Bayesian Networks: Belief Propagation in Singly Connected Networks

Huizhen Yu

janey.yu@cs.helsinki.fi
Dept. Computer Science, Univ. of Helsinki

Probabilistic Models, Spring, 2010
Outline

Belief Propagation

In Chains
In Trees
In Singly Connected Networks
Form of Evidence and Notation

We denote evidence (a finding) of $X = \{X_v, v \in V\}$ by $e$.

- Formally, we think of $e$ as a function of $x$ taking values in $\{0, 1\}$, representing a statement that some elements of $x$ are impossible, i.e.,

$$\{x \mid e(x) = 1\}$$

is the set of possible values of $x$ based on the evidence $e$. We also refer to this event as $e$.

- We consider $e$ that can be written in the factor form

$$e(x) = \prod_{v \in V} \ell_v(x_v), \quad \text{where } \ell_v(x_v) \in \{0, 1\}.$$

- For $A \subseteq V$, we use $e_A$ to denote the partial evidence of $X_A$:

$$e_A(x_A) = \prod_{v \in A} \ell_v(x_v).$$

Other short-hand notation we will use: $p(x_A \& e) = P(X_A = x_A, e)$,

$$p(x_A \& e_A \mid x_B) = P(X_A = x_A, e_A \mid X_B = x_B) = P(X_A = x_A \mid X_B = x_B) \cdot e_A(x_A),$$

and $p(x_A \mid e)$ denotes the conditional PMF of $X_A$ given the event $e$. 
Motivation

Inference tasks we consider here: calculate $p(x_v \mid e), \forall x_v$ and $P(e)$ for $P$ that is directed Markov w.r.t. a DAG $G$.

- Note that if we know $P(e)$, then we can calculate the posterior probability of a single $x$ given $e$ easily:

$$p(x \mid e) = p(x \& e)/P(e) = \left( \prod_{v \in V} p(x_v \mid x_{pa(v)}) \ell_v(x_v) \right) / P(e).$$

- Since $P(X = x, e) = p(x) e(x)$, in principle we can calculate

$$P(X_v = x_v, e) = \sum_{x_{V \setminus \{v\}}} P(X_{V \setminus \{v\}} = x_{V \setminus \{v\}}, X_v = x_v, e),$$

$$P(e) = \sum_x P(X = x, e).$$

But such calculation is not easy in most problems when $|V|$ is large.

The function $p(x_v \mid e)$ is referred to as the belief of $x_v$. 
Features of the Algorithms to be Introduced

In the algorithms to be introduced, the DAG $G$ is treated also as the architecture for distributed computation:

- Nodes: associated with autonomous processors
- Edges: communication links between processors

The independence relations represented by the DAG are exploited to separate the total evidence into pieces and streamline the computation.

The algorithms have performance guarantee on DAGs with simple structures – $G$ has no loops. But they have also been used successfully as approximate inference algorithms on loopy graphs.
Outline

Belief Propagation

In Chains

In Trees

In Singly Connected Networks
Evidence Structure in a Chain

Suppose $G$ is a chain. Consider a vertex $v$ with parent $u$ and child $w$:

We write $e$ as three pieces of evidence, $e = (e_{u+}, e_v, e_{v-})$, where

- $e_{u+}$: partial evidence of $X_{an(v)}$
- $e_v$: partial evidence of $X_v$
- $e_{v-}$: partial evidence of $X_{de(v)}$

We want to compute $p(x_v & e) = P(X_v = x_v, e)$ for all $x_v$. Since

$$P(X_{an(v)}, X_v, X_{de(v)}) = P(X_{an(v)}) \cdot P(X_v | X_u) \cdot P(X_{de(v)} | X_v),$$

we have

$$p((x_u, x_v) & e) = p(x_u & e_{u+}) \cdot p(x_v & e_v | x_u) \cdot p(e_{v-} | x_v).$$

If $v$ can get the first and third terms from $u$ and $w$ respectively, then $v$ can calculate its marginal $p(x_v & e)$ by summing over $x_u$. 

Huizhen Yu (U.H.)
Bayesian Networks: Belief Propagation in Singly Connected Networks
Feb. 16 7 / 25
Message Passing in a Chain

If node $v$ receives

- from parent $u$ the probabilities of $x_u$ and partial evidence $e_{u^+}$ on $u$'s side:
  \[ \pi_{u,v}(x_u) = p(x_u \& e_{u^+}), \forall x_u; \]
- from child $w$ the likelihoods of $x_v$ based on the partial evidence $e_{v^-}$ on $w$'s side:
  \[ \lambda_{w,v}(x_v) = p(e_{v^-} \mid x_v), \forall x_v, \]

then node $v$ can calculate

\[ p(x_v \& e) = \sum_{x_u} \pi_{u,v}(x_u) \cdot p(x_v \mid x_u) \ell_v(x_v) \cdot \lambda_{w,v}(x_v), \forall x_v. \]

What $u$ and $w$ need from $v$ in order to calculate their marginal probabilities?

- Parent $u$ needs for all $x_u$, the likelihood of $x_u$ based on $e_{u^-} = (e_v, e_{v^-})$:
  \[ \lambda_{v,u}(x_u) = p(e_{u^-} \mid x_u) = \sum_{x_v} p(x_v \& e_v \mid x_u) \cdot p(e_{v^-} \mid x_v) \]
  \[ = \sum_{x_v} p(x_v \mid x_u) \ell_v(x_v) \cdot \lambda_{w,v}(x_v). \]

- Child $w$ needs for all $x_v$, the probability of $x_v$ and $e_{v^+} = (e_{u^+}, e_v)$:
  \[ \pi_{v,w}(x_v) = p(x_v \& e_{v^+}) = \sum_{x_u} \pi_{u,v}(x_u) \cdot p(x_v \mid x_u) \ell_v(x_v). \]
Algorithm Summary

\begin{align*}
\lambda\text{-messages} & \quad (\text{likelihoods}) \\
\pi\text{-messages} & \quad (\text{probabilities})
\end{align*}

Each node \( v \)
- when receiving the message \( \lambda_{w,v} \) from its child, sends to its parent \( u \)

\[
\lambda_{v,u}(x_u) = \sum_{x_v} p(x_v | x_u) \ell_v(x_v) \cdot \lambda_{w,v}(x_v), \quad \forall x_u;
\]

- when receiving the message \( \pi_{u,v} \) from its parent, sends to its child \( w \)

\[
\pi_{v,w}(x_v) = \sum_{x_u} \pi_{u,v}(x_u) \cdot p(x_v | x_u) \ell_v(x_v), \quad \forall x_v;
\]

- when receiving both messages, calculates

\[
p(x_v \& e) = \sum_{x_u} \pi_{u,v}(x_u) \cdot p(x_v | x_u) \ell_v(x_v) \cdot \lambda_{w,v}(x_v), \quad \forall x_v,
\]

\[
P(e) = \sum_{x_v} p(x_v \& e), \quad p(x_v | e) = p(x_v \& e) / P(e).
\]
Outline

Belief Propagation

In Chains

In Trees

In Singly Connected Networks
Evidence Structure in a Rooted Tree

Suppose $G$ is a rooted tree. Then $G^m = G^\sim$.

Consider a vertex $v$ with parent $u$ and children $w_1, \ldots, w_m$:

We write the total evidence $e$ as several pieces of evidence,

$$e = (e_{\text{nd}(v)}, e_v, e_{T_{w_1}}, \ldots, e_{T_{w_m}}),$$

where

- $e_{\text{nd}(v)}$: partial evidence of $X_{\text{nd}(v)}$
- $e_v$: partial evidence of $X_v$
- $e_{T_w}, w \in \text{ch}(v)$: partial evidence of the variables associated with the subtree $T_w$ rooted at $w$, i.e., $X_{\{w\} \cup \text{de}(w)}$

Since

$$P(X_{\text{nd}(v)}, X_v, X_{\text{de}(v)}) = P(X_{\text{nd}(v)}) \cdot P(X_v | X_u) \cdot \prod_{w \in \text{ch}(v)} P(X_{T_w} | X_v),$$

$$p((x_u, x_v) & e) = p(x_u & e_{\text{nd}(v)}) \cdot p(x_v & e_v | x_u) \cdot \prod_{w \in \text{ch}(v)} p(e_{T_w} | x_v).$$
Message Passing in a Rooted Tree

From

\[ p((x_u, x_v) \& e) = p(x_u \& e_{\text{nd}(v)}) \cdot p(x_v \& e_v | x_u) \cdot \prod_{w \in \text{ch}(v)} p(e_{T_w} | x_v). \]

we see that if \( v \) receives

- from parent \( u \) the probabilities of \( x_u \) and evidence \( e_{\text{nd}(v)} \) for all \( x_u \):
  \[
  \pi_{u,v}(x_u) = p(x_u \& e_{\text{nd}(v)}), \quad \forall x_u;
  \]

- from every child \( w \) the likelihoods of all \( x_v \) based on the evidence \( e_{T_w} \):
  \[
  \lambda_{w,v}(x_v) = p(e_{T_w} | x_v), \quad \forall x_v,
  \]

then node \( v \) can calculate

\[
 p(x_v \& e) = \sum_{x_u} \pi_{u,v}(x_u) \cdot p(x_v | x_u) \ell_v(x_v) \cdot \prod_{w \in \text{ch}(v)} \lambda_{w,v}(x_v).
\]
Message Passing in a Rooted Tree

What do nodes $u$ and $w$ need from $v$ in order to calculate their marginals?

- Parent $u$ needs the likelihoods of $x_u$ based on $e_T$ for all $x_u$:

$$\lambda_{v,u}(x_u) = \sum_{x_v} p(x_v \& e_v \mid x_u) \cdot \prod_{w \in \text{ch}(v)} p(e_{T_w} \mid x_v)$$

$$= \sum_{x_v} p(x_v \mid x_u) \ell_v(x_v) \cdot \prod_{w \in \text{ch}(v)} \lambda_{w,v}(x_v).$$

- Child $w$ needs for all $x_v$, the probability of $x_v$ and

$$\pi_{v,w}(x_v) = p(x_v \& e_{\text{nd}(w)})$$

$$= \left( \sum_{x_u} p(x_u \& e_{\text{nd}(v)}) \cdot p(x_v \& e_v \mid x_u) \right) \cdot \prod_{w' \in \text{ch}(v) \setminus \{w\}} p(e_{T_{w'}} \mid x_v)$$

$$= \left( \sum_{x_u} \pi_{u,v}(x_u) \cdot p(x_v \mid x_u) \ell_v(x_v) \right) \cdot \prod_{w' \in \text{ch}(v) \setminus \{w\}} \lambda_{w',v}(x_v).$$
Algorithm Summary

Each node $v$

- sends to its parent $u$
  \[ \lambda_{v,u}(x_u) = \sum_{x_v} p(x_v \mid x_u) \ell_v(x_v) \cdot \prod_{w \in \text{ch}(v)} \lambda_{w,v}(x_v), \quad \forall x_u; \]

- sends to its child $w$
  \[ \pi_{v,w}(x_v) = \left( \sum_{x_u} \pi_{u,v}(x_u) \cdot p(x_v \mid x_u) \ell_v(x_v) \right) \cdot \prod_{w' \in \text{ch}(v) \setminus \{w\}} \lambda_{w',v}(x_v), \quad \forall x_v; \]

- when receiving all messages, calculates
  \[ p(x_v \& e) = \left( \sum_{x_u} \pi_{u,v}(x_u) \cdot p(x_v \mid x_u) \ell_v(x_v) \right) \cdot \prod_{w \in \text{ch}(v)} \lambda_{w,v}(x_v), \quad \forall x_v, \]
  \[ P(e) = \sum_{x_v} p(x_v \& e), \quad p(x_v \mid e) = p(x_v \& e) / P(e). \]

Message passing schemes:

(i) Each node can send a message to a linked node if it has received messages from all the other linked nodes.

(ii) Each node can send updated messages to linked nodes whenever it gets a new message from some node.
Illustration of Parallel Updating

From J. Peal's book, 1988:
At time 0, each node of the tree has calculated its own marginal.
At time 1, two new pieces of evidence arrive and trigger new messages.
After time 5, all nodes have updated their marginals incorporating the new evidence.
Outline

Belief Propagation

- In Chains
- In Trees
- In Singly Connected Networks
Definition of a Singly Connected Network

Definition: a DAG $G$ is *singly connected*, if its undirected version $G\sim$ is a tree. Such a $G$ is also called a *polytree*.

In a polytree $G$:

- Each node can have multiple parents and children.
- But there is only one trail between each pair of nodes.

Consider a vertex $v$ with parents $u_1, \ldots, u_n$ and children $w_1, \ldots, w_m$. When $v$ is viewed as the center, the branch of the polytree containing one of its parents or children is a sub-polytree. Denote

- $T_{vu_i}, i = 1, \ldots, n$: the sub-polytree containing the node $u_i$, resulting from removing the edge $(u_i, v)$;
- $T_{vw_i}, i = 1, \ldots, m$: the sub-polytree containing the node $w_i$, resulting from removing the edge $(v, w_i)$.
Evidence Structure in a Singly Connected Network

For a sub-polytree $T$, denote

- $X_T$: the variables associated with nodes in $T$
- $e_T$: the partial evidence of $X_T$

We have

$$P(X_{T_{vu_1}}, \ldots, X_{T_{vu_n}}) = P(X_{T_{vu_1}}) \cdots P(X_{T_{vu_n}}),$$

and

$$P(X_{T_{vw_1}}, \ldots, X_{T_{vw_m}} \mid X_v) = P(X_{T_{vw_1}} \mid X_v) \cdots P(X_{T_{vw_m}} \mid X_v).$$

(Why? We may argue this using (DG) or d-separation – the latter is also simple in this case because there is only one trail between each pair of nodes.)

Therefore,

$$p\left((x_{pa(v)}, x_v) \& e\right) = \left( \prod_{u \in pa(v)} p(x_u \& e_{T_{vu}}) \right) \cdot p(x_v \& e_v \mid x_{pa(v)}) \cdot \prod_{w \in ch(v)} p(e_{T_{vw}} \mid x_v).$$
Message Passing in a Singly Connected Network

From

\[ p((x_{pa(v)}, x_v) \& e) = \left( \prod_{u \in pa(v)} p(x_u \& e_{T_{vu}}) \right) \cdot p(x_v \& e_v | x_{pa(v)}) \cdot \prod_{w \in ch(v)} p(e_{T_{vw}} | x_v). \]

we see that \( v \) can calculate its marginal if it receives messages

- \( \pi_{u,v} \) from all parents, where
  \[ \pi_{u,v}(x_u) = p(x_u \& e_{T_{vu}}), \ \forall x_u; \]
- and \( \lambda_{w,v} \) from all children, where
  \[ \lambda_{w,v}(x_v) = p(e_{T_{vw}} | x_v), \ \forall x_v. \]

Then, \( p(x_v \& e) \) is given by

\[ p(x_v \& e) = \left( \sum_{x_{pa(v)}} \prod_{u \in pa(v)} \pi_{u,v}(x_u) \cdot p(x_v | x_{pa(v)}) \ell_v(x_v) \right) \cdot \prod_{w \in ch(v)} \lambda_{w,v}(x_v). \]
Message Passing in a Singly Connected Network

What do parents need from \( v \) in order to calculate their marginals?

- A parent \( u \) needs the likelihoods of all \( x_u \) based on the partial evidence \( e_{T_{uv}} \) from the sub-polytree on \( v \)'s side with respect to \( u \):

\[
\lambda_{v,u}(x_u) = \sum_{x_v} \sum_{x_{\text{pa}(v)} \setminus \{u\}} p(x_v & e_v | x_{\text{pa}(v)}) \cdot \prod_{u' \in \text{pa}(v) \setminus \{u\}} p(x_{u'} & e_{T_{vu'}}) \cdot \prod_{w \in \text{ch}(v)} p(e_{T_{vw}} | x_v) \\
= \sum_{x_v} \left( \sum_{x_{\text{pa}(v)} \setminus \{u\}} p(x_v | x_{\text{pa}(v)}) \ell_v(x_v) \cdot \prod_{u' \in \text{pa}(v) \setminus \{u\}} \pi_{u',v}(x_{u'}) \right) \cdot \prod_{w \in \text{ch}(v)} \lambda_{w,v}(x_v).
\]
Message Passing in a Singly Connected Network

What do children need from \( v \) in order to calculate their marginals?

- A child \( w \) needs for all \( x_v \), the probability of \( x_v \) and the partial evidence \( e_{T_{wv}} \) from the sub-polytree on \( v \)’s side with respect to \( w \):

\[
p(x_v & e_{T_{wv}}) = \sum_{x_{pa(v)}} p(x_v & e_v | x_{pa(v)}) \cdot \prod_{u \in pa(v)} p(x_u & e_{T_{vu}}) \prod_{w' \in ch(v) \setminus \{w\}} p(e_{T_{vw'}} | x_v) \\
= \left( \sum_{x_{pa(v)}} p(x_v | x_{pa(v)}) \ell_v(x_v) \cdot \prod_{u \in pa(v)} \pi_{u,v}(x_u) \right) \cdot \prod_{w' \in ch(v) \setminus \{w\}} \lambda_{w',v}(x_v).
\]

\[
\begin{align*}
\text{Belief Propagation in Trees} & , \\
\text{Belief Propagation in Chains} & , \\
\text{Loopy Belief Propagation} & .
\end{align*}
\]
Algorithm Summary

Each node $v$

- sends to each $u$ of its parents

$$\lambda_{v,u}(x_u) = \sum_{x_v} \sum_{x_{pa(v)} \setminus \{u\}} p(x_v | x_{pa(v)}) \ell_v(x_v) \cdot \prod_{u' \in pa(v) \setminus \{u\}} \pi_{u',v}(x_{u'}) \cdot \prod_{w \in ch(v)} \lambda_{w,v}(x_v), \forall x_u;$$

- sends to each $w$ of its children

$$\pi_{v,w}(x_v) = \prod_{w' \in ch(v) \setminus \{w\}} \lambda_{w',v}(x_v) \cdot \sum_{x_{pa(v)}} p(x_v | x_{pa(v)}) \ell_v(x_v) \cdot \prod_{u \in pa(v)} \pi_{u,v}(x_u), \forall x_v;$$

- when receiving all messages from parents and children, calculates

$$p(x_v \& e) = \left( \prod_{w \in ch(v)} \lambda_{w,v}(x_v) \right) \cdot \sum_{x_{pa(v)}} \prod_{u \in pa(v)} \pi_{u,v}(x_u) \cdot p(x_v | x_{pa(v)}) \ell_v(x_v), \forall x_v,$$

$$P(e) = \sum_{x_v} p(x_v \& e), \quad p(x_v | e) = p(x_v \& e) / P(e).$$

Message passing schemes:

1. Each node can send a message to a linked node if it has received messages from all the other linked nodes.

2. Each node can send updated messages to linked nodes whenever it gets a new message from some node.
Example of Noisy-Or Gate

\[ x_i, y_i, y \in \{0, 1\}. \]

\[ P(X_i = 1) = p_i, \]

\[ P(Y_i = 1 \mid X_i = 0) = 0, \]

\[ P(Y_i = 1 \mid X_i = 1) = 1 - q_i. \]

\[ p(y \mid x_1, \ldots, x_n) = \begin{cases} 
\prod_{i : x_i = 1} q_i, & \text{if } y = 0; \\
1 - \prod_{i : x_i = 1} q_i, & \text{if } y = 1.
\end{cases} \]

Express the message \( \pi_{X_i, Y_i}(x_i) \) in the vector form \([ \pi_{X_i, Y_i}(1), \pi_{X_i, Y_i}(0) ]\):

\[ \pi_{X_i, Y_i} = [ p_i, 1 - p_i ]. \]

Similarly, express \( \pi_{Y_i, Y}(y_i) \) as \([ \pi_{Y_i, Y}(1), \pi_{Y_i, Y}(0) ]\):

\[ \pi_{Y_i, Y}(y_i) = \sum_{x_i \in \{0, 1\}} \pi_{X_i, Y_i}(x_i)p(y_i \mid x_i), \quad \text{so } \pi_{Y_i, Y} = [ p_i(1 - q_i), p_iq_i + (1 - p_i) ]. \]
Example of Noisy-Or Gate

Suppose $e : \{ Y = 1 \}$ is received. Then, $Y$ sends a message
$
\lambda_{Y,Y_i} = [ \lambda_{Y,Y_i}(1), \lambda_{Y,Y_i}(0) ]
$
to each $Y_i$, where

$$
\lambda_{Y,Y_i}(y_i) = \sum_{k \neq i} \sum_{y_k \in \{0,1\}} p(1 | y_1, \ldots, y_n) \prod_{j \neq i} \pi_{Y_j,Y}(y_j).
$$

(What are these values?)

Subsequently, each $Y_i$ sends to $X_i$ the message $\lambda_{Y_i,X_i}(x_i)$:

$$
\lambda_{Y_i,X_i}(1) = (1 - q_i) \cdot \lambda_{Y,Y_i}(1) + q_i \cdot \lambda_{Y,Y_i}(0), \quad \lambda_{Y_i,X_i}(0) = \lambda_{Y,Y_i}(0).
$$

Each $X_i$ can calculate its marginal and posterior probability of $X_i = 1$ as

$$
P(X_i = 1, e) = P(X_i = 1) \cdot \lambda_{Y_i,X_i}(1),
$$

$$
P(X_i = 0, e) = P(X_i = 0) \cdot \lambda_{Y_i,X_i}(0),
$$

$$
P(X_i = 1 | e) = \frac{p_i \cdot \lambda_{Y_i,X_i}(1)}{p_i \cdot \lambda_{Y_i,X_i}(1) + (1 - p_i) \cdot \lambda_{Y_i,X_i}(0)}.
$$
Generalizations and Further Reading

• Find most probable configurations: \( \max_x p(x \& e) \)

• Conditioning:

  \( G \sim \) is not a tree. We condition on certain variables to create several singly connected networks and then fuse together the calculated results.

• Loopy belief propagation:

  \( G \sim \) is not a tree, but we apply the message passing algorithm any way. Algorithm variants and convergence analysis are active research topics.

Further reading: