Exact inference in singly-connected networks

- A singly connected BN = polytree (disregarding the arc directions, no two nodes can be connected with more than one path).

![Singly-connected and multi-connected networks](image)
Belief propagation

- Developed by Judea Pearl
- Computes the marginal distribution of an unobserved variable given the observed ones.
- A message-passing algorithm:
  - Each node maintains a belief of its state
  - Nodes pass messages to their neighbors and update their beliefs based on received messages
Belief propagation in chains

- A node can have at most one parent and child, no loops.
- We want to compute the marginal probability \( P(X \mid e) \), where the evidence \( e \) is an instantiation of node set \( E \).
- Let us partition the evidence \( e \) into evidence from “downstream” \( e^+ \) and evidence from “upstream” \( e^- \).

\[
P(X \mid e) = P(X \mid e^+, e^-) \\
\propto P(e^- \mid X, e^+) P(X \mid e^+) = P(e^- \mid X) P(X \mid e^+)
\]

where \( X \rightarrow \) is a directed edge.
Belief propagation in chains

- Let us define messages
  - Diagnostic support:
    \[ \lambda(X) = P(e^- | X) \]
  - Causal support:
    \[ \pi(X) = P(X | e^+) \].
- Now \( P(X | e) \propto \lambda(X)\pi(X) = BEL(X) \).
- Node \( X \) sends the message \( \lambda(U) \) to its parent and \( \pi(X) \) to its child.
- Note that the messages \( \lambda(X) \) and \( \pi(X) \) are vectors; they contain one entry per each possible value of \( X \).
- Let \( \lambda(X = x) = P(e^- | X = x) \) or shortly \( \lambda(x) \). Similarly for \( \pi \).
Belief propagation in chains

\[ \pi(e^+) \quad \pi(T) \quad \pi(U) \quad \pi(X) \quad \pi(Y) \quad \pi(Z) \]

\[ e^+ \quad T \quad U \quad X \quad Y \quad Z \quad e^- \]

\[ \lambda(e^+) \quad \lambda(T) \quad \lambda(U) \quad \lambda(X) \quad \lambda(Y) \quad \lambda(Z) \]
Belief propagation in chains

How to compute the messages?

\[ \lambda(U = u) = P(e^- | U = u) \]
\[ = \sum_x P(e^- | X = x)P(X = x | U = u) \]
\[ = \sum_x \lambda(X = x)P(X = x | U = u) \]

\[ \pi(X = x) = P(X = x | e^+) \]
\[ = \sum_u P(X = x | U = u)P(U = u | e^+) \]
\[ = \sum_u P(X = x | U = u)\pi(U = u) \]
Belief propagation in chains

- **Initialization:**
  - For nodes $E$ with evidence $e$
    \[
    \lambda(E = e) = 1, \text{ otherwise } \lambda(E = x) = 0 \\
    \pi(E = e) = 1, \text{ otherwise } \pi(E = x) = 0
    \]
  
  - Nodes with no parents
    \[
    \pi(x) = P(x) \quad \text{(prior probabilities)}
    \]
  
  - Nodes with no children
    \[
    \lambda(x) = 1, \text{ for all } x
    \]
Example

- 5 binary variables
- \( P(E2 = 0) = 0.4 \)
- \( P(U = 0 | E2 = 0) = 0.8, P(U = 0 | E2 = 1) = 0.3 \)
- \( P(X = 0 | U = 0) = 0.6, P(X = 0 | U = 1) = 0.9 \)
- \( P(Y = 0 | X = 0) = 0.7, P(Y = 0 | X = 1) = 0.3 \)
- \( P(E1 = 0 | Y = 0) = 0.4, P(E1 = 0 | Y = 1) = 0.7 \)

We observe that \( E1 = 0 \) and \( E2 = 0 \). What are \( P(U | E) \), \( P(X | E) \) and \( P(Y | E) \)?
Belief propagation in trees

- Every node has at most one parent.
- Differences compared to chains:
  - Each node must combine impacts of the $\lambda$-messages obtained from its children.
  - Each node should distribute a separate $\pi$-message to each of its children.
Belief propagation in trees

- Messages
Belief propagation in trees

- Notation: $\lambda_Y(X)$ is the $\lambda$-message that $Y$ sends to $X$, $\pi_Y(X)$ is the $\pi$-message that $X$ sends to $Y$, $e_Y^{-}$ is the evidence that is “connected” to $X$ via node $Y$.

- How to compute $P(X | e)$?

$$\begin{align*}
P(X | e) &= P(X | e^+, e_Y^-, e_Z^-) \\
&\propto P(X | e^+)P(e_Y^- | X, e^+)P(e_Z^- | X, e^+) \\
&= P(X | e^+)P(e_Y^- | X)P(e_Z^- | X) \\
&= \pi(X)\lambda_Y(X)\lambda_Z(X) \\
&= BEL(X)
\end{align*}$$
Belief propagation in trees

- Node $X$ has a parent $U$ and children $Y_1, \ldots, Y_k$.
- Belief updating

\[ \text{BEL}(x) = \lambda(x)\pi(x), \]

where

\[ \lambda(x) = \prod_{i=1}^{k} \lambda_{Y_i}(x) \]

and

\[ \pi(x) = \sum_u P(X = x \mid U = u)\pi_X(U = u) \]
Belief propagation in trees

- How to compute a message from $X$ to its parent $U$?

\[
\lambda_X(u) = P(e_Y^-, e_Z^- | u)
\]
\[
= \sum_x P(e_Y^-, e_Z^- | u, x)P(x | u)
\]
\[
= \sum_x P(e_Y^-, e_Z^- | x)P(x | u)
\]
\[
= \sum_x P(e_Y^- | x)P(e_Z^- | x)P(x | u)
\]
\[
= \sum_x \lambda(x)P(x | u)
\]
Belief propagation in trees

- How about a message from $X$ to its child $Y$?

\[
\pi_Y(x) \propto P(e^-_Z \mid x, e^+e)P(x \mid e^+)
= P(e^-_Z \mid x)P(x \mid e^+)
= \lambda_Z(x)\pi(x)
= \frac{BEL(x)}{\lambda_Y(x)}
\]
Example

- 4 binary variables
- \( P(U = 0) = 0.4 \)
- \( P(X = 0 \mid U = 0) = 0.8, \ P(X = 0 \mid U = 1) = 0.3 \)
- \( P(Y = 0 \mid X = 0) = 0.9, \ P(Y = 0 \mid X = 1) = 0.6 \)
- \( P(Z = 0 \mid X = 0) = 0.6, \ P(Z = 0 \mid X = 1) = 0.8 \)
- We observe that \( Z = 0 \). What are \( P(U \mid Z = 0) \), \( P(X \mid Z = 0) \) and \( P(Y \mid Z = 0) \)?
Belief propagation in polytrees

- Nodes can have multiple parents
- No loops

Differences compared to trees:

- Each node must combine impacts of the $\pi$-messages obtained from parents.
- Each node should distribute a separate $\lambda$-message to each of its parents.
Belief propagation in polytrees
Belief propagation in polytrees

Let $U_1, \ldots, U_k$ be the parents of $X$ and $e_{U_i}^+$ the evidence that is “connected” to $X$ via $U_i$.

$$
\pi(x) = P(x \mid e^+)
= P(x \mid e_{U_1}^+, \ldots, e_{U_k}^+)
= \sum_{u_1,\ldots,u_k} P(x \mid u_1,\ldots,u_k)P(u_1,\ldots,u_k \mid e_{U_1}^+, \ldots, e_{U_k}^+)
= \sum_{u_1,\ldots,u_k} P(x \mid u_1,\ldots,u_k)P(u_1 \mid e_{U_1}^+) \cdots P(u_k \mid e_{U_k}^+)
= \sum_{u_1,\ldots,u_k} P(x \mid u_1,\ldots,u_k) \prod_{i=1}^k \pi_X(u_i)
$$
Belief propagation in polytrees

Consider the \( \lambda \)-message to be sent from \( X \) to \( U_i \). Denote the set of all other parents of \( X \) with \( V \) and \( v \) is a instantiation of variables in \( V \).

\[
\lambda_X(u_i) = P(e_V^+, e_X^- | u_i) = \sum_x \sum_v P(e_V^+, e_X^- | u_i, x, v)P(x, v | u_i) = \sum_x \sum_v P(e_X^- | x)P(e_V^+ | v)P(x, v | u_i)
\]

\[
\propto \sum_x P(e_X^- | x) \sum_v P(v | e_V^+) \frac{P(x | v, u_i)P(v | u_i)}{P(v)} = \sum_x P(e_X^- | x) \sum_v P(v | e_V^+)P(x | v, u_i)
\]

\[
= \sum_x \lambda(x) \prod_{v \neq i} \pi_X(v_k)P(x | v, u_i)
\]
Belief propagation in polytrees

- Initialize the network according to the evidence
- Repeat until convergence
  - For each node
    - Inspect $\pi$-messages from its parents and $\lambda$-messages from its children.
    - Update the beliefs.
    - Propagate the new messages to parents and children.
Example

- 5 Binary variables
- \( P(U = 0) = 0.4 \)
- \( P(W = 0) = 0.7 \)
- \( P(X = 0 \mid U = 0, W = 0) = 0.1, \)
  \( P(X = 0 \mid U = 0, W = 1) = 0.3, \)
  \( P(X = 0 \mid U = 1, W = 0) = 0.6, \)
  \( P(X = 0 \mid U = 1, W = 1) = 0.9 \)
- \( P(Y = 0 \mid X = 0) = 0.9, \ P(Y = 0 \mid X = 1) = 0.6 \)
- \( P(Z = 0 \mid X = 0) = 0.6, \ P(Z = 0 \mid X = 1) = 0.8 \)
- We observe that \( Z = 0 \). What are \( P(U \mid Z = 0), \ P(X \mid Z = 0), \)
  \( P(Y \mid Z = 0), \ P(W \mid Z = 0) \)?
Belief propagation

- Time complexity
  - Number of messages sent depends linearly on the diameter of the network
  - The time needed to compute a message is exponential with respect of the number of parents.

- (Conditional) Independence assumptions do not hold in multi-connected networks.
Further readings

- Belief propagation
  - Neapolitan (2004), Chapter 3.2
  - Pearl (1988), Chapters 4.2