STAT 535 Lecture 6

Variable Elimination

©(2007) Marina Meilă, Carl de Marcken
mmp@stat.washington.edu

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 $\mathcal{G}$ be the conditional indpendence graph and P the probability distribution. Let $\mathcal{G}^m$ and P' be the graph and distribution after moralization. Moralization exactly maintains the probability distribution: $P' = P$, and $\mathcal{G}^m$ remains an I-map of P. But some of the conditional independencies encoded in $\mathcal{G}$ may be missing from $\mathcal{G}^m$. These ae 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 $\mathcal{G}$, 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 $\mathcal{G}$:

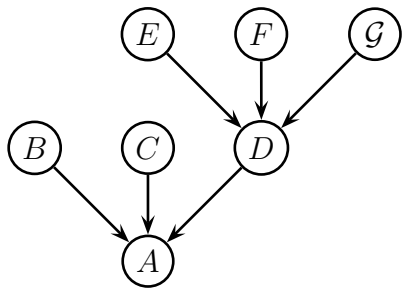
$$\phi_{X_i}(x) = P_{X_i}(x_i|\text{pa}(x_i)).$$

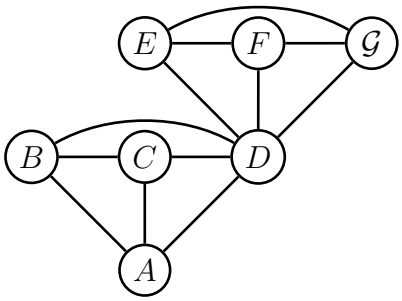
By the Hammersley-Clifford theorem, $\mathcal{G}^m$ is an I-map of P if the domain of every potential function is a (not necessarily maximal) clique in $\mathcal{G}^m$, or equivalently, if every family in $\mathcal{G}$ is a clique in $\mathcal{G}^m$. Thus, two equivalent methods of constructing the edges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

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

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

Here's an example of moralization:





$$P = P_B P_C P_E P_F P_G P_{D|EFG} P_{A|BCD}$$

$$P = \phi_B \phi_C \phi_E \phi_F \phi_G \phi_{DEFG} \phi_{ABCD}$$

Notice that $\mathcal{G}^m$ 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 Ψ_{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 of potential functions over V;
- $O = \{\langle X_j = x_j \rangle\}$, observations of a subset of V (possibly empty);
- π, the *elimination order*, an ordered subset of V; for simplicity, π is assumed to include the observation varibles O.

The output:

- ϕ_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.

ϕ_Z = ELIMINATE(V, Φ = $\{\phi\}$, π = $X_1 \ldots X_m$, O = $\{\langle X_j = x_j \rangle\}$)

```
for X_i in π
    Φ_i = {φ ∈ Φ | X_i ∈ domain(φ)} // potentials involving X_i

    if ⟨X_i = v⟩ ∈ O
```

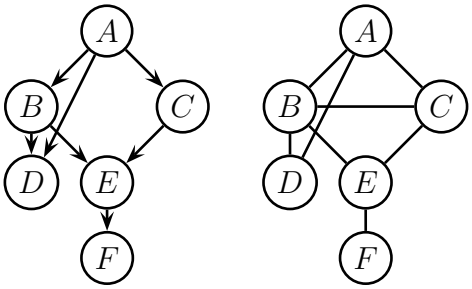
```
      // X_i is observed: create new potentials by fixing X_i
      for φ ∈ Φ_i
         N = domain(φ) \ {X_i}
         φ' = new potential over variables N
         for each config x in Ω_N
            φ'(x) = φ(x, v)
         Φ = Φ ∪ {φ'}
   else
      // compute new potential by marginalizing over X_i
      Ω = ⋃_{φ∈Φ_i} domain(φ) \ {X_i} // neighbors of X_i
      φ' = new potential over variables N
      for each config x in Ω
         φ'(x) = Σ_{x_i∈X_i} Π_{φ∈Φ_i} φ(x, x_i)
      Φ = Φ ∪ {φ'}

   //eliminate X_i and its potentials
   V = V \ {X_i}
   Φ = Φ \ Φ_i

//compute table over remaining variables
φ_Z = new potential over remaining variables V
for each config x in V
   φ_Z(x) = Π_{ψ∈Φ} φ(x)
return φ_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 $P = \phi_A \phi_{AB} \phi_{AC} \phi_{ABD} \phi_{BCE} \phi_{EF}$ and graph above right.

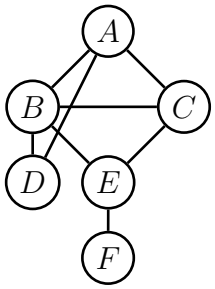
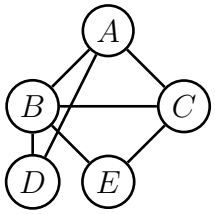
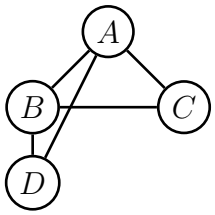
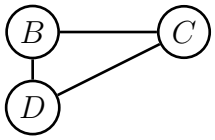
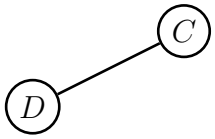
Suppose the goal is to compute $P_{CD|F=1}$; this is accomplished by first computing the joint $P_{CD,F=1}$ using VE, by eliminating A, B, E and F. After choosing an elimination order (here, arbitrarily, $FEAB$), VE gives a table of over the remaining variables:

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

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:

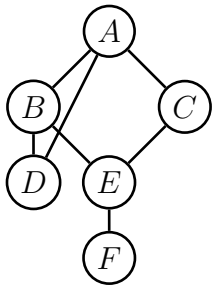
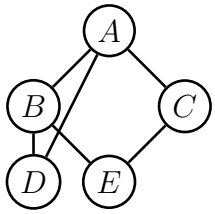
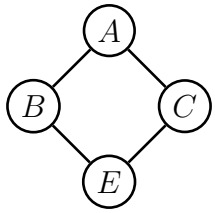
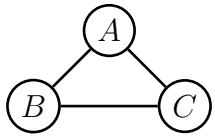
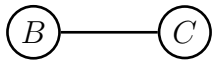
Var.	Neigh	New factorization	New graph
		$P = \phi_A \phi_{AB} \phi_{AC} \phi_{ABD} \phi_{BCE} \phi_{EF}$	
F	E	$P = \phi_A \phi_{AB} \phi_{AC} \phi_{ABD} \phi_{BCE} \phi^1_E$ $\phi^1_E(e) = \phi_{EF}(e, 1)$	
E	B, C	$P = \phi_A \phi_{AB} \phi_{AC} \phi_{ABD} \phi^2_{BC}$ $\phi^2_{BC}(bc) = \sum_e \phi_{BCE}(bce) \phi^1_E(e)$	
A	B, C, D	$P = \phi^2_{BC} \phi^3_{BCD}$ $\phi^3_{BCD}(bcd) = \sum_a \phi_A(a) \phi_{AB}(ab) \phi_{AC}(ac) \phi_{ABD}(abd)$	
B	C, D	$P = \phi^4_{CD}$ $\phi^4_{CD}(cd) = \sum_b \phi^2_{BC}(bc) \phi^3_{BCD}(bcd)$	

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 $P_{X|pa(X)}$. Hence, whenver we encounter an elimination of the form $\phi_Y^{new}(y) = \sum_x P_{X|Y}(x|y)$ we will know that the resulting potential $\phi_Y^{new} \equiv 1$ 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.]

Var.	Neigh	New factorization	New graph
		$P = P_A P_{B\|A} P_{C\|A} P_{D\|AB} P_{E\|BC} P_{F\|E}$	
F	E	$\phi^1_E(e) = P_{F\|E}(e,1)$ $P = P_A P_{B\|A} P_{C\|A} P_{D\|AB} P_{E\|BC} \phi^1_E$	
D	A, B	$\phi^2_{AB}(ab) = \sum_d P_{D\|AB}(d\|ab) \equiv 1$ $P = P_A P_{B\|A} P_{C\|A} P_{E\|BC} \phi^1_E$	
E	B, C	$\phi^2_{BC}(bc) = \sum_e P_{E\|BC}(bce) \phi^1_E(e)$ $P = P_A P_{B\|A} P_{C\|A} \phi^2_{BC}$	
A	B, C	$\phi^3_{BC}(bc) = \sum_a P_A(a) P_{B\|A}(b\|a) P_{C\|A}(c\|a)$ $P = \phi^2_{BC} \phi^3_{BC}$	
B	C	$\phi^4_C(c) = \sum_b \phi^2_{BC}(bc) \phi^3_{BC}(bc)$ $P = \phi^4_C$	

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 $2^{n+1}\times$(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 $\mathcal{O}(n)$ potentials (this istrue for Bayes nets but not always for MRF's), the total number of potentials involved is also $\mathcal{O}(n)$. If c is the maximum number of variables in any potential under a particular elimination order π. It follows that the time complexity of the entire variable elimination is $\mathcal{O}(n2^{c+1})$. (However finding the elimination order that minimizes c is an NP-complete problem!)

Define the *induced width of* $\mathcal{G}$ *under* π, w_π, to be the maximum size of the neighbor set N in the execution of `ELIMINATE` for an ordering π over all the variables in the graph (total elimination). Define the *elimination width* to be the minimum of w_π 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 *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.