Machine Learning for Data Science (CS4786)
Lecture 20

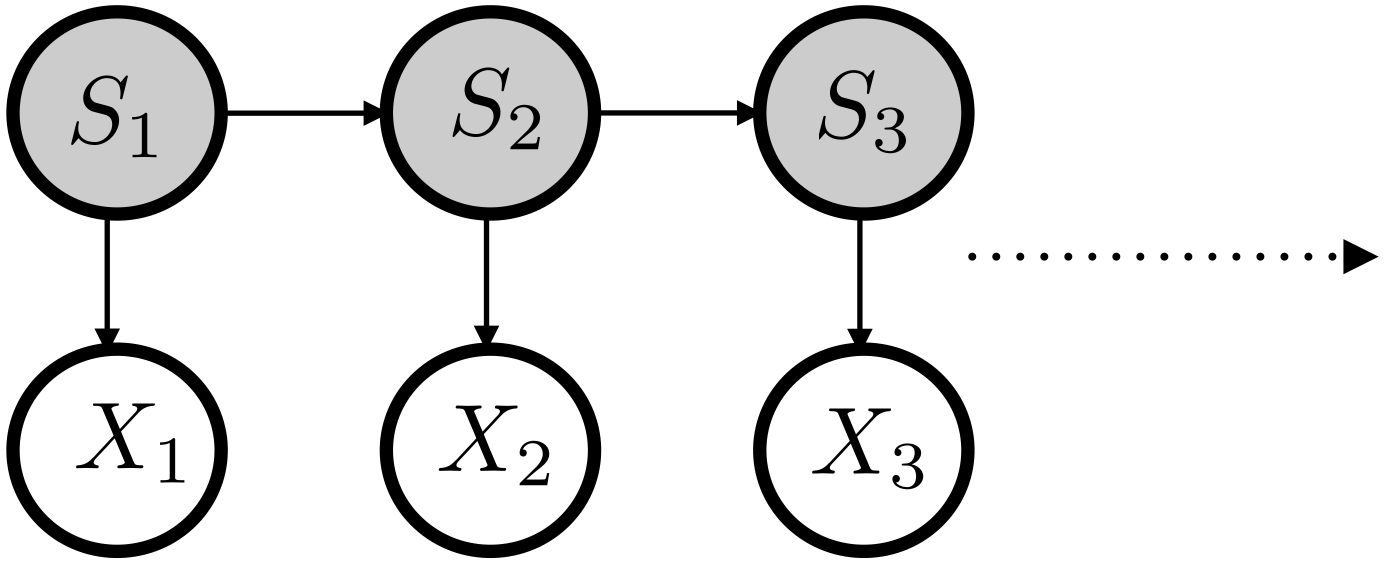
Finish HMM, Inference in Graphical Models

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

Announcement

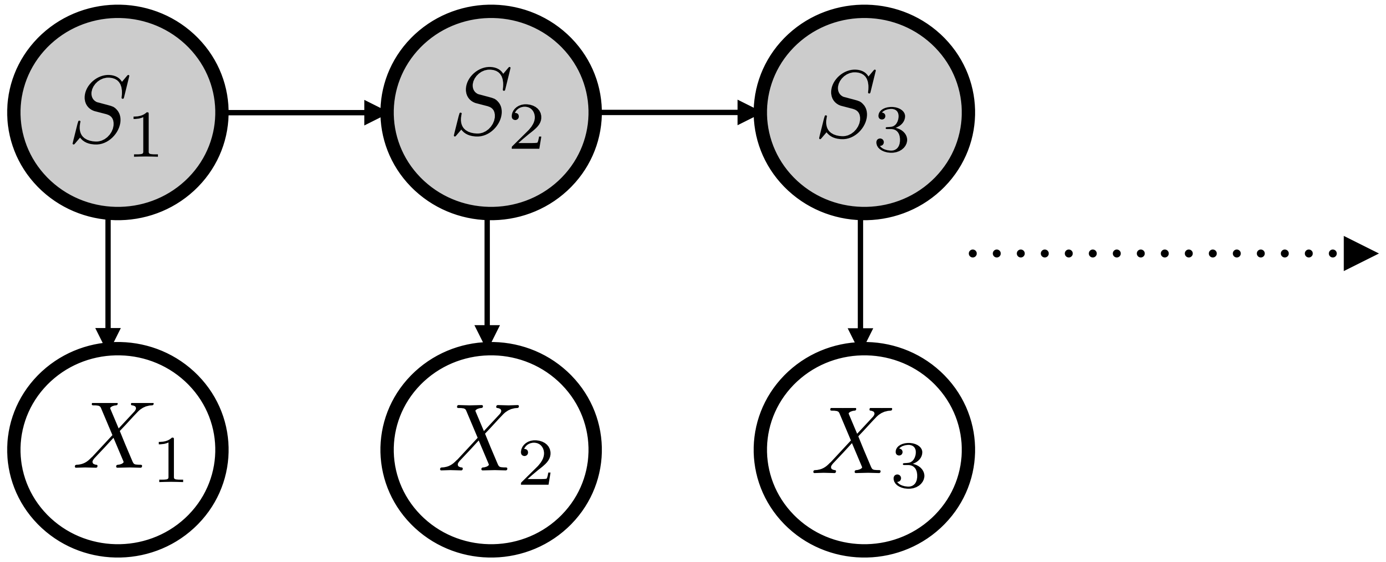
- Good job on competition I, report due today
- No lecture Tuesday, Nov 8th!
- Next Thursday Nov 10th, guest lecture by Prof. Kilian Weinberger on TSNE

HIDDEN MARKOV MODEL (HMM)



Parameters: Transition probability matrix T
Emission probability matrix E

INFERENCE IN HMM

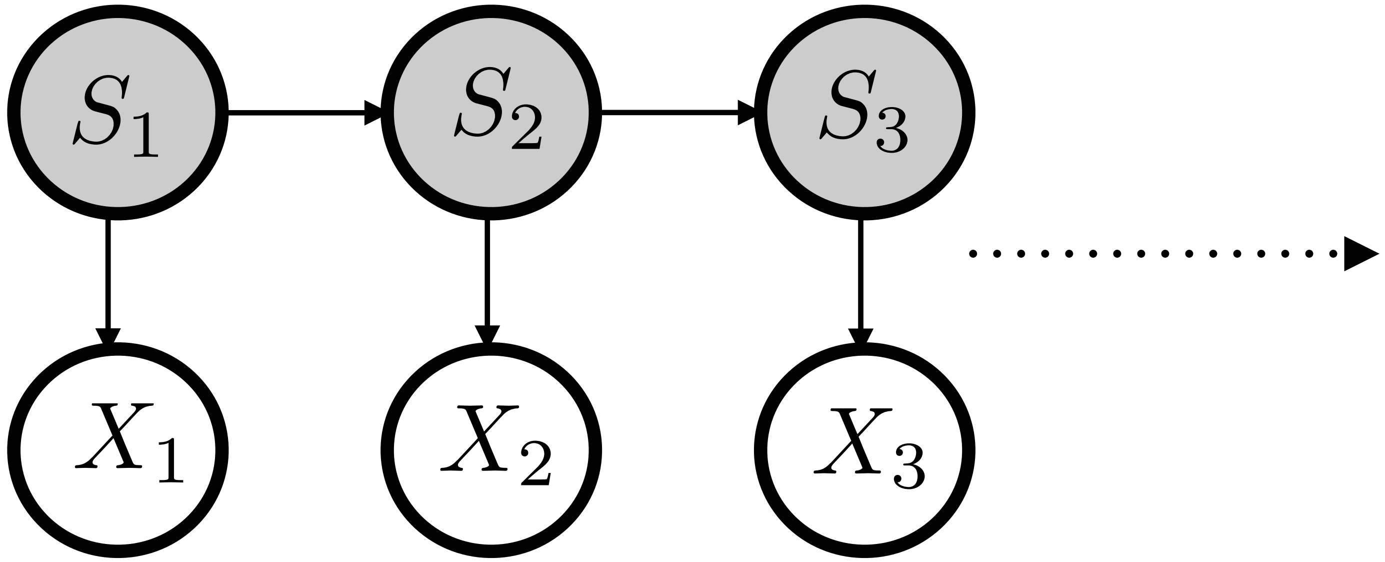


$$\text{message}_{S_{t-1} \mapsto S_t}(k) = P(S_t = k, X_1, \ldots, X_{t-1})$$

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

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

Inference in HMM



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

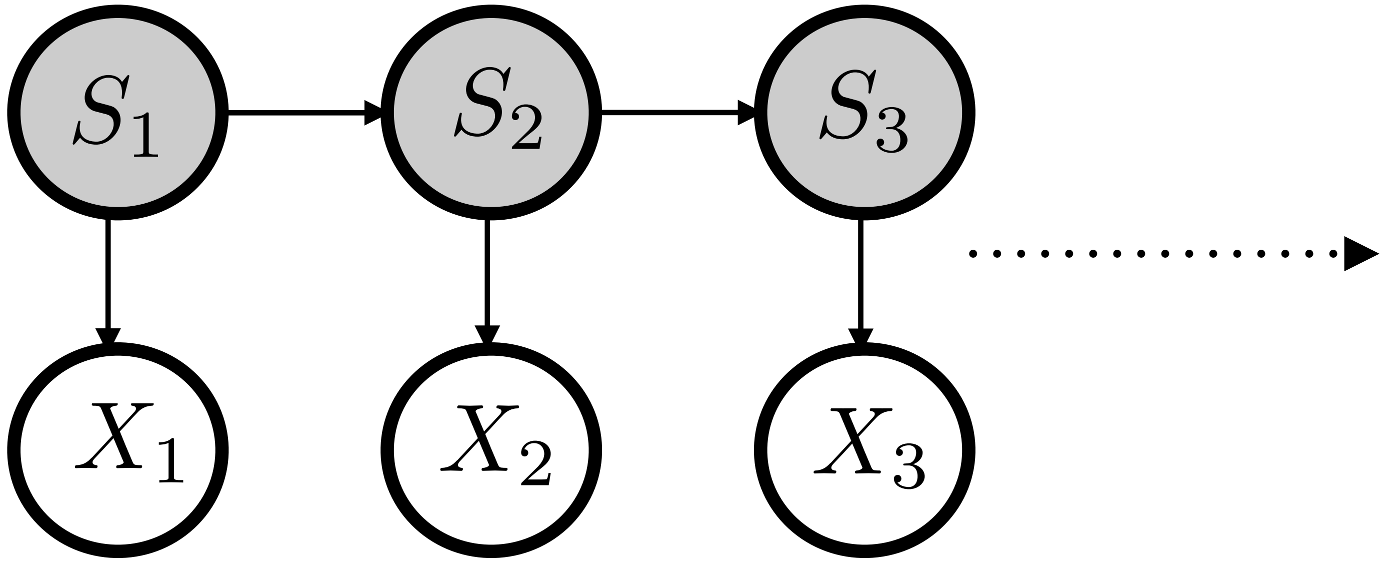
$$\text{message}_{S_{t+1}\mapsto S_t}(k) = P(X_n, \ldots, X_{t+1}|S_t = k)$$

Forward:

$$P(X_1, \ldots, X_{t-1}, S_t = k) = \sum_{j=1}^{K} P(S_t = k|S_{t-1} = j)P(X_{t-1}|S_{t-1} = j)P(X_1, \ldots, X_{t-2}, S_{t-1} = j)$$

$$\text{message}_{S_{t-1}\mapsto S_t}(k) = \sum_{j=1}^{K} P(S_t = k|S_{t-1} = j)P(X_{t-1}|S_{t-1} = j)\text{message}_{S_{t-2}\mapsto S_{t-1}}(j)$$

Inference in HMM



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

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

Backward:

$$P(X_n, \ldots, X_{t+1}|S_t = k) = \sum_{j=1}^{K} P(X_n, \ldots, X_{t+2}|S_{t+1} = j)P(X_{t+1}|S_{t+1} = j)P(S_{t+1} = j|S_t = k)$$

$$\text{message}_{S_{t+1}\mapsto S_t}(k) = \sum_{j=1}^{K} \text{message}_{S_{t+2}\mapsto S_{t+1}}(j)P(X_{t+1}|S_{t+1} = j)P(S_{t+1} = j|S_t = k)$$

Learning Parameters for HMM

- Now that we have algorithm for inference, what about learning
- Given observations, how do we estimate parameters for HMM? Three guesses . . .

EM for HMM (Baum Welch)

- EM algorithm of course, for HMM its referred to as Baum Welch algorithm
- Initialize Transition and Emission probability tables arbitrarily
- For $i = 1$ to convergence:

E-step: For every state variable $t \in \{1, \ldots, n\}$,
Use forward-backward algorithm to compute probabilities of latent variables given obervation

M-step: Optimize weighted log likelihood as usual:

$$\theta^{(i)} = \arg\max_{\theta \in \Theta} \sum_{S_{1,\ldots,n}} P(S_{1,\ldots,n} | X_{1,\ldots,n}, \theta^{(i-1)}) \log P(X_{1,\ldots,n}, S_{1,\ldots,n} | \theta)$$

Lets Simplify M-step

$$\log P(X_{1,\ldots,n}, S_{1,\ldots,n}|\theta) = \log\left(\prod_{t=1}^{n} P(X_t|S_t,\theta)\prod_{t=1}^{n} P(S_t|S_{t-1},\theta)\right)$$

$$= \sum_{t=1}^{n}\log P(X_t|S_t,\theta) + \sum_{t=1}^{n}\log P(S_t|S_{t-1},\theta)$$

Hence,

$$\sum_{S_{1,\ldots,n}} P(S_{1,\ldots,n}|X_{1,\ldots,n},\theta^{(i-1)})\log P(X_{1,\ldots,n}, S_{1,\ldots,n}|\theta)$$

$$= \sum_{t=1}^{n}\sum_{s_t=1}^{K} P(S_t = s_t|X_{1,\ldots,n},\theta^{i-1})\log P(X_t|S_t = s_t,\theta)$$

$$+ \sum_{t=1}^{n}\sum_{s_t,s_{t-1}=1}^{K} P(S_t = s_t, S_{t-1} = s_{t-1}|X_{1,\ldots,n},\theta^{i-1})\log P(S_t|S_{t-1},\theta)$$

E-STEP

- Only need to compute $P(S_t = s_t | X_{1,\ldots,n}, \theta^{i-1})$ and $P(S_t = s_t, S_{t-1} = s_{t-1} | X_{1,\ldots,n}, \theta^{i-1})$ using forward-backward
- First term is immediate

$$P(S_t = s_t | X_{1,\ldots,n}, \theta^{i-1}) \propto m_{S_{t-1} \mapsto S_t}(s_t) \cdot m_{S_{t+1} \mapsto S_t}(s_t) \cdot E^{(i-1)}[s_t, X_t]$$

- For second term,

$$\begin{aligned} & P(S_t = s_i, S_{t-1} = s_{t-1} | X_{1,\ldots,n}, \theta^{i-1}) \\ & \propto m_{S_{t-1} \mapsto S_t}(s_t) T^{(i-1)}[s_{t-1}, s_t] P(S_{t-1} = s_{t-1} | X_{1,\ldots,n}, \theta^{i-1}) \\ & \propto m_{S_{t-1} \mapsto S_t}(s_t) T^{(i-1)}[s_{t-1}, s_t] m_{S_{t-2} \mapsto S_{t-1}}(s_{t-1}) m_{S_t \mapsto S_{t-1}}(s_{t-1}) E^{(i-1)}[s_{t-1}, X_{t-1}] \end{aligned}$$

Why?

E-STEP

$$
\begin{aligned}
&P(S_t = s_t, S_{t-1} = s_{t-1} | X_{1,\ldots,n}, \theta^{i-1}) \\
&\quad = P(S_t = s_t, | S_{t-1} = s_{t-1}, X_{1,\ldots,n}, \theta^{t-1}) P(S_{t-1} = s_{t-1} | X_{1,\ldots,n}, \theta^{i-1}) \\
&\quad = P(S_t = s_t, | S_{t-1} = s_{t-1}, X_{t,\ldots,n}, \theta^{i-1}) P(S_{t-1} = s_{t-1} | X_{1,\ldots,n}, \theta^{i-1}) \\
&\quad \propto P(X_{t,\ldots,n} | S_t = s_t, S_{t-1} = s_{t-1}, \theta^{i-1}) \\
&\qquad\qquad P(S_t = s_t | S_{t-1} = s_{t-1}, \theta^{i-1}) P(S_{t-1} = s_{t-1} | X_{1,\ldots,n}, \theta^{i-1}) \\
&\quad \propto P(X_{t,\ldots,n} | S_t = s_t, \theta^{i-1}) \\
&\qquad\qquad T^{(i-1)}[s_{t-1}, s_t] P(S_{t-1} = s_{t-1} | X_{1,\ldots,n}, \theta^{i-1}) \\
&\quad \propto m_{S_{t-1} \mapsto S_t}(s_t) \cdot T^{(i-1)}[s_{t-1}, s_t] \cdot P(S_{t-1} = s_{t-1} | X_{1,\ldots,n}, \theta^{i-1}) \\
&\quad \propto m_{S_{t-1} \mapsto S_t}(s_t) \cdot T^{(i-1)}[s_{t-1}, s_t] \\
&\qquad\qquad m_{S_{t-2} \mapsto S_{t-1}}(s_{t-1}) \cdot m_{S_t \mapsto S_{t-1}}(s_{t-1}) \cdot E^{(i-1)}[s_{t-1}, X_{t-1}]
\end{aligned}
$$

Baum Welch Algorithm

Initialize T^0, E^0 probability tables

For $i = 1$ to convergence

- E-step:
  - Run Forward-Backward algorithm and compute messages
  - For every t compute $P(S_t = s_t, S_{t-1} = s_{t-1} | X_{1,\ldots,n}, \theta^{i-1})$ and $P(S_t = s_t | X_{1,\ldots,n}, \theta^{i-1})$ as in previous slides
- M-step:

$$\forall u, v \quad T^{(i)}[u, v] = \frac{\sum_{t=2}^{n} P(S_t = v, S_{t-1} = u | X_{1,\ldots,n}, \theta^{i-1})}{\sum_{t=2}^{n} P(S_{t-1} = u | X_{1,\ldots,n}, \theta^{i-1})}$$

$$\forall v, e \quad E^{(i)}[v, e] = \frac{\sum_{t=1}^{n} P(S_t = v | X_{1,\ldots,n}, \theta^{i-1}) \cdot \mathbf{1}_{X_t = e}}{\sum_{t=1}^{n} P(S_t = v | X_{1,\ldots,n}, \theta^{i-1})}$$

Inference for general BN

Bayesian Networks

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

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

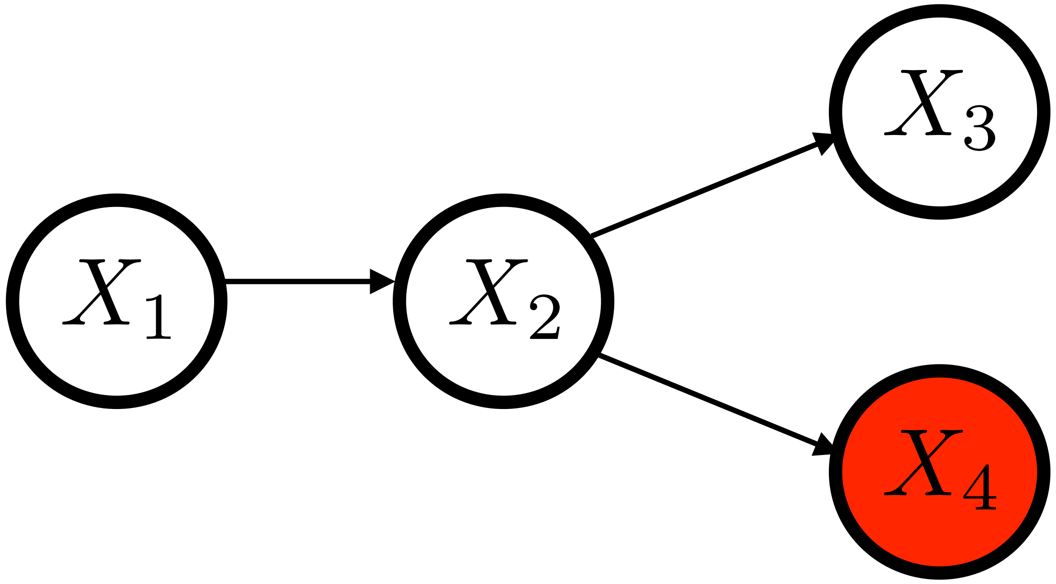
- Hence Bayesian Networks are specified by G along with CPD's over the variables (given their parents)

Variable Elimination: Examples

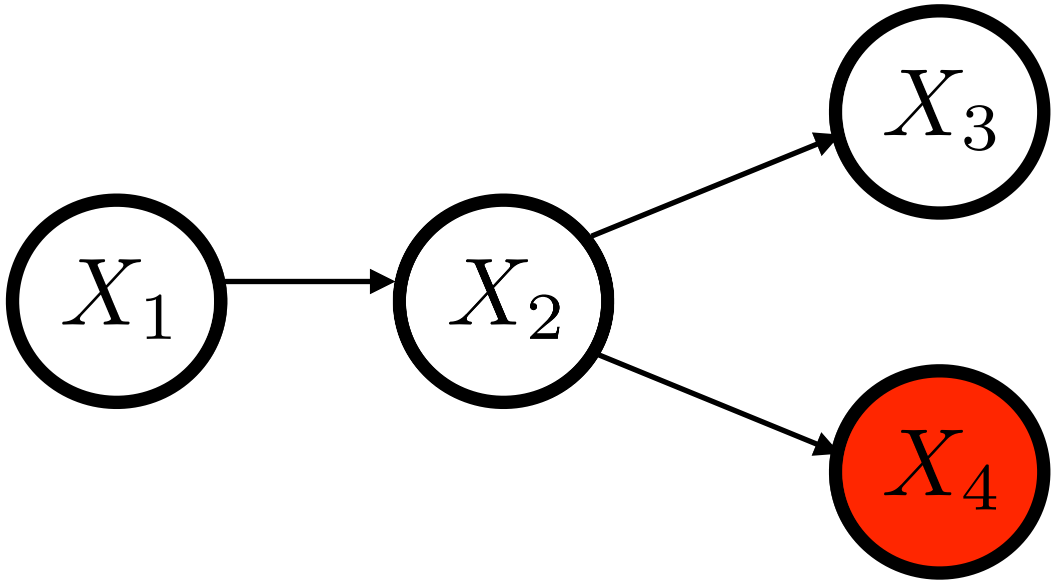
- Marginals are enough:

$$P(X_j = x_j, X_k = x_k | X_i = x_i, X_h = x_h) = \frac{P(X_j = x_j, X_k = x_k, X_i = x_i, X_h = x_h)}{P(X_i = x_i, X_h = x_h)}$$

Variable Elimination: Examples



Variable Elimination: Examples



$$
\begin{aligned}
P(X_4) &= \sum_{x_1} \sum_{x_2} \sum_{x_3} P(X_1 = x_1, X_2 = x_2, X_3 = x_3, X_4) \\
&= \sum_{x_1} \left(P(X_1 = x_1) \sum_{x_2} \left(P(X_2 = x_2 | X_1 = x_1) P(X_4 | X_2 = x_2) \left(\sum_{x_3} P(X_3 = x_3 | X_2 = x_2) \right) \right) \right) \\
&= \sum_{x_1} \left(P(X_1 = x_1) \left(\sum_{x_2} P(X_2 = x_2 | X_1 = x_1) P(X_4 | X_2 = x_2) \right) \right)
\end{aligned}
$$

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

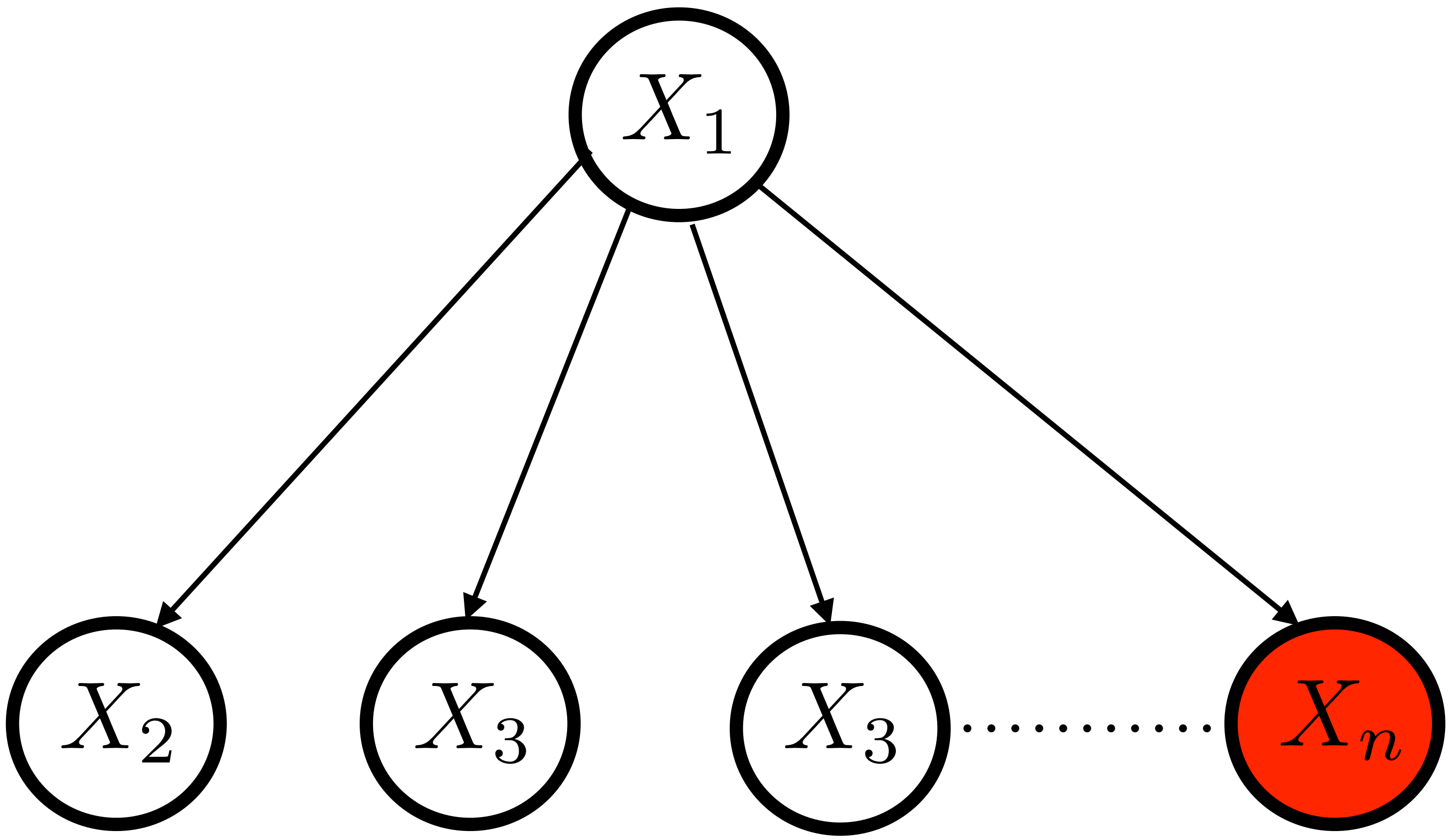
Define new distribution as the sum (over values of X_i) of the product of these distributions

Place the new distribution on List

End

Return List

Variable Elimination: Order Matters



Right order: $O(n)$

Wrong order: $O(2^n)$

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?

Eg. forward backward algorithm for HMM