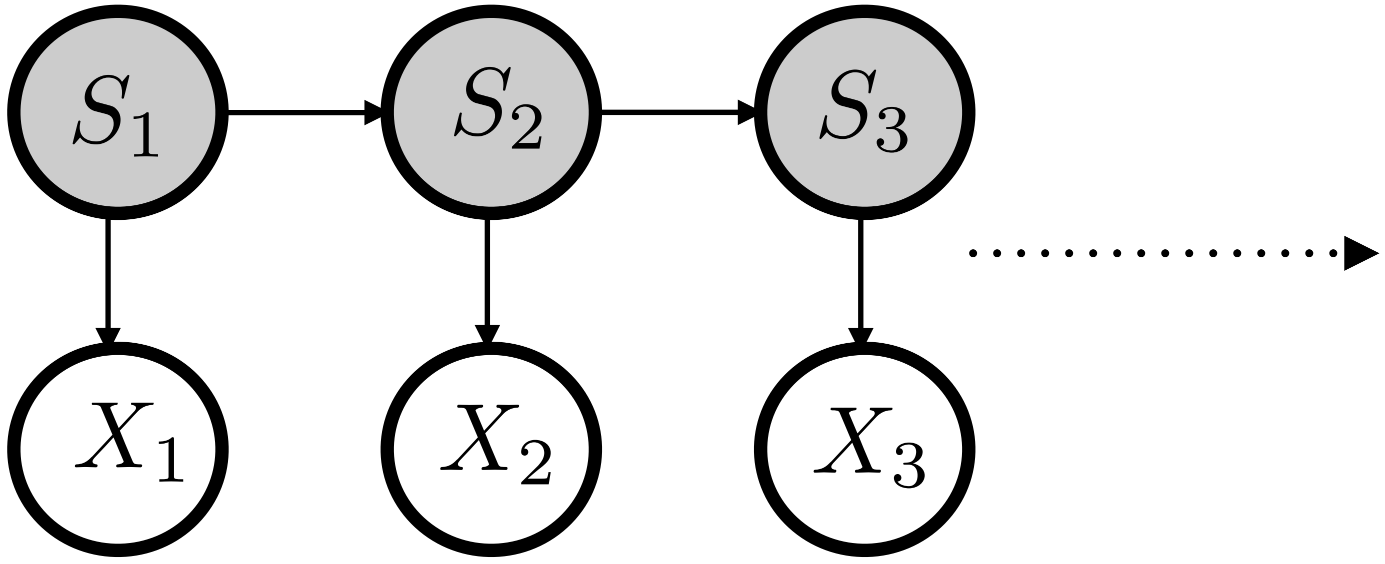
Example: Hidden Markov Model



$$\text{message}_{S_{t-1} \mapsto S_t}(S_{t-1}) = P(X_1, \ldots, X_{t-1}, S_{t-1})$$

$$\text{message}_{S_{t+1} \mapsto S_t}(S_t) = P(X_n, \ldots, X_{t+1} | S_t)$$

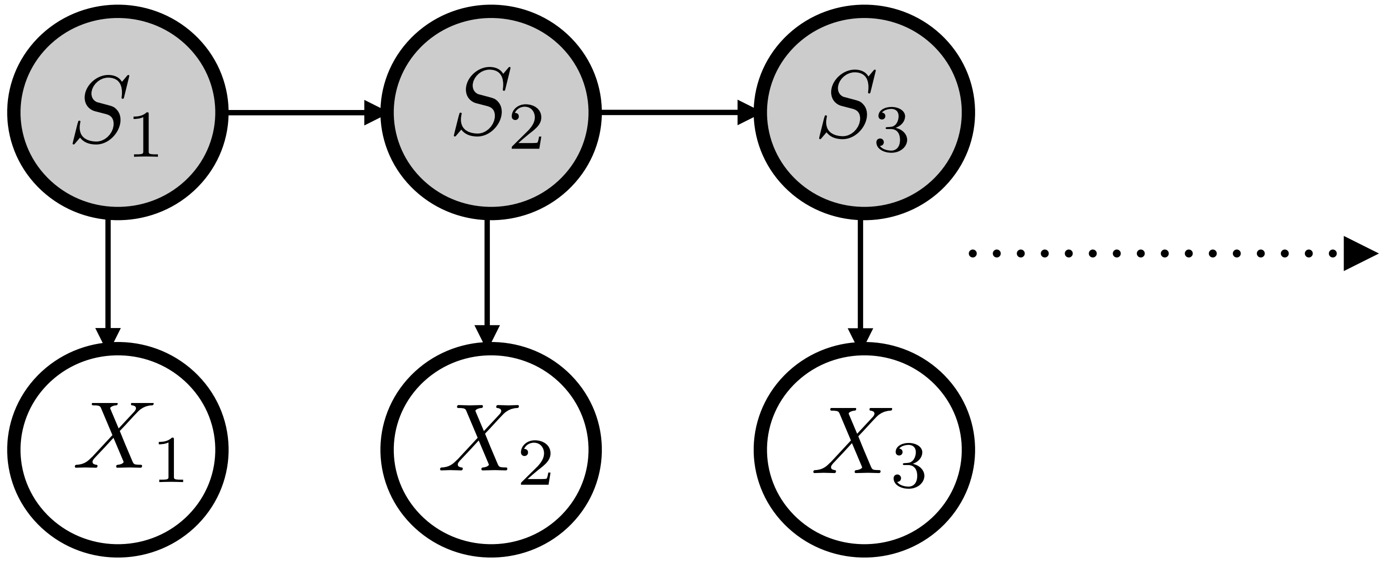
Example: Hidden Markov Model



$$\text{message}_{S_{t-1} \mapsto S_t}(S_{t-1}) = P(X_1, \ldots, X_{t-1}, S_{t-1})$$

$$\text{message}_{S_{t+1} \mapsto S_t}(S_t) = P(X_n, \ldots, X_{t+1} | S_t)$$

Example: Hidden Markov Model

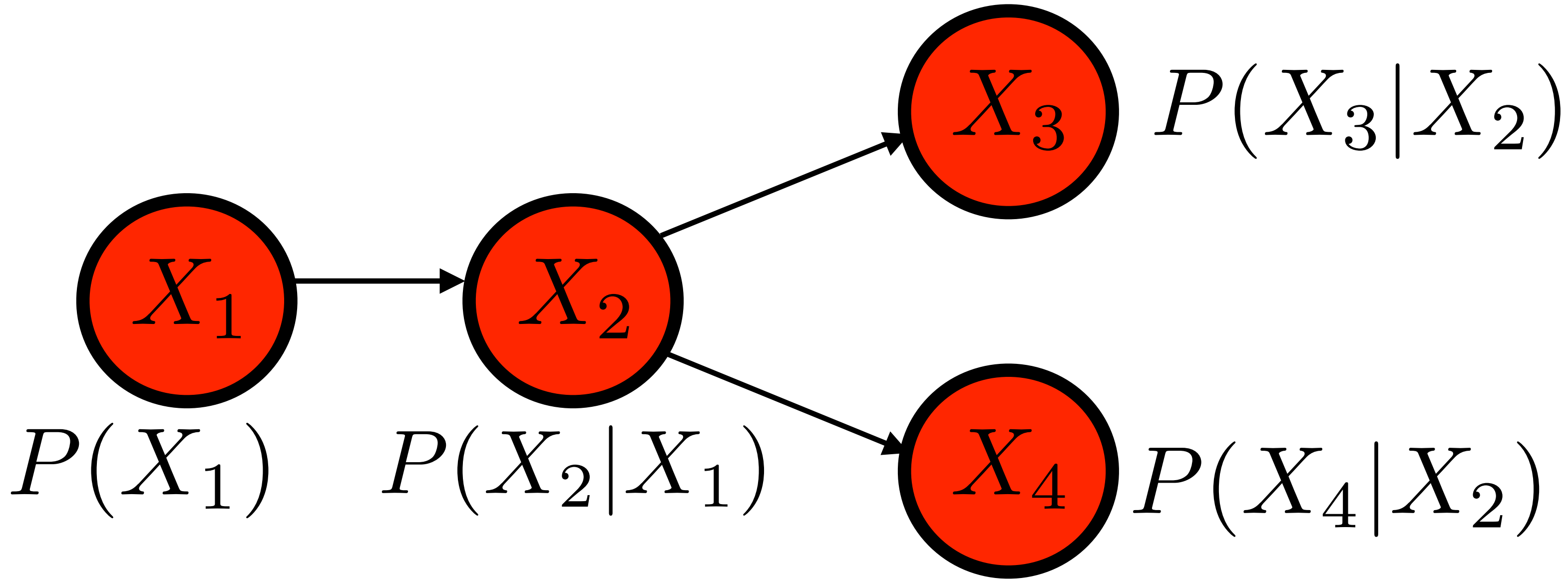


$$\text{message}_{S_{t-1} \mapsto S_t}(S_{t-1}) = P(X_1, \ldots, X_{t-1}, S_{t-1})$$
$$\text{message}_{S_{t+1} \mapsto S_t}(S_t) = P(X_n, \ldots, X_{t+1} | S_t)$$

Belief Propagation

- Think of variables as nodes in a network, each node is allowed to chat with its neighbors
- Adjacent nodes receive messages from neighbors telling the node how to update its belief
- Each node in turn sends messages to its neighbors:
  based on observation, previous received messages, marginal and conditional distributions telling the other how to update beliefs
- (Hopefully) All the nodes converge on their beliefs

Message Passing Example



Belief Propagation

1. For every observation $X_j = x_j$ define $E_{X_j}(x) = \mathbf{1}\left\{x = x_j\right\}$, for unobserved variables set $E_{X_j}(x) = 1$
2. At round 0, all messages between nodes are 1

Belief Propagation

For any node X_i

- Incoming message to node from children:

$$\lambda(x) = E_{X_i}(x) \prod_{j \in \text{children}(X_i)} \lambda_{X_j}(x)$$

- Incoming message from Parents:

$$\pi(x) = \sum_u P(X_i = x | \text{Parent}(X_i) = u) \prod_{k \in \text{Parent}(X_i)} \pi_{X_i}(u_k)$$

- Outgoing message to Parent X_j:

$$\lambda_{X_i}(u_i) \propto \sum_x \lambda(x) \sum_{u \setminus u_i} P(X_i = x | \text{Parent}(X_i) = u) \prod_{k \neq i} \pi_{X_i}(u_k)$$

- Outgoing message to child X_j:

$$\pi_{X_j}(x) \propto \pi(x) E_{X_i}(x) \prod_{k \neq j} \lambda_{X_j}(x)$$