10-708: Probabilistic Graphical Models 10-708, Spring 2018

5 : Exact Inference

Lecturer: Kayhan Batmanghelich *Scribes: Ming-Fang Chang, Dheeraj Rajagopal*

A Graphical Model(GM), M is a compact representation of probability distribution p. Typical tasks are

- Inference : answer queries about probability P
- Learning : estimate a plausible model M from data (But for Bayesian, it's an inference problem.)

1 Queries

- **Likelihood**: Compute probability $P(\boldsymbol{e})$ of evidence $\boldsymbol{e}$
- **Conditional Probability**:
  Compute the conditional probability distribution $P(X|\boldsymbol{e})$
  - **Marginalization**: eliminate "don't care" or "unobserved" parameters by
    $$P(Y|\boldsymbol{e}) = \sum_{Z} P(Y, Z|\boldsymbol{e}) \tag{1}$$
  - **Prediction**: compute the probability of an outcome given the starting condition
  - **Diagnosis**: compute the probability of starting condition given outcomes
- **Most Probable Assignment(MPA)**:
  Find the MPA for some variables of interest. For example, the maximum a posteriori of y is
  $$MPA(Y|\boldsymbol{e}) = argmax_{y \in Y} P(Y|\boldsymbol{e}) = argmax_{y \in Y} \sum_{Z} P(Y, Z|\boldsymbol{e}) \tag{2}$$
  - **Classification**: find most likely label, given the evidence
  - **Explanation**: what is the most likely scenario, given the evidence

  Notice that best marginal might not be the same as joint distribution.

Generally speaking, computing $P(X = x|e)$ in a GM is NP-hard. The complexity depends on GM structure.

2 Variable Elimination

Consider following GM, we want to compute the likelihood that E is active. Let k be the dimension of parameters and n be the number of parameters.

Figure 1: Caption

- **Naive sum** $O(k^n)$: expensive!

$$P(e) = \sum_d \sum_c \sum_b \sum_a P(a, b, c, d, e) \tag{3}$$

- **Chain decomposition and elimination** $O(nk^2)$:

$$\begin{aligned} P(e) &= \sum_d \sum_c \sum_b \sum_a P(a)P(b|a)P(c|b)P(d|c)P(e|d) \\ &= \sum_d \sum_c \sum_b P(b)P(c|b)P(d|c)P(e|d) \\ &= \sum_d \sum_c P(c)P(d|c)P(e|d) \\ &= \sum_d P(d)P(e|d) \end{aligned} \tag{4}$$

Each summation operation is actually a matrix-vector multiplication, whose complexity is $O(k^2)$. Repeat for all variables then we get $O(nk^2)$. See the Message Passing example on slides. Each member only infer it's status from what currently have and the information passed from neighbors.

3 Example of Variable Elimination

In this section, we visit an example of how to do Variable Elimination in a directed or undirected graphical model.

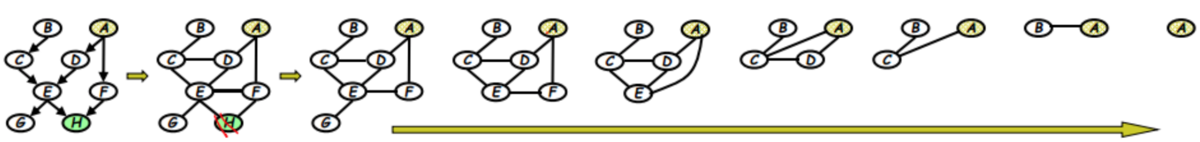

Figure 2: Variable Elimination example

Lets consider the example given in the figure 2. Consider the query $P(A|h)$.

These are the following steps in the elimination. The query can be into the initial factors

$$P(A|h) = P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f) \tag{5}$$

Fix the evidence node h on its observed value $\tilde{h}$, where $m_h(e,f) = \sum_k p(h|e,f)\delta(h = \tilde{h})$

$$\implies P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f) \tag{6}$$

eliminating G, where $m_g(e) = \sum_g p(g|e) = 1$

$$\implies P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)m_h(e,f) \tag{7}$$

eliminate F, $m_f(e,a) = \sum_f p(f|a)m_h(e,f)$

$$\implies P(a)P(b)P(c|d)P(d|a)P(e|c,d)m_f(a,e) \tag{8}$$

eliminate E, $m_e(a,c,d) = \sum_e p(e|c,d)m_f(a,e)$

$$\implies P(a)P(b)P(c|d)P(d|a)m_d(a,c) \tag{9}$$

eliminate D, $m_d(a,c) = \sum_d p(d|a)m_e(a,c,d)$

$$\implies P(a)P(b)m_c(a,b) \tag{10}$$

eliminate C, $m_c(a,b) = \sum_c p(c|b)m_d(a,c)$

$$\implies P(a)m_b(a) \tag{11}$$

And finally eliminate B where $m_b(a) = \sum_b p(b)m_c(a,b)$, we get

$$P(a,\tilde{h}) = p(a)m_b(a), \quad p(\tilde{h}) = \sum_a p(a)m_b(a)$$
$$\implies P(a|\tilde{h}) = \frac{p(a)m_b(a)}{\sum_a p(a)m_b(a)}$$

4 Graph Elimination

As we see in the example in the previous section, the elimination procedure for undirected graphs works in the following steps

1. Start with the undirected graph $G(V,E)$. Based on the query, begin with the order of the elimination I.

2. For each of the next variable in I from step 1, remove the node from the graph. Connect the neighbors of that node

3. In the reconstituted graph $G'(V,E')$, retain the edges during the elimination procedure.

4. Go to step 2

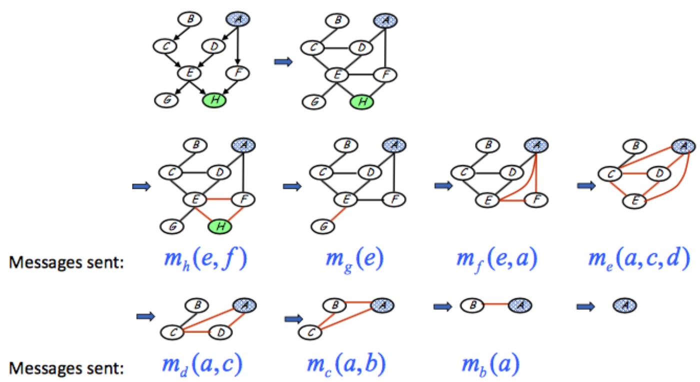

Figure 3: Elimination Cliques, where m is the message passed at each stage

To repeat the same procedure for directed graphical models, we first need to moralize the graph. The remaining procedure stays the same as in undirected graphs in the example above. After each iteration, the factors result during the Variable Elimination are captured by recording the elimination clique. We can view this a *message* passed from factors to nodes, or vice-versa. Elimination can be hence viewed as message passing on a clique tree where messages can be reused. This is shown in figure 3.

The overall complexity is determined by the number of the largest elimination clique. Also, finding the best elimination ordering of a graph is NP-hard, but often there are 'obvious' optimal or near-optimal elimination ordering. Good elimination techniques often optimize to create smaller cliques.