Introduction to Loopy Belief Propagation
Belief Propagation

What is BP?

Belief Propagation is a dynamic programming approach to answering conditional probability queries in a graphical model.

Given some subset of the graph as evidence nodes (observed variables $E$), compute conditional probabilities on the rest of the graph (hidden variables $X$).

BP gives exact marginals when the graph is a tree (ie. has no loops), but only approximates the true marginals in loopy graphs.
Belief Propagation

- **Idea**: BP works by peer-pressure: a node $X$ determines a final belief distribution by listening to its neighbors.
- Evidence enters the network at the observed nodes and propagates throughout the network.
- Adjacent nodes exchange messages telling each other how to update *beliefs*, based on priors, conditional probabilities and evidence.
- We keep passing messages around until a stable belief state is reached (if ever).
Define $\lambda_Y(x)$ as the message to $X$ from a child node $Y$, indicating $Y$’s opinion of how likely it is that $X = x$.

If $X$ is observed ($X \in E$), allow a message to itself: $\lambda_X(x)$.

Define $\pi_X(u)$ as the message to $X$ from its parent $U$, used to reweight the distribution of $X$ given that $U = u$.

Keep passing messages around until the beliefs converge. We allow messages to change over time: $\lambda^{(t)}(x)$ is a message at time $t$.

Belief is the normalized product of all incoming messages after convergence:

$$BEL_X(x) = \alpha \lambda(x) \pi(x) \approx \Pr[X = x | E]$$
Belief Propagation

Message Passing Example (Incoming)

At step $t$:

$$
\lambda_{Y_k}^{(t)}(x), \pi_X^{(t)}(u_k)
$$
Belief Propagation

Message Passing Example (Outgoing)

At step $t + 1$:

Messages:

$\lambda_{Y_k}^{(t)}(x)$, $\pi_{X}^{(t)}(u_k)$, $\lambda_{X}^{(t+1)}(u_i)$, $\pi_{Y_j}^{(t+1)}(x)$
Belief Propagation

Initial Conditions

- If $X$ has no parents, initialize with the prior:
  \[ \pi(x) = \Pr[X = x]. \]

- If $X$ is an observed node with value $e$,
  \[ \lambda(x) = \begin{cases} 1 & x = e \\ 0 & \text{otherwise} \end{cases} \]

- If $X$ is not observed and has no children, $\lambda^{(0)}(x) = 1$.

- We start sending messages from observed nodes, and instantiate messages for hidden variables along the way.
Belief Propagation

Building Messages

- For a node $X$ with parents $U = \{U_1, \ldots, U_n\}$ and children $Y = \{Y_1, \ldots, Y_m\}$:

- Incoming messages to node $X$ at time $t$:

  \[
  \lambda^{(t)}(x) = \lambda_X(x) \prod_j \lambda_{Y_j}^{(t)}(x)
  \]

  \[
  \pi^{(t)}(x) = \sum_u \Pr[X = x | U = u] \prod_k \pi_X^{(t)}(u_k)
  \]

- Outgoing messages from node $X$ at time $t + 1$:

  \[
  \lambda_X^{(t+1)}(u_i) = \alpha \sum_x \lambda^{(t)}(x) \sum_{u \setminus u_i} \Pr[X = x | U = u] \prod_{k \neq i} \pi_X^{(t)}(u_k)
  \]

  \[
  \pi_{Y_j}^{(t+1)}(x) = \alpha \pi^{(t)}(x) \lambda_X(x) \prod_{k \neq j} \lambda_{Y_k}^{(t)}(x)
  \]
Belief Propagation in the cloudy/rainy/sprinkler/wet grass network. We observe that the grass is wet $W = 1$ and calculate a posterior distribution.
Loopy BP

- BP may not give exact results on loopy graphs, but we use it anyway: iterate until convergence.
- The marginals are often good approximations to the true marginals found by the junction tree algorithm.
- If BP does not converge, it may oscillate between belief states.