The Elimination Algorithm

Chris Williams

School of Informatics, University of Edinburgh

October 2009

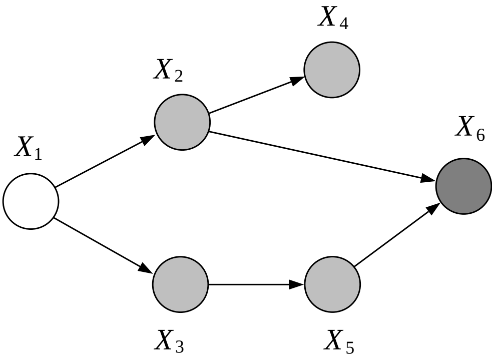
Overview

- Inference in Belief Networks
- Elimination in an Example Network
- The Elimination Algorithm
- Evidence Potentials
- Graph Elimination
- Reconstituted Graph
- Reading: Jordan chapter 3

Inference in Belief Networks

- Partition the random variables $\mathbf{x}$ into three disjoint subsets $\mathbf{x}_E$, $\mathbf{x}_F$ and $\mathbf{x}_R$. We wish to compute the posterior $p(\mathbf{x}_F|\mathbf{x}_E)$ over the query nodes $\mathbf{x}_F$
- This involves conditioning on the evidence nodes $\mathbf{x}_E$ and summing out (integrating out) the hidden nodes $\mathbf{x}_R$
- If the joint distribution is simply a huge table this is trivial: select the appropriate indices in the columns corresponding to $\mathbf{x}_E$, sum over the columns corresponding to $\mathbf{x}_R$, and renormalize the resulting table over $\mathbf{x}_F$
- But what if the distribution is represented by a directed graphical model?

An Example Network



$$p(x_1|\overline{x}_6) = p(x_1, \overline{x}_6)/p(\overline{x}_6) = p(x_1, \overline{x}_6)/\sum_{x_1'} p(x_1', \overline{x}_6)$$

$$
\begin{aligned}
p(x_1, \bar{x}_6) &= \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} p(x_1) p(x_2|x_1) p(x_3|x_1) p(x_4|x_2) p(x_5|x_3) p(\bar{x}_6|x_2, x_5) \\
&= p(x_1) \sum_{x_2} p(x_2|x_1) \sum_{x_3} p(x_3|x_1) \sum_{x_4} p(x_4|x_2) \sum_{x_5} p(x_5|x_3) p(\bar{x}_6|x_2, x_5) \\
&= p(x_1) \sum_{x_2} p(x_2|x_1) \sum_{x_3} p(x_3|x_1) m_5(x_2, x_3) \sum_{x_4} p(x_4|x_2) \\
&= p(x_1) \sum_{x_2} p(x_2|x_1) m_4(x_2) \sum_{x_3} p(x_3|x_1) m_5(x_2, x_3) \\
&= p(x_1) \sum_{x_2} p(x_2|x_1) m_4(x_2) m_3(x_1, x_2) \\
&= p(x_1)\ m_2(x_1)
\end{aligned}
$$

using elimination ordering $(6, 5, 4, 3, 2, 1)$

Notes on the example

- $\overline{x}_6$ means that x_6 is fixed to a *specific* value
- $m_5(x_2, x_3) = \sum_{x_5} p(x_5|x_3)p(\overline{x}_6|x_2, x_5)$ etc
- Note that $m_4(x_2) = 1$; why?
- Key idea 1: push sums inside products
- Key idea 2: cache subexpressions

Evidence Potentials

- Elimination uses a book-keeping trick, evidential potentials

$$g(\overline{x}_i) = \sum_{x_i} g(x_i)\delta(x_i, \overline{x}_i)$$

- This trick allows us to treat conditioning in the same way as marginalization

Elimination Algorithm, part I

ELIMINATE(G,E,F)
    INITIALIZE(G,F)
    EVIDENCE(E)
    UPDATE(G)
    NORMALIZE(F)

INITIALIZE(G,F)
    choose an ordering O such that F appears last
    for each node X_i in V
        place potential $p(x_i|parents_i)$ on the active list
    end for

Elimination Algorithm, part II

EVIDENCE(E)
for each i in E
place potential $\delta(x_i, \overline{x}_i)$ on the active list
end for

UPDATE(G)
for each i in O
find all potentials in the active list that reference x_i
and remove them from the active list
Let $\phi_i(x_{T_i})$ denote the product of these potentials
Let $m_i(x_{S_i}) = \sum_{x_i} \phi_i(x_{T_i})$
Place $m_i(x_{S_i})$ on the active list
end for

NORMALIZE(F)
$p(x_F | \overline{x}_E) \leftarrow \phi_F(x_F) / \sum_{x_F} \phi_F(x_F)$

Graph Elimination

Consider first *undirected* graphs with

$$p^E(x) = \frac{1}{Z} \prod_C \psi^E_{X_C}(x_C)$$

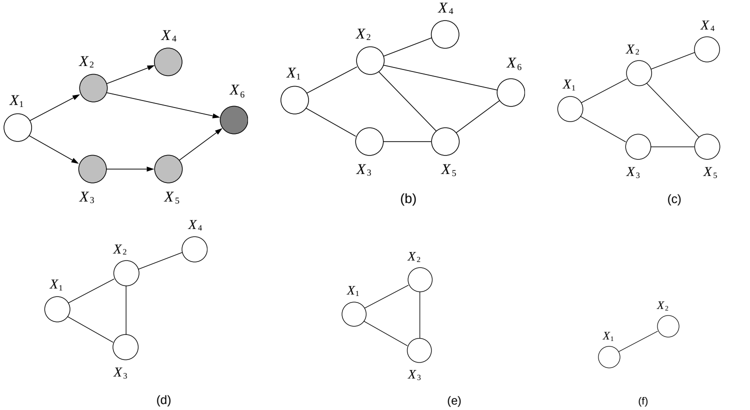
where the product is over cliques

- Start with an elimination ordering O
- At each step the algorithm eliminates the next node in O, where “eliminate” means removing the node from the graph and connecting the (remaining) neighbours of the node

Moralization

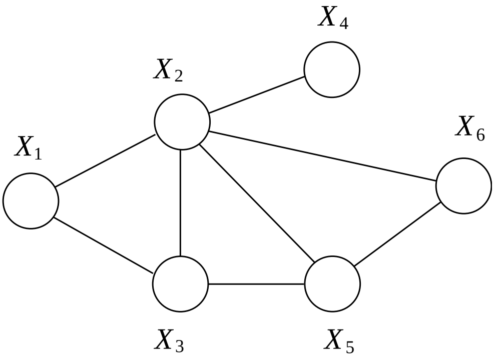
- There is one additional complexity for directed graphs: parents may not be explicitly connected, but are involved in the same potential function
- Thus to think of the ELIMINATION algorithm as node removal, we must first connect all the parents of every node and drop the directions of the links: this is known as "moralization"

Graphically the stages for the example are:



Reconstituted Graph

- The *reconstituted* graph is the graph whose edge set includes all the original edges as well as any new edges created during elimination
- In fact, the reconstituted graph is a *triangulated* graph, see forthcoming lecture on the junction tree algorithm. Elimination is a simple algorithm for triangulating a graph



The *reconstituted* graph; $X_2 - X_5$ is added via moralization, and $X_2 - X_3$ is added when eliminating X_5