Variable elimination is probably the single most important algorithm to understand for manipulating graphical models, because

- it solves the most important problem, “inference”, computing the conditional distribution over one set of variables given assignments to another;
- it works for any directed or undirected graphical model over discrete variables;
- it’s easy to understand and re-derive;
- with the right elimination order, it’s as efficient as any other method that works for arbitrary probability tables (that is, that relies only on conditional independencies and not the numeric values of the probability distribution).

1 Moralization

VE can be done on either directed or undirected graphs but it’s convenient to consider only the undirected (Markov random field) case, and for problems on directed graphs (Bayes’ nets), convert the BN net to an MRF prior to elimination. The conversion process is called moralization, since it involves the “marriage” of a child’s parents.

For a directed graphical model, let $G$ be the conditional independence graph and $P$ the probability distribution. Let $G^m$ and $P'$ be the graph and distribution after moralization. Moralization exactly maintains the probability distribution: $P' = P$, and $G^m$ remains an I-map of $P$. But some of the conditional independencies encoded in $G$ may be missing from $G^m$. These are the independencies corresponding to the moralization edges; as you recall, adding an edge will destroy at least one independence in a graph. Thus, the moral graph is a less accurate description of the independencies in $P$ than the original graph.
Given the directed graph $G$, we know that $P$ factors

$$P(x) = \prod_{x_i \in V} P_{X_i}(x_i|\text{pa}(x_i)).$$

And we know that $P'$ must factor

$$P'(x) = \prod_i \phi_i(x).$$

The correspondence $P'(x) = P(x)$ is achieved by creating a potential for every family in $G$:

$$\phi_{X_i}(x) = P_{X_i}(x_i|\text{pa}(x_i)).$$

By the Hammersley-Clifford theorem, $G^m$ is an I-map of $P$ if the domain of every potential function is a (not necessarily maximal) clique in $G^m$, or equivalently, if every family in $G$ is a clique in $G^m$. Thus, two equivalent methods of constructing the edges for $G^m = (V, E^m)$:

- $E^m$ is the union of cliques over all families of $G$;
- $E^m$ is $E$ after adding edges between the parents of each variable (*marrying* the parents to create cliques).

$G^m$ may contain cliques that do not correspond to families of $G$; there are no corresponding potentials in the factorization of $P'$.

Here’s an example of moralization:

\[
\begin{align*}
\text{Original Graph} & : G & \text{Moralized Graph} & : G^m \\
A & \rightarrow B & \overset{E}{\rightarrow} C & \rightarrow D & \rightarrow E & \rightarrow F & \rightarrow G \\
A & \rightarrow B & \rightarrow C & \rightarrow D & \rightarrow E & \rightarrow F & \rightarrow G \\
\end{align*}
\]

\[
\begin{align*}
&= P_B P_C P_E P_F P_G \\
&\quad P_{D|EFG} P_{A|BCD} \\
&= \phi_B \phi_C \phi_E \phi_F \phi_G \\
&\quad \phi_{DEFG} \phi_{ABCD}
\end{align*}
\]
Notice that $G^n$ has lost numerous conditional independencies: $E \perp F, F \perp G, A \perp C$, etc.

## 2 Elimination Algorithm

Variable elimination works by eliminating variables from an undirected graphical model one-by-one, until only the variables of interest are left. As each (unobserved) variable $X_i$ is eliminated, the set of potentials involving $X_i$ is replaced by a single new potential $\Psi_{X_i}$ constructed by marginalizing over $X_i$. Graphically, this corresponds to eliminating the variable and its edges, and fully connecting its neighbors - since with the elimination of $X_i$ they are generally no longer conditionally independent.

The input to variable elimination:

- $V$, a set of variables;
- $\Phi = \{\phi\}$, a set of potential functions over $V$;
- $O = \{\langle X_j = x_j \rangle\}$, observations of a subset of $V$ (possibly empty);
- $\pi$, the elimination order, an ordered subset of $V$; for simplicity, $\pi$ is assumed to include the observation variables $O$.

The output:

- $\phi_Z$, a table over $Z = V - \pi$, the set of variables not eliminated. Each entry $\phi_Z(z)$ is the joint probability of $Z = z$ and the observations, marginalized over the eliminated variables.

$$
\phi_Z = \text{ELIMINATE}(V, \Phi = \{\phi\}, \pi = X_1 \ldots X_m, O = \{\langle X_j = x_j \rangle\})
$$

for $X_i$ in $\pi$

$$
\Phi_i = \{\phi \in \Phi | X_i \in \text{domain}(\phi)\} // \text{potentials involving } X_i
$$

if $\langle X_i = v \rangle \in O$
// $X_i$ is observed: create new potentials by fixing $X_i$
for $\phi \in \Phi_i$
    $N = \text{domain}(\phi) \setminus \{X_i\}$
    $\phi' = \text{new potential over variables } N$
    for each config $x$ in $\Omega_N$
        $\phi'(x) = \phi(x, v)$
    $\Phi = \Phi \cup \{\phi'\}$
else
// compute new potential by marginalizing over $X_i$
    $\Omega = \bigcup_{\phi \in \Phi_i} \text{domain}(\phi) \setminus \{X_i\}$ // neighbors of $X_i$
    $\phi' = \text{new potential over variables } N$
    for each config $x$ in $\Omega$
        $\phi'(x) = \sum_{x_i \in X_i} \prod_{\phi \in \Phi_i} \phi(x, x_i)$
    $\Phi = \Phi \cup \{\phi'\}$

// eliminate $X_i$ and its potentials
$V = V \setminus \{X_i\}$
$\Phi = \Phi \setminus \Phi_i$

// compute table over remaining variables
$\phi_Z = \text{new potential over remaining variables } V$
for each config $x$ in $V$
    $\phi_Z(x) = \prod_{\psi \in \Phi} \phi(x)$
return $\phi_Z$
3 Example

3.1 An elimination that is valid for any potentials

The factorization $P = P_A P_B | A P_C | A P_D | AB P_E | BC P_F | E$ with directed graph above left, after moralization has factorization $P = \phi_{AB} \phi_{AC} \phi_{ABD} \phi_{BCE} \phi_{EF}$ and graph above right.

Suppose the goal is to compute $P_{CD|F=1}$; this is accomplished by first computing the joint $P_{CD,F=1}$ using VE, by eliminating $A$, $B$, $E$ and $F$. After choosing an elimination order (here, arbitrarily, $FEAB$), VE gives a table of over the remaining variables:

$$\phi_{CD} = \text{ELIMINATE}(\{ABCDEF\}, \{\phi_{AB} \phi_{AC} \phi_{ABD} \phi_{BCE} \phi_{EF}\}, \{FEAB\}, \{\langle F = 1 \rangle\})$$

and then

$$P(C = c, D = d | F = 1) = \frac{\phi_{CD}(c, d)}{\sum_{cd} \phi_{CD}(c, d)}.$$

The intermediate stages of variable elimination are as follows:
<table>
<thead>
<tr>
<th>Var.</th>
<th>Neigh</th>
<th>New factorization</th>
<th>New graph</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( P = \phi_A \phi_{AB} \phi_{AC} \phi_{ABD} \phi_{BCE} \phi_{EF} )</td>
<td><img src="image1" alt="Diagram" /></td>
</tr>
<tr>
<td>F</td>
<td>E</td>
<td>( P = \phi_A \phi_{AB} \phi_{AC} \phi_{ABD} \phi_{BCE} \phi^1_E )</td>
<td><img src="image2" alt="Diagram" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \phi^1_E(e) = \phi_{EF}(e, 1) )</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>B, C</td>
<td>( P = \phi_A \phi_{AB} \phi_{AC} \phi_{ABD} \phi_{BC} \phi_{E} )</td>
<td><img src="image3" alt="Diagram" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \phi^2_{BC}(bc) = \sum_e \phi_{BCE}(bce) \phi^1_E(e) )</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B, C, D</td>
<td>( P = \phi^3_{BC} \phi^3_{BCD} )</td>
<td><img src="image4" alt="Diagram" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \phi^3_{BCD}(bcd) = \sum_a \phi_A(a) \phi_{AB}(ab) ) ( \phi_{AC}(ac) \phi_{ABD}(abd) )</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>C, D</td>
<td>( P = \phi^4_{CD} )</td>
<td><img src="image5" alt="Diagram" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \phi^4_{CD}(cd) = \sum_b \phi^2_{BC}(bc) \phi^3_{BCD}(bcd) )</td>
<td></td>
</tr>
</tbody>
</table>
3.2 Elimination that takes advantage of Bayes Net parametrization

In the previous section, the VE was carried out without regard to the probabilistic semantics of the potentials. Now we will repeat the elimination exploiting the fact that the potentials are conditional probabilities of the form $P_{X|\text{pa}(X)}$. Hence, whenever we encounter an elimination of the form $\phi_{Y}^{\text{new}}(y) = \sum_{x} P_{X|Y}(x|y)$ we will know that the resulting potential $\phi_{Y}^{\text{new}} \equiv 1$ and therefore we will not carry it over in the computation.

In terms of the graph representation, we shall start with the original DAG, ignoring the directionality of the edges, but not moralizing [Exercise: why?]. When a new potential is equal to 1, no new edges will be added to the graph. The rest of the algorithm proceeds as before.

We repeat the elimination in section 3.1 under the modified algorithm. To better illustrate what happens, a new elimination ordering ($F, D, E, A, B$, with query variable $C$) is used [Exercise: do the elimination with the previous ordering too.]
<table>
<thead>
<tr>
<th>Var.</th>
<th>Neigh</th>
<th>New factorization</th>
<th>New graph</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( P = P_A P_{B</td>
<td>A} P_{C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \phi_1^E(e) = P_{F</td>
<td>E}(e, 1) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \phi_2^{AB}(ab) = \sum_d P_{D</td>
<td>AB}(d</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \phi_3^{BC}(bc) = \sum_e P_{E</td>
<td>BC}(bc e) \phi_1^E(e) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \phi_4^{A}(c) = \sum_b \phi_3^{BC}(bc) \phi_3^{BC}(bc) )</td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \phi_4^{B}(c) = \sum_b \phi_3^{BC}(bc) \phi_3^{BC}(bc) )</td>
<td><img src="image" alt="Graph" /></td>
</tr>
</tbody>
</table>
4 Computational complexity and induced width

The computational work done in VE is primarily the calculation of new potentials. A potential over \(n\) binary variables is a table of \(2^n\) values, each computed by summing a product of potentials over an eliminated variable. So the total computational cost of computing a new potential over \(n\) variables is \(2^n + 1\times (\text{the number of potentials multiplied})\).

Every step eliminates one variable, and create one new potential. Thus, the total number of steps is \(n = |V|\) and the total number of new potentials is (at most) \(n\). Thus, if we start with \(O(n)\) potentials (this is true for Bayes nets but not always for MRF’s), the total number of potentials involved is also \(O(n)\). If \(c\) is the maximum number of variables in any potential under a particular elimination order \(\pi\). It follows that the time complexity of the entire variable elimination is \(O(n2^{c+1})\). (However finding the elimination order that minimizes \(c\) is an NP-complete problem!)

Define the \textit{induced width of} \(G\) \textit{under} \(\pi, w_\pi\), to be the maximum size of the neighbor set \(N\) in the execution of \texttt{ELIMINATE} for an ordering \(\pi\) over all the variables in the graph (total elimination). Define the \textit{elimination width} to be the minimum of \(w_\pi\) over all elimination orders. It is not hard to see that the elimination width of an unconnected graph is 0, of any tree is 1 (achieving this requires the right choice of elimination order), and of a loop is 2.

There are a variety of different ways of defining graph complexity that are all equivalent; a graph’s elimination width can be shown to be equal to its \textit{tree width}, a complexity measure related to the decomposition of the graph into a tree of cliques that we will explore as part of the junction tree algorithm.