A Variable Elimination Algorithm for Belief Networks

This handout describes a general algorithm for computing $P(X \mid E)$ in a belief network, where $X$ is a node and $E$ is the evidence (those nodes known to be true or false). The example is computing $P(R \mid W)$ for the network in Figure 14.11, p. 510. The algorithm (named elim-bel) comes from Rina Dechter, Bucket elimination: A unifying framework for probabilistic inference, Proceedings 1996 Conference on Uncertainty in Artificial Intelligence.

Conversion of Conditional Probability Tables to $\lambda$ Tables

Each conditional probability table is converted to a $\lambda$ (lambda) table. Each possible value assignment to the node and its parents are associated with a $\lambda$ value, which is initialized to the corresponding conditional probability.

<table>
<thead>
<tr>
<th>$C$</th>
<th>$P(S)$</th>
<th>$C$</th>
<th>$S$</th>
<th>$\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$0.1 = P(S \mid C)$</td>
<td>$T$</td>
<td>$T$</td>
<td>$0.1 = P(S \mid C)$</td>
</tr>
<tr>
<td>$F$</td>
<td>$0.5 = P(S \mid \neg C)$</td>
<td>$T$</td>
<td>$F$</td>
<td>$0.9 = P(\neg S \mid C)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$F$</td>
<td>$T$</td>
<td>$0.5 = P(S \mid \neg C)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$F$</td>
<td>$F$</td>
<td>$0.5 = P(\neg S \mid \neg C)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$C$</th>
<th>$P(R)$</th>
<th>$C$</th>
<th>$R$</th>
<th>$\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$0.8 = P(R \mid C)$</td>
<td>$T$</td>
<td>$T$</td>
<td>$0.8 = P(R \mid C)$</td>
</tr>
<tr>
<td>$F$</td>
<td>$0.2 = P(R \mid \neg C)$</td>
<td>$T$</td>
<td>$F$</td>
<td>$0.2 = P(\neg R \mid C)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$F$</td>
<td>$T$</td>
<td>$0.2 = P(R \mid \neg C)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$F$</td>
<td>$F$</td>
<td>$0.8 = P(\neg R \mid \neg C)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$S$</th>
<th>$R$</th>
<th>$P(W)$</th>
<th>$S$</th>
<th>$R$</th>
<th>$W$</th>
<th>$\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$0.99 = P(W \mid S, R)$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$0.99 = P(W \mid S, R)$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$0.90 = P(W \mid S, \neg R)$</td>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
<td>$0.90 = P(W \mid S, \neg R)$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$0.90 = P(W \mid \neg S, R)$</td>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
<td>$0.10 = P(\neg W \mid S, \neg R)$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$0.00 = P(W \mid \neg S, \neg R)$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$0.10 = P(\neg W \mid \neg S, \neg R)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$0.00 = P(W \mid \neg S, \neg R)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$F$</td>
<td>$F$</td>
<td>$1.00 = P(\neg W \mid \neg S, \neg R)$</td>
</tr>
</tbody>
</table>

Eliminating Evidence Variables

It is given that $W$ is true. In each $\lambda$ table that contains $W$, we delete all rows that have $W$ as false, and eliminate the $W$ column. [If $W$ was given as false, then we would delete the rows that had $W$ as true.]
Eliminating Other Variables

Now we want to eliminate $S$ and $C$, leaving only $R$, the variable that we are interested in. First, we eliminate $S$. [Order of elimination does not affect correctness.]

1. First find all the $\lambda$ tables that reference $S$. This includes the $S, R \lambda$ table just created by evidence elimination, and the $C, S \lambda$ table generated earlier.

2. These tables also reference $C$ and $R$. The new $\lambda$ table is for these two variables.

3. To compute an entry in the new $C, R \lambda$ table (say $C$ false and $R$ true), we add two values. One value is the product of the $\lambda$ values that are consistent with $C$ false, $R$ true, and $S$ true. The other value is the product of the $\lambda$ values that are consistent with $C$ false, $R$ true, and $S$ false.

\[
\begin{array}{c|c|c|c}
C & S & \lambda & C & R & \lambda \\
\hline
T & T & 0.1 & T & T & 0.99 \\
T & F & 0.9 & T & F & 0.90 \\
F & T & 0.5 & F & T & 0.90 \\
F & F & 0.5 & F & F & 0.00 \\
\end{array}
\]

To eliminate $C$, there is one $C \lambda$ table and two $C, R \lambda$ tables to combine.

\[
\begin{array}{c|c|c|c|c|c|c}
C & \lambda & C & R & \lambda & C & R & \lambda \\
\hline
T & 0.5 & T & T & 0.8 & T & T & 0.909 \\
F & 0.5 & T & F & 0.2 & T & F & 0.090 \\
F & 0.5 & F & T & 0.2 & F & T & 0.945 \\
F & 0.5 & F & F & 0.8 & F & F & 0.450 \\
\end{array}
\Rightarrow
\begin{array}{c|c|c|c|c|c|c}
R & \lambda \\
\hline
T & 0.4581 = 0.5(0.8)(0.909) + 0.5(0.2)(0.945) \\
F & 0.1890 = 0.5(0.2)(0.090) + 0.5(0.8)(0.450) \\
\end{array}
\]

Now, we have one $\lambda$ table for $R$ alone, and we can compute $P(R \mid W)$ by:

\[
P(R \mid W) = \frac{0.4581}{0.4581 + 0.1890} \approx 0.7079
\]

If we ended up with more than one $\lambda$ table for $R$, we would compute one value $p$ by multiplying all the $\lambda$ values for $R$ as true, compute a second value $q$ by multiplying all the $\lambda$ values for $R$ as false, and then perform $p/(p + q)$.  

2
Pseudocode for Elim-Bel

In this pseudocode, I assume the existence of the following subroutines. VARIABLES(a) returns the set of variables in a, which might be a belief network, a value assignment, or a lambda table. PARENTS(x) returns the set of the parents of the variable x. ASSIGNMENTS(S) returns all possible value assignments to a set of variables S; if S has n variables, then ASSIGNMENTS(S) returns $2^n$ assignments. $P(x \mid v)$ looks up the probability that x is true given value assignment v in the probability table for x. LAMBDA(v, ltable) returns the value in the lambda table ltable consistent with value assignment v; any variables in v not mentioned by ltable are ignored.

Elim-Bel determines the probability that x is true given value assignment v. v does not assign a value to x. v can be a partial value assignment.

**function** Elim-Bel(x, v)

"/* Create lambda tables. */"

ltables ← ∅

for each y in VARIABLES(belief net) do
  S ← PARENTS(y)
  ltable ← new lambda table
  for each u in ASSIGNMENTS(S)
    LAMBDA(u ∪ {y = true}, ltable) ← $P(y \mid u)$
    LAMBDA(u ∪ {y = false}, ltable) ← 1 − $P(y \mid u)$
  end for
  Add ltable to ltables

end for

" /* Eliminate evidence variables. */"

for each ltable in ltables do
  Remove all LAMBDA(u, ltable) values where u is inconsistent with v
  Remove VARIABLES(v) from ltable

end for
/* Eliminate other variables. */
for each y in VARIABLES(ltables) − {x} do
    ytables ← subset of ltables that refer to y
    S ← VARIABLES(ytables) − {y}
ltable ← new lambda table
    for each u in ASSIGNMENTS(S) do
        ytrue ← 1
        yfalse ← 1
        for each ytable in ytables do
            ytrue ← ytrue * LAMBDA(u ∪ {y = true}, ytable)
            yfalse ← yfalse * LAMBDA(u ∪ {y = false}, ytable)
        end for
        LAMBDA(u, ltable) ← ytrue + yfalse
    end for
    Remove ytables from ltables
    Add ltable to ltables
end for

/* Calculate conditional probability. */
xtrue ← 1
xfalse ← 1
for each ltable in ltables do
    xtrue ← xtrue * LAMBDA({x = true}, ltable)
    xfalse ← xfalse * LAMBDA({x = false}, ltable)
end for
return xtrue/(xtrue + xfalse)