CS281 Section #5 Notes, v 1.1

Due:

Boring Title: Undirected Graphical Models, Exciting Title: Markov Random Fields

1 Independence properties are simpler than in DAGS

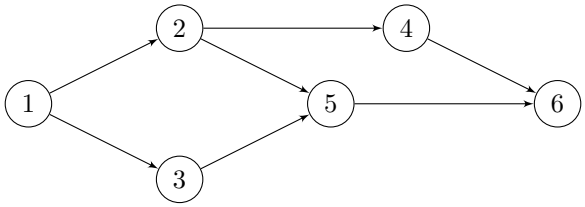
- **global Markov property**: for node sets A, B, C in graph G, $\mathbf{x}_A \perp \mathbf{x}_B \,|\, \mathbf{x}_C$ iff removing all nodes in C leaves no path connecting any node in A to any node in B
- **undirected local Markov property**: given its set of immediate neighbors (i.e. its **Markov blanket**), a node t is conditionally independent of all other nodes
- **pairwise Markov property**: two nodes conditionally independent given the rest if no direct edge between them
- global implies pairwise and vice versa

Example 1. Markov blanket for UGMs is defined above. What does the Markov blanket in DGMs include?

2 DGM to UGM Conversion

For each pair of parent nodes that share a child, marry them by adding an edge. Then drop the direction of all edges. Finally, use the CI properties of UGMs to answer a CI query more easily than in a DGM. At worst, we lose a few CI assumptions. However, we should not introduce new ones.

Example 2. Convert the following DGM to UGM:



3 Distributions and Graphs

- graph G is an **I-map** of distribution p if $I(G) \subseteq I(p)$ where $I(.)$ is "the set of conditional independencies encoded by...". **Question:** is a naive bayes DGM with extra connections among the x's an I-map for $p(x, y)$ with the typical naive Bayes assumptions?
- G is a **perfect map** of p if $I(G) = I(p)$
- DGMs, UGMs perfect maps for different (but overlapping) sets of distributions
- Graphs only specify the conditional independence structure, nothing about the parameters, distributions.
- Shape of graph determines difficulty of inference

4 Parameterization of UGMs

Each edge in a DGM represents a normalized conditional probability distribution. What do undirected edges represent?

- A **clique** is a set of nodes such that each node has an edge with all others in the set. A **maximal clique** is the click of the largest size that a node belongs to.
- Each clique has a **potential function** that assigns each possible configuration a score. Considering just 8 nodes with binary values, this is 2^8 states to score for just the one clique.
- We use **log-potentials** $\theta_c(x_c)$. The book uses potentials $\psi_c(x_c) = \exp[\theta_c(x_c)]$
- joint distribution over $\mathbf{x}$'s proportional to product over all potentials

$$\begin{aligned} p(x_1, ..., x_T) &= \exp\big[\sum_c \theta_c(x_c) - A(\theta)\big] \\ &\propto \prod_c \exp\big[\theta_c(x_c)\big] = \prod_c \psi_c(x_c) \end{aligned}$$

Very important: UGMs are **globally normalized** ($A(\theta)$ term) instead of **locally normalized** like each edge of a DGM. More about this below in CRFs.

Example 3. Draw the UGM corresponding to the Naive Bayes DGM and describe how you would form the potential functions

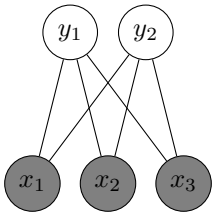
5 Ising model

Example 4. Let X_s be the Bernoulli random variable associated with node s on a graph $G = (V, E)$ with $|V| = m$. X_s takes on the **spin** value $\{-1, +1\}$. This could correspond to magnet orientation, pixel value, and many other things. X_s and X_t interact if they have a direct edge in the graph (we expect a certain relationship between neighboring pixels). Let $\theta_{st} \in \mathbb{R}$ be the strength of the edge (s, t) and θ_s be the marginal potential for node s (determined by an external field). Write down the joint probability of the set of $\mathbf{x}$ in terms of potential functions that correspond to the model above. It should look like an exponential family. How do you calculate $A(\theta)$? What is the distribution's dimension?

6 Conditional Random Fields

- Clique potentials functions over hidden configurations y conditioned on observed features x
- Can be thought of as a **structured output** form of logistic regression
- CRF advantage over MRF like logistic regression vs. naive Bayes, i.e. don't need to model things that we always observe
- **x are not necessarily independent**

An example with pairwise cliques:



7 Stereo Depth Reconstruction CRF

Example 5. Given two images $\mathbf{x}_L$ and $\mathbf{x}_R$ taken at a small known angle difference, recover the depth measurement of each pixel by estimating the position of pixel $\mathbf{x}_L(i,j)$ in $\mathbf{x}_R$, which will have a different horizontal value and the same vertical value. This difference, call it y_s, is the **disparity** at i,j. By computing each such disparity, a disparity map shows an estimate of depth. Write down a suitable potential function $\psi_s(y_s, \mathbf{x})$. How could you define a $\psi_{st}(y_s, y_t)$ to enforce similar disparity in neighboring pixels?

8 Distributive Property

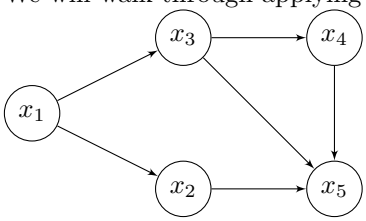
8.1 Review and Walkthrough

We are interested in calculating the marginal probability of a random variable x_i in our graphical model. However, we have so many other variables that summing over all of these other variables in the joint distribution can quickly be infeasible. We will use a key idea, "*pushing sums inside products*", which is just to say that multiplication distributes over addition:

$$\sum_i \sum_j \sum_k a_i b_j c_k = \sum_i a_i \left[\sum_j b_j \left(\sum_k c_k \right) \right]$$

but for $i \in [0,1], j \in [0,1,2], k \in [0,1,2,3]$ the LHS corresponds to 48 multiplications and 23 additions whereas the RHS is only 5 multiplications and 6 additions!

We will walk through applying the distributive property on the following UGM:



We factor the joint probability distribution for our UGM by the potential functions for maximal cliques:

$$p(\mathbf{X}) \propto \psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{25}(x_2, x_5)\psi_{345}(x_3, x_4, x_5)$$

Suppose we are interested in the marginal probability of x_1. We have that:

$$\begin{aligned} p(x_1) &= \sum_{x_2:x_5} p(\mathbf{X}) \\ &= \sum_{x_2:x_5} \psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{25}(x_2, x_5)\psi_{345}(x_3, x_4, x_5) \end{aligned}$$

From here, we can see that the order in which we sum out variables can make a difference. Consider the elimination ordering (x_5, x_4, x_3, x_2):

$$\begin{aligned} p(x_1) &= \sum_{x_2:x_5} \psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{25}(x_2, x_5)\psi_{345}(x_3, x_4, x_5) \\ &= \sum_{x_2:x_4} \psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3) \underbrace{\sum_{x_5} \psi_{25}(x_2, x_5)\psi_{345}(x_3, x_4, x_5)}_{m_5(2,3,4)} \end{aligned}$$

(The notation here foreshadows that $m_5(2,3,4) = \sum_{x_5} \psi_{25}(x_2, x_5)\psi_{345}(x_3, x_4, x_5)$ can be thought of as the 'message' from 5 to its neighbors 2, 3, and 4.)

Continuing on in this way, we have:

$$
\begin{aligned}
p(x_1) &= \sum_{x_2:x_4} \psi_{12}(x_1,x_2)\psi_{13}(x_1,x_3)m_5(2,3,4) \\
&= \sum_{x_2:x_3} \psi_{12}(x_1,x_2)\psi_{13}(x_1,x_3) \underbrace{\sum_{x_4} m_5(2,3,4)}_{m_4(x_2,x_3)} \\
&= \sum_{x_2:x_3} \psi_{12}(x_1,x_2)\psi_{13}(x_1,x_3)m_4(x_2,x_3) \\
&= \sum_{x_2} \psi_{12}(x_1,x_2) \underbrace{\sum_{x_3} \psi_{13}(x_1,x_3)m_4(x_2,x_3)}_{m_3(x_1,x_2)} \\
&= \sum_{x_2} \psi_{12}(x_1,x_2)m_3(x_1,x_2) = m_2(x_1)
\end{aligned}
$$

Then we just need to normalize $m_2(x_1)$ to calculate the marginal probability on x_1.

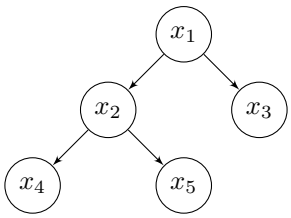
9 Belief Propagation for Trees

9.1 Motivation, Review

- For an undirected tree, the edges are exactly the maximal cliques
- The Hammersley-Clifford Theorem: represent the joint distribution as the product of potentials $\psi_{i,j}(i,j)$ for each edge.
- Murphy and class notation includes unary potentials. So we have a tree represented by

$$
p(\mathbf{x}|\mathbf{v}) = \frac{1}{Z(v)} \prod_{s\in\nu} \psi_s(x_s) \prod_{(s,t)\in E} \psi_{(s,t)}(x_s,x_t)
$$

Consider the following tree



Suppose we want to compute the marginal for x_3. Pushing sums inside products with an ordering of $(5,4,2,1)$, we would have to calculate and store the following messages:

$$
\begin{aligned}
m_5(x_2) &= \sum_{x_5} \psi_5(x_5)\psi_{2,5}(x_2, x_5) \\
m_4(x_2) &= \sum_{x_4} \psi_4(x_4)\psi_{2,4}(x_2, x_4) \\
m_2(x_1) &= \sum_{x_2} \psi_2(x_2)\psi_{1,2}(x_1, x_2)m_4(x_2)m_5(x_2) \\
m_1(x_3) &= \sum_{x_1} \psi_1(x_1)\psi_{1,3}(x_1, x_3)m_2(x_1)
\end{aligned}
$$

Similarly, to compute the marginal for x_1 with elimination ordering $(4, 5, 3, 2, 1)$, the following messages would be needed:

$$
\begin{aligned}
m_4(x_2) &= \sum_{x_4} \psi(x_4)\psi(x_4, x_2) \\
m_5(x_2) &= \sum_{x_5} \psi(x_5)\psi(x_5, x_2) \\
m_3(x_1) &= \sum_{x_3} \psi(x_3)\psi(x_3, x_1) \\
m_2(x_1) &= \sum_{x_2} \psi(x_2)\psi(x_2, x_1)m_5(x_2)m_4(x_2)
\end{aligned}
$$

So that

$$
\begin{aligned}
p(x_1) &\propto \sum_{x_2:x_5} \psi_1(x_1)\psi_2(x_2)\psi_3(x_3)\psi_4(x_4)\psi_5(x_5)\psi(x_1, x_3)\psi(x_1, x_2)\psi(x_2, x_4)\psi(x_2, x_5) \\
&= \psi_1(x_1)m_3(x_1)m_2(x_1)
\end{aligned}
$$

Notice the following:

- We see that to compute the marginal probability at node x_1, we multiply its unary potential with all its incoming messages:

  $$p(x_i) \propto \psi_i(x_i) \prod_{n \in \text{neighbors}(i)} m_{n \to i}(x_i)$$

  In this case, it is the messages from x_3 and x_2. However, these messages that x_3 and x_2 deliver to x_1 also depends on messages from *their* neighbors. More formally, to compute the message $m_{s \to t}(x_t)$, we need to have already computed the messages from the incoming neighbors of s: $m_{r \to s}(x_s)$ for $r \in N(s) \setminus t$.

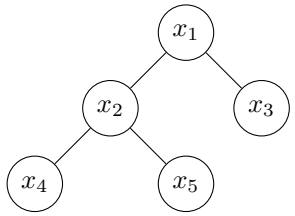
- In computing the marginals for x_1 and x_3, we need some of the same computations, namely, $m_4(x_2), m_5(x_2)$, $m_2(x_1)$. It would be nice to have a procedure that identifies and caches these computations that are 'reused' for the calculations of all marginal probabilities of interest.

This can be done through *belief propagation*, also called *sum-product algorithm on trees*. We can think of the partial sums reused to calculate marginals as the *messages*. Belief propagation also has a nice effect that we will end up calculating all the marginals (*beliefs*) for our tree.

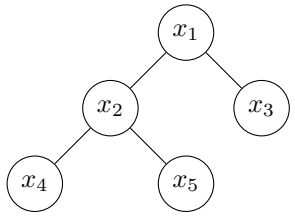
9.2 Sequential Belief Propagation

The sequential implementation of belief propagation is as follows:

- Pick an arbitrary node as the root. This will give the tree an ordering of parent, child relationships. For example, if we pick x_2 to be the root, we can represent the parent-child relationships with the following graph, where the directed edge indicates parent $\rightarrow$ child. Note our tree itself is still an undirected tree - the directed edges only have meaning relative to the chosen root.



- Bottom up (leaves to root): Graphically, we calculate these messages:
(Note) Each graph below shows multiple time steps. In the serial BP procedure, a node can only send a message after it has received all its messages, so we proceed one by one starting with a leaf node.



Defns (Murphy):

$$m^-_{s\rightarrow t}(x_t) = \sum_{x_s} \psi_{s,t}(x_s, x_t)\text{bel}^-_s(x_s)$$

$$\text{bel}^-_s(x_s) = \frac{1}{Z_s}\psi_s(x_s) \prod_{c\in \text{Ch}(s)} m^-_{c\rightarrow s}(x_s)$$

So on the bottom up sweep of belief propagation, we have calculated:

$$m^-_{4\rightarrow 2}(x_2) = \sum_{x_4} \psi_{2,4}(x_2, x_4)\text{bel}^-_4(x_4) = \sum_{x_4} \psi_{2,4}(x_2, x_4)\psi_4(x_4)$$

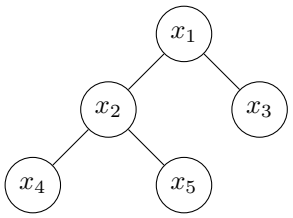
$$m^-_{5\rightarrow 2}(x_2) = \sum_{x_5} \psi_{2,5}(x_2, x_5)\psi_5(x_5)$$

$$m^-_{3\rightarrow 1}(x_1) = \sum_{x_3} \psi_{1,3}(x_1, x_3)\psi_3(x_3)$$

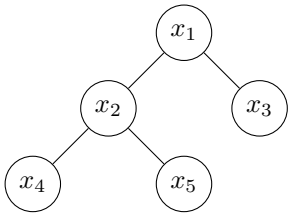
$$\begin{aligned} m^-_{1\rightarrow 2}(x_2) &= \sum_{x_1} \psi_{1,2}(x_1, x_2)\text{bel}^-_1(x_1) \\ &= \sum_{x_1} \psi_{1,2}(x_1, x_2)\psi_1(x_1)m_{3\rightarrow 1}(x_1) \end{aligned}$$

In the upward sweep, we have calculated half of the $2(N-1)$ total messages in our graph.

- Top down (root to leaves):
  We will be calculating these top-down messages:



  Since a node can only send an outgoing message if it has received all of its incoming messages, we see that we will reuse our previous bottom-up messages in our calculations for top-down messages. For example, the outgoing top-down message from x_2 to x_4 relies on the incoming messages to x_2. By design, we have already calculated these!



  Formally:

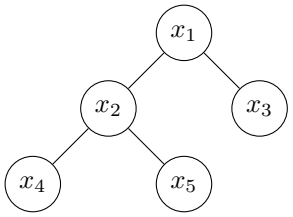
$$m^+_{t\to s}(x_s) = \sum_{x_t} \psi_{s,t}(x_s, x_t)\psi_t(x_t) \prod_{c\in \mathrm{ch}(t)\setminus s} m^-_{c\to t}(x_t) \prod_{p\in pa(t)} m^+_{p\to t}(x_t)$$

  So the messages we produce on the top-down sweep are:

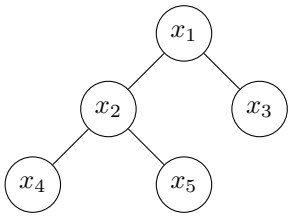
$$\begin{aligned}
m^+_{2\to 4}(x_4) &= \sum_{x_2} \psi_{2,4}(x_2, x_4)\psi_2(x_2) m^-_{5\to 2}(x_2) m^-_{1\to 2}(x_2) \\
m^+_{2\to 5}(x_5) &= \sum_{x_2} \psi_{2,5}(x_2, x_5)\psi_2(x_2) m^-_{4\to 2}(x_2) m^-_{1\to 2}(x_2) \\
m^+_{2\to 1}(x_1) &= \sum_{x_2} \psi_{2,1}(x_2, x_1)\psi_2(x_2) m^-_{4\to 2}(x_2) m^-_{5\to 2}(x_2) \\
m^+_{1\to 3}(x_3) &= \sum_{x_1} \psi_{1,3}(x_1, x_3)\psi_2(x_2) m^+_{2\to 1}(x_1)
\end{aligned}$$

In doing both the bottom-up and top-down sweeps, we have calculated *all* the messages needed to find the beliefs at *every node*. This is perhaps best shown graphically:

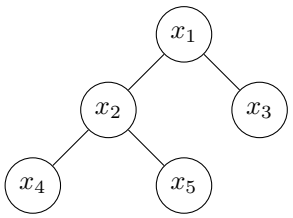
- After doing the steps above, calculating belief at node x_1 requires:
  $\mathrm{bel}(x_1) \propto \psi_1(x_1) m^+_{2\to 1}(x_1) m^-_{3\to 1}(x_1)$, and indeed we have the required messages to compute $m^+_{2\to 1}(x_1)$



- Calculating the belief at node x_3 requires the following messages: $\text{bel}(x_3) \propto \psi_3(x_3) m^+_{1\to 3}(x_3)$

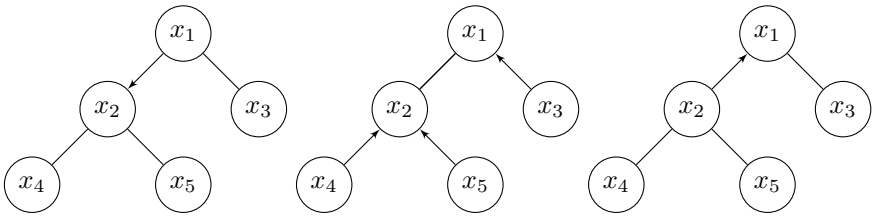


- We get the belief for node x_2 just from the upward pass, since we chose x_2 to be our root node: $\text{bel}(x_2) \propto \psi_2(x_2) m^-_{1\to 2}(x_2) m^-_{4\to 2}(x_2) m^-_{5\to 2}(x_2)$



9.3 Parallel Belief Propagation

The BP algorithm in the previous section is inherently sequential. But we can also implement BP in parallel. We no longer pick a root node and propagate messages up and down. Instead, at every time step, we propagate outgoing messages from all nodes that are ready to send them, i.e. all nodes that have received all their other incoming messages.



After these 3 time steps, we have computed all the messages needed to find beliefs at every node. The procedure is equivalent to serial BP.

The serial Belief Propagation is optimal for tree structure. For a graph with loop, we could still use parallel BP to get approximation marginals. At each step, all nodes receive messages from their neighbors in parallel, update their belief states,

$$\text{bel}_s(x_s) \propto \psi_s(x_s) \prod_{c \in \text{nb}(s)} m_{c\to s}(x_s)$$

and finally send new messages back out to their neighbors:

$$m_{s\to t}(x_t) = \sum_{x_s} \psi_{s,t}(x_s, x_t) \psi_s(x_s) \prod_{c \in \text{nb}(s)\setminus t} m_{c\to s}(x_s)$$

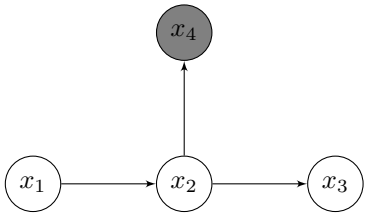
This process repeats until convergence.

9.4 Only Trees Please

BP can get exact solutions for trees. For graphs with loops, it is NP-hard to compute exact marginals and the complexity depends on the treewidth of the graph. Nevertheless, BP can give approximate marginals in many cases. There are also other algorithms to deal with with graphs with loops.

9.5 Exercises

1. Numerical example: All x_i are binary RV's.



$$\psi_{12}(x_1, x_2) = \begin{bmatrix} 1.0 & 0.9 \\ 0.9 & 1.0 \end{bmatrix}, \psi_{23}(x_2, x_3) = \begin{bmatrix} 0.1 & 1.0 \\ 1.0 & 0.1 \end{bmatrix}, \psi_{24}(x_2, x_4) = \begin{bmatrix} 1.0 & 0.1 \\ 0.1 & 1.0 \end{bmatrix}, \text{ and suppose } x_4 = 0$$

2. Show BP on a HMM is exactly the forwards-backwards algorithm.