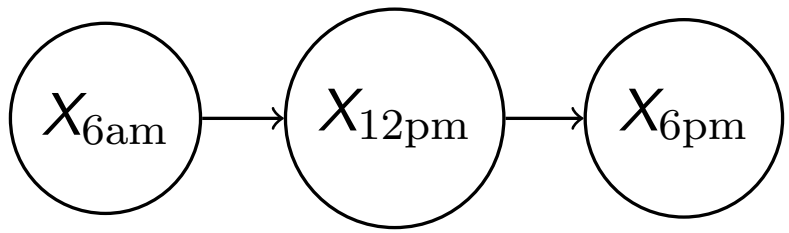
COMP 790-125: Goals for today

- Inference and learning in graphical models
- Forward-backward, root-leaf-root message passing algorithms

A simple location tracking model

Suppose you model person's location by assuming they are at 4 different locations $L = \{\text{home}, \text{work}, \text{gym}, \text{elsewhere}\}$.



Joint distribution over 3 locations for a single day is

$$p(X_{6\text{am}}, X_{12\text{pm}}, X_{6\text{pm}}) = p(X_{6\text{am}})p(X_{12\text{pm}}|X_{6\text{am}})p(X_{6\text{pm}}|X_{12\text{pm}})$$

Q: What is bad about this model?

A simple location tracking model

$$p(X_{6\text{am}} = x) = \begin{cases} 0.8, & x = \text{home}, \\ 0.09, & x = \text{work} \\ 0.09, & x = \text{gym} \\ 0.02, & x = \text{elsewhere} \end{cases}$$

And probability matrices for morning transition

$$p(X_{12\text{pm}}|X_{6\text{am}}) = \begin{array}{l} \textit{from}\backslash\textit{to} \\ \text{home} \\ \text{work} \\ \text{gym} \\ \text{elsewhere} \end{array} \begin{array}{c} \begin{array}{cccc} \text{home} & \text{work} & \text{gym} & \text{elsewhere} \end{array} \\ \begin{pmatrix} 0.01 & 0.9 & 0.045 & 0.045 \\ 0.04 & 0.96 & 0 & 0 \\ 0 & 0.9 & 0 & 0.1 \\ 0 & 0.01 & 0 & 0.99 \end{pmatrix} \end{array}$$

and afternoon transition

$$p(X_{6\text{pm}}|X_{12\text{pm}}) = \begin{array}{l} \textit{from}\backslash\textit{to} \\ \text{home} \\ \text{work} \\ \text{gym} \\ \text{elsewhere} \end{array} \begin{array}{c} \begin{array}{cccc} \text{home} & \text{work} & \text{gym} & \text{elsewhere} \end{array} \\ \begin{pmatrix} 0.99 & 0 & 0.01 & 0 \\ 0 & 0.99 & 0.01 & 0 \\ 0.9 & 0 & 0.01 & 0.09 \\ 0.5 & 0 & 0.5 & 0 \end{pmatrix} \end{array}$$

Questions we can ask in this model

1. Marginal probability of being at work at 12pm

$$p(X_{12\text{pm}} = \text{work})$$

2. Conditional probability of being at work at 12pm if user went to gym in the morning

$$p(X_{12\text{pm}} = \text{work} | X_{6\text{a}} = \text{gym})$$

3. Conditional probability of being at work at 12pm if user went to gym in the morning and is home in the evening

$$p(X_{12\text{pm}} = \text{work} | X_{6\text{am}} = \text{gym}, X_{6\text{pm}} = \text{home})$$

4. Most likely location for the user at 6pm, if user was in the gym at noon

$$\underset{x \in \{\text{home},\text{work},\text{gym},\text{elsewhere}\}}{\text{argmax}} \quad p(X_{6\text{pm}} = x | X_{12\text{pm}} = \text{gym})$$

Questions we can ask in this model

Let's focus on the last one, an example of Maximum-A-Posteriori inference

$$\operatorname*{argmax}_{x \in \{\text{home},\text{work},\text{gym},\text{elsewhere}\}} p(X_{6\text{pm}} = x | X_{12\text{pm}} = \text{gym})$$

We need to compute table $p(X_{6\text{pm}} = x | X_{12\text{pm}} = \text{gym})$

$$\begin{aligned} p(X_{6\text{pm}} = x | X_{12\text{pm}} = \text{gym}) &= \frac{p(X_{12\text{pm}} = \text{gym}, X_{6\text{pm}} = x)}{p(X_{12\text{pm}} = \text{gym})} \\ &= \frac{\sum_v p(X_{6\text{am}} = v, X_{12\text{pm}} = \text{gym}, X_{6\text{pm}} = x)}{\sum_{s,t} p(X_{6\text{am}} = s, X_{12\text{pm}} = \text{gym}, X_{6\text{pm}} = t)} \end{aligned}$$

This table is

$$p(X_{6\text{pm}} = x) = \begin{cases} 0.9, & x = \text{home}, \\ 0, & x = \text{work} \\ 0.01, & x = \text{gym} \\ 0.09, & x = \text{elsewhere} \end{cases}$$

Hence MAP assignment for $X_{6\text{pm}}$ is home.

Inference in a graphical model

We did not use conditional independencies to simplify computation

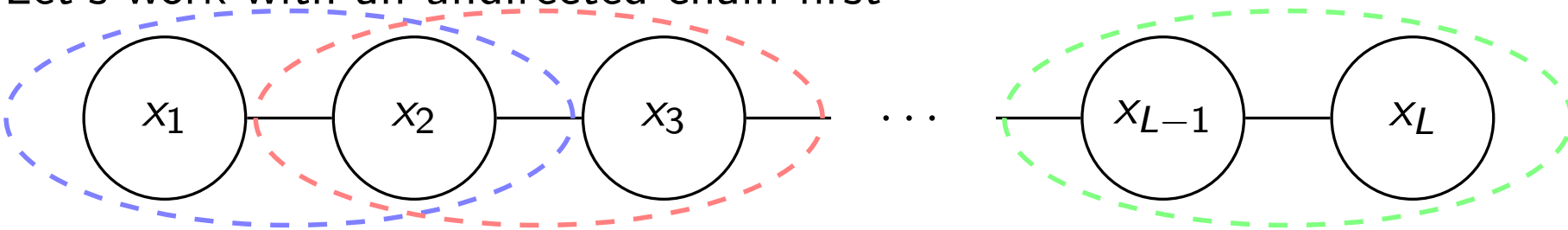
$$p(X_{6\text{pm}} = x | X_{12\text{pm}} = \text{gym}) = \frac{p(X_{12\text{pm}} = \text{gym}, X_{6\text{pm}} = x)}{p(X_{12\text{pm}} = \text{gym})}$$
$$= \frac{\sum_v p(X_{6\text{am}} = v, X_{12\text{pm}} = \text{gym}, X_{6\text{pm}} = x)}{\sum_{s,t} p(X_{6\text{am}} = s, X_{12\text{pm}} = \text{gym}, X_{6\text{pm}} = t)}$$

We need to sum 4 different terms for numerator, and 4^2 different terms for denominator.

Q: If we had tracking for every hour, and we asked the same question – if gym at noon, where to at 6pm –, how many different terms would need to be summed?

Inference in a graphical model

Let's work with an undirected chain first



Recall that for an undirected model joint probability is given by product of potentials across cliques

$$
\begin{aligned}
p(\mathbf{x}) &= \frac{1}{Z}\phi_1(x_1, x_2)\phi_2(x_2, x_3)\ldots\phi(x_{L-1}, x_L) \\
&= \frac{1}{Z}\prod_{l=1}^{L-1}\phi_l(x_l, x_{l+1})
\end{aligned}
$$

Marginalization

Say we are interested in computing a marginal $p(x_k)$ of our undirected chain model

$$p(x_k) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_{k-1}} \sum_{x_k+1} \cdots \sum_{x_L} \frac{1}{Z} \prod_{l=1}^{L-1} \phi(x_l, x_{l+1})$$

doing this naively will be exponential in L.
Instead we distribute ϕs shrewdly

$$p(x_k) = \frac{1}{Z} \times$$
$$\sum_{x_{k-1}} \phi_{k-1}(x_{k-1}, x_k) \sum_{x_{k-2}} \phi_{k-2}(x_{k-2}, x_{k-1}) \cdots \sum_{x_2} \phi_2(x_2, x_3) \sum_{x_1} \phi_1(x_1, x_2)$$
$$\times \sum_{x_{k+1}} \phi_k(x_k, x_{k+1}) \cdots \sum_{x_{L-1}} \phi_{L-1}(x_{L-2}, x_{L-1}) \sum_{x_L} \phi_{L-1}(x_{L-1}, x_L)$$

Marginalization – α, β messages

$$p(x_k) = \frac{1}{Z} \times$$

$$\overbrace{\sum_{x_{k-1}} \phi_{k-1}(x_{k-1}, x_k) \sum_{x_{k-2}} \phi_{k-2}(x_{k-2}, x_{k-1}) \cdots \sum_{x_2} \phi_2(x_2, x_3) \sum_{x_1} \phi_1(x_1, x_2)}^{\alpha_k(x_k)}$$

$$\times \underbrace{\sum_{x_{k+1}} \phi_k(x_k, x_{k+1}) \cdots \sum_{x_{L-1}} \phi_{L-1}(x_{L-2}, x_{L-1}) \sum_{x_L} \phi_{L-1}(x_{L-1}, x_L)}_{\beta_k(x_k)}$$

A complicated way of saying:

$$\alpha_k(x_k) = \sum_{x_1} \cdots \sum_{x_{k-1}} \prod_{l=1}^{k-1} \phi_l(x_l, x_{l+1})$$

$$\beta_k(x_k) = \sum_{x_{k+1}} \cdots \sum_{x_L} \prod_{l=k}^{L-1} \phi_l(x_l, x_{l+1})$$

Marginalization – α, β messages

But also

$$p(x_k) = \frac{1}{Z} \times$$

$$\overbrace{\underbrace{\sum_{x_{k-1}} \phi_{k-1}(x_{k-1}, x_k) \overbrace{\sum_{x_{k-2}} \phi_{k-2}(x_{k-2}, x_{k-1}) \cdots \sum_{x_2} \phi_2(x_2, x_3) \sum_{x_1} \phi_1(x_1, x_2)}^{\alpha_{k-1}(x_{k-1})}}_{\times \sum_{x_{k+1}} \phi_k(x_k, x_{k+1}) \cdots \sum_{x_{L-1}} \phi_{L-1}(x_{L-2}, x_{L-1}) \sum_{x_L} \phi_{L-1}(x_{L-1}, x_L)}}^{\alpha_k(x_k)}$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\beta_{k+1}(x_k+1)}$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\beta_k(x_k)}$$

Lookie a recursion and in case you did not recognize a dynamic programming opportunity 2 slides ago it is winking at you now.

Marginalization – α, β messages

$$\begin{aligned}\alpha_i(x_i) &= \sum_{x_{i-1}} \phi_{x_{i-1}}(x_{i-1}, x_i)\alpha_{i-1}(x_{i-1}) \\ \beta_i(x_i) &= \sum_{x_{i+1}} \phi_{x_i}(x_i, x_{i+1})\beta_{i+1}(x_{i+1})\end{aligned}$$

In case of αs you work forward: using the table computed for α_{i-1} you compute table for α_i.
In case of β you work backward: using the table computed for β_{i+1} you compute table for β_i.

Once you have all α and all β tables, getting marginal probabilities of any stretch of variables is trivial, for example

$$p(x_i, x_{i+1}, x_{i+2}) = \frac{1}{Z}\alpha_i(x_i)\phi_i(x_i, x_{i+1})\phi_{i+1}(x_{i+1}, x_{i+2})\beta_{i+2}(x_{i+2})$$

What about Z?

$$
\begin{aligned}
Z &= \sum_{x_1} \cdots \sum_{x_L} \prod_j \phi_j(x_j, x_j + 1) \\
&= \sum_{x_L} \alpha_L(x_L) \\
&= \sum_{x_1} \beta_1(x_1) \\
&= \sum_{x_i} \alpha_i(x_i)\beta_i(x_i)
\end{aligned}
$$

Avoiding forward-backward bugs

Off by one is the favorite – as a result either a potential is not multiplied in or it is multiplied in a couple of times.

Luckily you know a useful fact about Z and you can assert that your forward and backward pass yield the same Z.

In your code do not multiply potentials and αs, they should be stored in the log domain and you should use the log-sum trick when you need a real domain sum, e.g.

$$\log \alpha_k(x_k) = \mathrm{logsum}(\log \alpha_{k-1}(:) + \log \phi_{k-1}(:, x_k))$$

Inference in an undirected tree

A node x_k splits a chain into two components and α_k β_k gave us their contribution to the marginal $p(x_k)$

With trees, a node splits the tree into multiple components.

$$p(x_k) = \frac{1}{Z} \prod_{x_j \in n(x_k)} m_{x_j, x_k}(x_k)$$

where $n(x_k)$ is the set of neighbors of node x_k in the tree.
In this notation

$$\begin{aligned} \alpha_k(x_k) &= m_{x_{k-1}, x_k}(x_k) \\ \beta_k(x_k) &= m_{x_{k+1}, x_k}(x_k) \end{aligned}$$

Computing messages

Similar to the recursion we derived for chain we can construct a recursion for a tree

$$m_{x_j,x_k}(x_k) = \sum_{x_j} \phi_{jk}(x_j, x_k) \prod_{x_l \in n(x_j), l \neq k} m_{x_l,x_j}(x_j)$$

where ϕ_{jk} is the potential on the clique consisting of variables x_j, x_k.

In the case of αs our forward pass always ensured that the α_{k-1} was computed prior to α_k. We need an order for message computations that provides a similar guarantee.

Root-leaf-root

In an undirected tree we can designate any node a root.

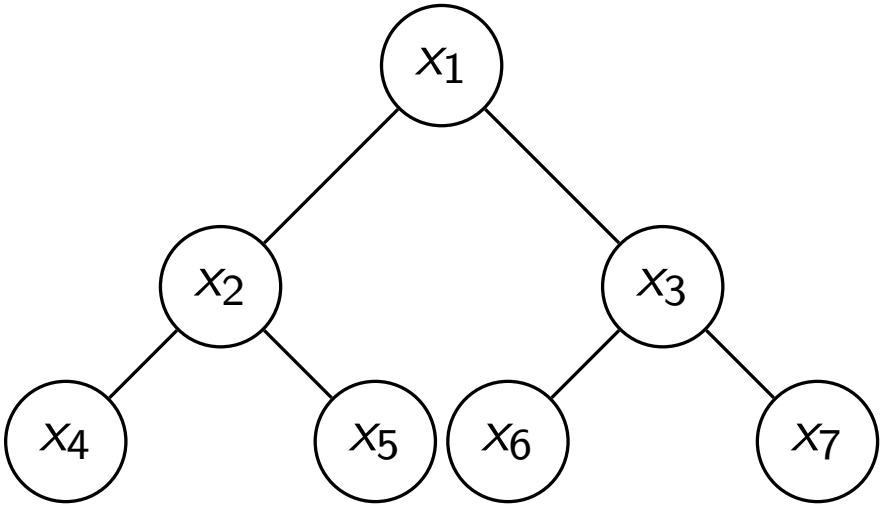
Given a root we can sort nodes by their distance from it.

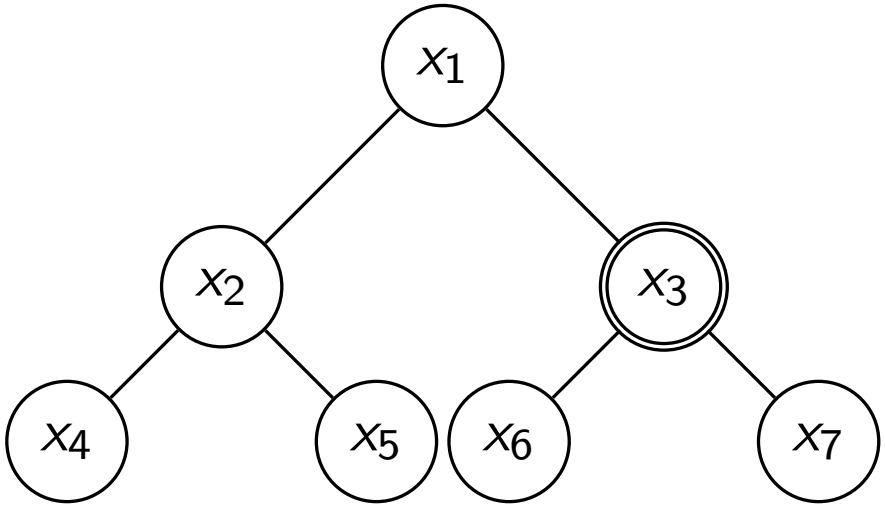
We can now think of a downward pass and a upward pass.

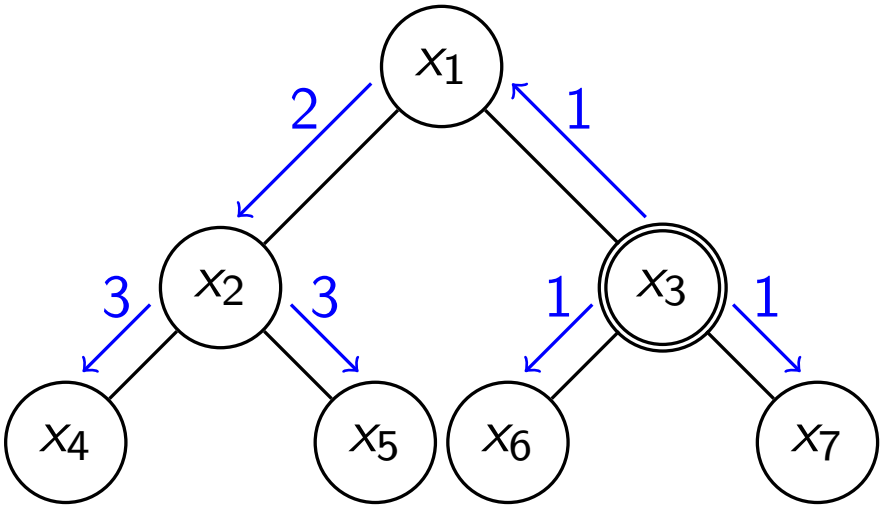
In an upward pass we compute messages from nodes further away to nodes closer and in the order of the distance from the root.

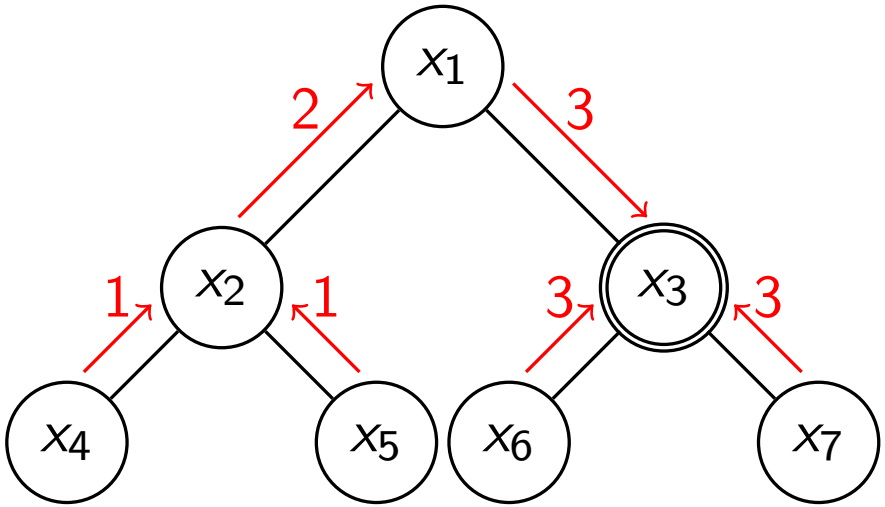
In an downward pass we simply proceed in the opposite order.

Illustration of message computation









Marginal

As in the case of chain, marginal is product of all incoming messages

$$p(x_k) \propto \prod_{x_j \in n(x_k)} m_{x_j,x_k}(x_k)$$

And again

$$Z = \sum_{x_k} \prod_{x_j \in n(x_k)} m_{x_j,x_k}(x_k)$$

for any x_k so you can check your code for producing consistent Z across all variables.

Incorporating observations

Most of the time you will be interested in computing a marginal distribution *but* conditioned on some set of observed variables x_O.

$$p(x_k|x_O = v_O) = \sum_{x_H} p(x_k, x_H|x_O = v_O)$$

where $k \notin H$ and $H \cap O = \emptyset$ and $\{k\} \cup O \cup H = \{1, \ldots p\}$.

We can incorporate observations directly into our potentials by adjusting them

$$\phi^E(x_j = v_j, x_l = v_l) = \begin{cases} 0, & j \in O, x_j \neq v_j \text{ or } l \in O, x_l \neq v_l \\ \phi(x_j, x_l), & \text{otherwise} \end{cases}$$

thus using potentials ϕ^E ensures that configurations that violate the observations have probability 0.