Machine Learning for Data Science (CS4786)
Lecture 22

Graphical Models

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

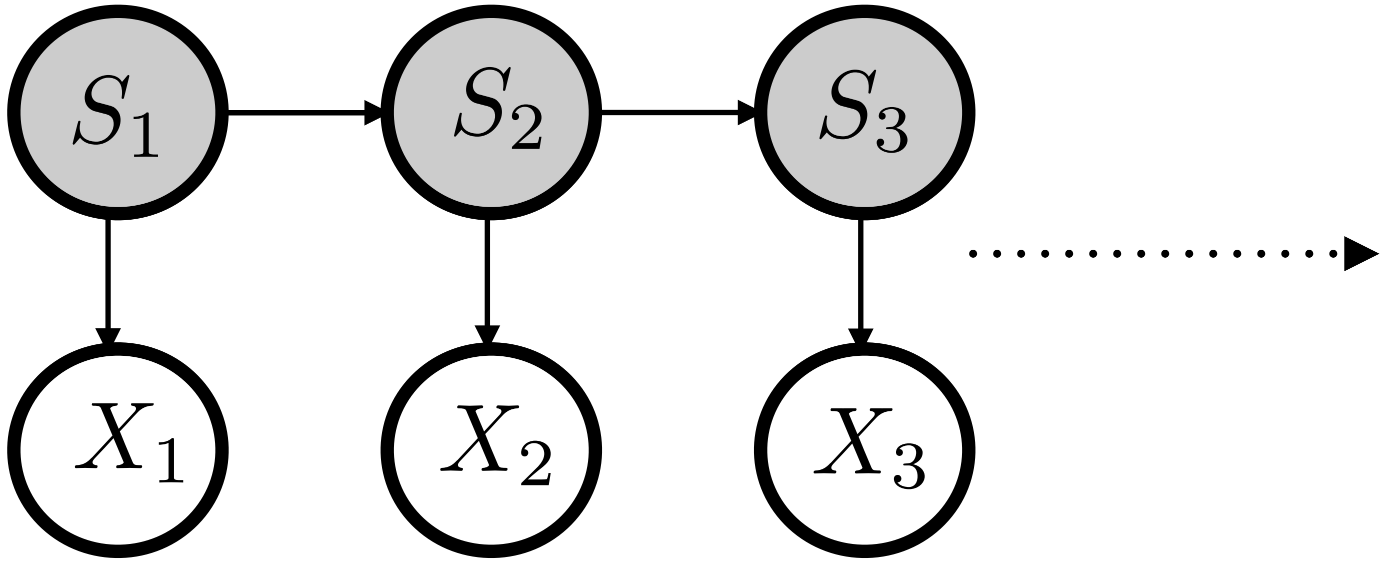
Bayesian Networks

- 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)

Example: Hidden Markov Model



$$\text{message}_{1\mapsto 2}(S_1) = P(X_1, S_1) = P(X_1|S_1)P(S_1)$$

$$\text{message}_{n+1\mapsto n}(S_n) = (1, \ldots, 1)$$

$$\text{message}_{S_t\mapsto S_{t+1}}(S_t) = P(X_t|S_t)\left(\sum_{S_{t-1}} \text{message}_{S_{t-1}\mapsto S_t}(S_{t-1}) \cdot P(S_t|S_{t-1})\right)$$

$$\text{message}_{S_t\mapsto S_{t-1}}(S_{t-1}) = \sum_{S_{t+1}} \text{message}_{S_{t+1}\cdot\mapsto S_t}(S_t) \cdot P(X_t|S_t)P(S_t|S_{t-1})$$

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

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

Belief Propagation

Message to Parent X_j

Message to Parent X_j

$$\lambda_{X_i}(u_j) \propto \sum_{x} \sum_{u \setminus u_j} E_{X_i}(x) P(X_i = x | \mathrm{Parent}(X_i) = u) \left(\prod_{k \in \mathrm{children}} \lambda_{X_k}(x) \prod_{k \in \mathrm{Parent}(X_i), k \neq j} \pi_{X_i}(u_k) \right)$$

Message to Parent X_j

$$\lambda_{X_i}(u_j) \propto \sum_{x} \sum_{u \setminus u_j} E_{X_i}(x) P(X_i = x | \text{Parent}(X_i) = u) \left(\prod_{k \in \text{children}} \lambda_{X_k}(x) \prod_{k \in \text{Parent}(X_i), k \neq j} \pi_{X_i}(u_k) \right)$$

$$\sum_{x, \text{all parents but } \mathrm{X_j}} E_{X_i}(x) P(X_i = x | \text{Parent}(X_i) = u)(\text{product of all messages but one from } \mathrm{X_j})$$

Belief Propagation

Message to Parent X_j

$$\lambda_{X_i}(u_j) \propto \sum_x \sum_{u \setminus u_j} E_{X_i}(x) P(X_i = x | \text{Parent}(X_i) = u) \left(\prod_{k \in \text{children}} \lambda_{X_k}(x) \prod_{k \in \text{Parent}(X_i), k \neq j} \pi_{X_i}(u_k) \right)$$

$$\sum_{x, \text{all parents but } \mathrm{X_j}} E_{X_i}(x) P(X_i = x | \text{Parent}(X_i) = u)(\text{product of all messages but one from } \mathrm{X_j})$$

Message to child X_j

Belief Propagation

Message to Parent X_j

$$\lambda_{X_i}(u_j) \propto \sum_{x} \sum_{u \setminus u_j} E_{X_i}(x) P(X_i = x | \text{Parent}(X_i) = u) \left(\prod_{k \in \text{children}} \lambda_{X_k}(x) \prod_{k \in \text{Parent}(X_i), k \neq j} \pi_{X_i}(u_k) \right)$$

$$\sum_{x, \text{all parents but } \mathrm{X_j}} E_{X_i}(x) P(X_i = x | \text{Parent}(X_i) = u)(\text{product of all messages but one from } \mathrm{X_j})$$

Message to child X_j

$$\pi_{X_i}(x) \propto E_{X_i}(x) \sum_{u} P(X_i = x | \text{Parent}(X_i) = u) \left(\prod_{k \in \text{Parent}(X_i)} \pi_{X_i}(u_k) \prod_{k \in \text{children}, k \neq j} \lambda_{X_k}(x) \right)$$

BELIEF PROPAGATION

Message to Parent X_j

$$\lambda_{X_i}(u_j) \propto \sum_{x} \sum_{u \backslash u_j} E_{X_i}(x) P(X_i = x | \text{Parent}(X_i) = u) \left(\prod_{k \in \text{children}} \lambda_{X_k}(x) \prod_{k \in \text{Parent}(X_i), k \neq j} \pi_{X_i}(u_k) \right)$$

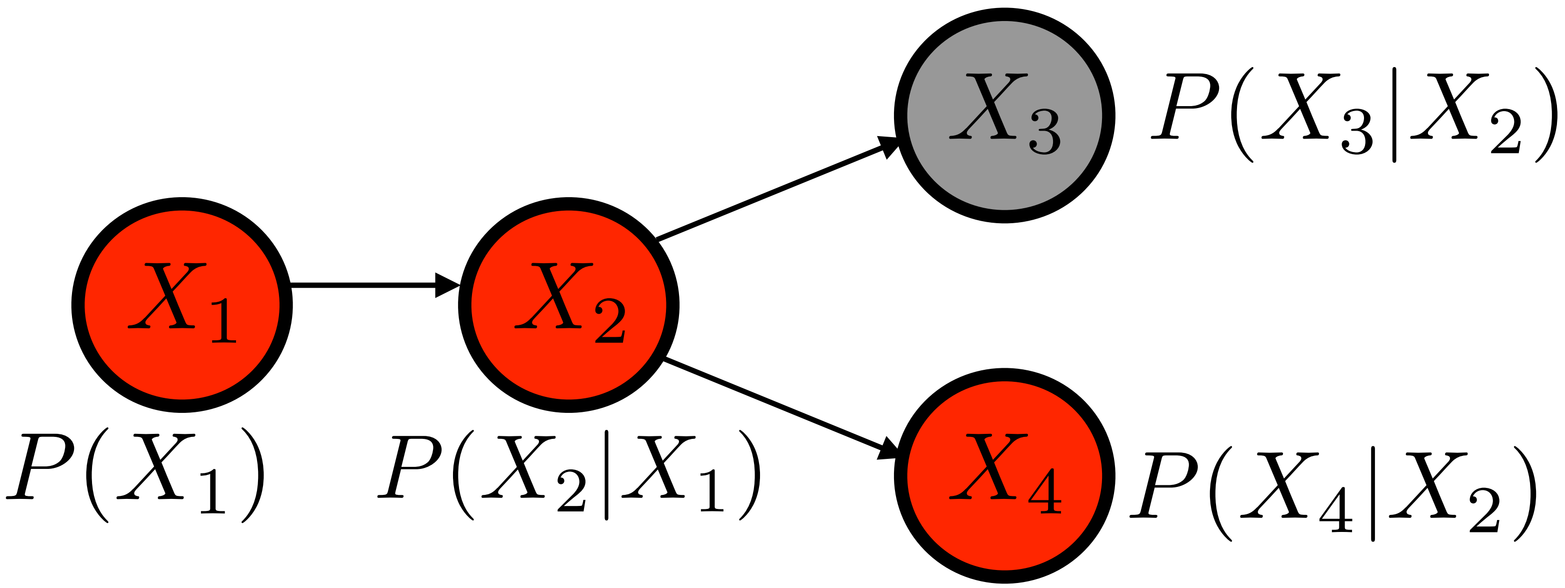
$$\sum_{x,\text{all parents but } \mathrm{X_j}} E_{X_i}(x) P(X_i = x | \text{Parent}(X_i) = u)(\text{product of all messages but one from } \mathrm{X_j})$$

Message to child X_j

$$\pi_{X_i}(x) \propto E_{X_i}(x) \sum_{u} P(X_i = x | \text{Parent}(X_i) = u) \left(\prod_{k \in \text{Parent}(X_i)} \pi_{X_i}(u_k) \prod_{k \in \text{children}, k \neq j} \lambda_{X_k}(x) \right)$$

$$\sum_{\text{all parents}} E_{X_i}(x) P(X_i = x | \text{Parent}(X_i) = u)(\text{product of all messages but one from } \mathrm{X_j})$$

Message Passing Example



Message to Parent Xj

$$\sum_{x,\text{all parents but } \mathrm{X_j}} E_{X_i}(x) P(X_i = x|\text{Parent}(X_i) = u)(\text{product of all messages but one from } \mathrm{X_j})$$

Message to child Xj

$$\sum_{\text{all parents}} E_{X_i}(x) P(X_i = x|\text{Parent}(X_i) = u)(\text{product of all messages but one from } \mathrm{X_j})$$

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:

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

Parameter Estimation (Learning)

- What are the parameters for a Baysian Network?

Parameter Estimation (Learning)

- What are the parameters for a Baysian Network?
  - The conditional probability distributions/tables/density functions

Parameter Estimation (Learning)

- MLE: n independent samples $(X_1^1, \ldots, X_N^1), \ldots, (X_1^n, \ldots, X_N^n)$ where each $(X_1^t, \ldots, X_N^t)$ is drawn from the Bayesian network

Parameter Estimation (Learning)

- MLE: n independent samples $(X_1^1, \ldots, X_N^1), \ldots, (X_1^n, \ldots, X_N^n)$ where each $(X_1^t, \ldots, X_N^t)$ is drawn from the Bayesian network

$$\arg\max_{\theta} \sum_{t=1}^{n} \log(P_\theta(X_1^t, \ldots, X_N^t))$$

$$= \arg\max_{\theta} \sum_{t=1}^{n} \sum_{i=1}^{N} \log(P_\theta(X_i^t | \text{Parent}(X_i^t)))$$

If θ_i is the parameter only involving $P_\theta(X_i^t | \text{Parent}(X_i^t))$ then

$$\theta_i^{MLE} = \arg\max_{\theta_i} \sum_{t=1}^{n} \log(P_{\theta_i}(X_i^t | \text{Parent}(X_i^t)))$$

Parameter Estimation (Learning)

- Simple case of finite outcomes

$$\theta_i^{MLE} = \text{empirical conditional probability table}$$

Parameter Estimation: Latent Variables

- EM Algorithm: Initialize parameters randomly
- For $j = 1$ to convergence
  - E-step: For each of the Latent variable X_i, perform inference to compute

$$Q^{(j)}(\text{Latent variables}) = P_{\theta^{(j-1)}}(\text{Latent variables}|\text{Observation})$$

  - M-step:

$$\theta^{(j)} = \arg\max_{\theta} \sum_{\text{Latent variables}} Q^{(j)}(\text{Latent variables}) \sum_{t=1}^{n} \log P_{\theta}(X_1^t, \ldots, X_N^t)$$

which can be simplified to:

$$\theta_i^{(j)} = \arg\max_{\theta_i} \sum_{\text{Latent}} Q^{(j)}(\text{Latent}) \sum_{t=1}^{n} \log P_{\theta_i}(X_i^t|\text{Parent}(X_i^t))$$

Parameter Estimation: Latent Variables

M-step for simple case of finite outcomes

$\theta_i^{(j)}$ = empirical conditional probability table weighted by $Q^{(j)}$

For HMM this is called the Baum Welch algorithm