10-601b: Machine Learning, Spring 2014

Recitation 9: Graphical Models: D-separation, Variable Elimination and Inference

Jing Xiang

March 18, 2014

1 D-separation

Let's start by getting some intuition about why D-separation works. First consider the following Bayesian network which shows explains how metal becomes rusty.

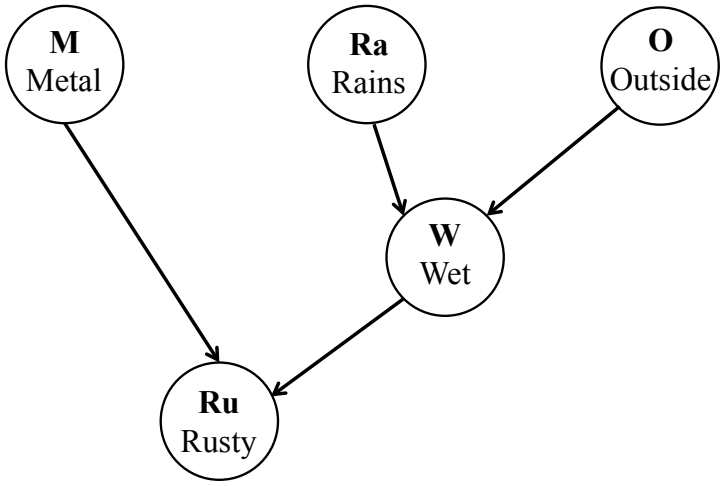

Figure 1: Bayesian Network Example: What happens to metal? (Credit: Geoff Gordon)

Blocking!

What happens if W is shaded, that is the following true? Ra $\perp$ Ru | W

Rains $\rightarrow$ Wet $\rightarrow$ Rusty
P(Ra) P(W | Ra) P(Ru | W)

Rains $\rightarrow$ Wet (shaded) $\rightarrow$ Rusty
P(Ra) P(W=T | Ra) P(Ru | W=T) / P(W=T)
(P(Ra) P(W=T | Ra)) (P(Ru | W=T) / P(W=T))
Ra $\perp$ Ru | W

Explaining Away!

If W is not shaded, then is Ra $\perp$ O?
If W is shaded, then is Ra $\perp$ O | W?

Rains $\rightarrow$ Wet $\leftarrow$ Outside
$\sum_{W}$ P(Ra) P(O) P(W | Ra, O) = P(Ra) P(O)

Rains $\rightarrow$ Wet (shaded) $\leftarrow$ Outside
P(Ra) P(O) P(W = F | Ra, O) / P(W=F)
Now they become dependent. So, Ra $\not\perp$ O | W
Intuitively, if we know were not wet and we find out its raining: then we know we're not outside.

1.1 What happens when you have multiple paths?

For any variables X, Y and Z, X and Z are d-separated if EVERY undirected path from X to Z is blocked by Y. Consider the following graphical model which is littered with multiple paths.

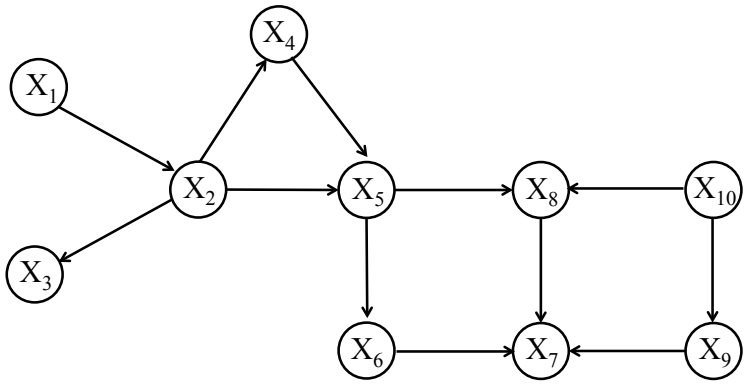

Figure 2: Bayesian Network with Multiple Paths (Credit: 10-708 Spring 2013 TA's)

Do the following independence relations hold?

1. $X_7 \perp X_{10}|X_9$
2. $X_6 \perp X_8|X_5$
3. $X_2 \perp X_8|X_5$
4. $X_1 \perp X_9|X_8$

Solution

1. No. There are two paths: a) $\{X_7, X_8, X_{10}\}$ and b) $\{X_7, X_9, X_{10}\}$. Only b) is blocked by X_9 .

2. Yes. There are two paths: a) $\{X_5, X_6, X_7\}$ and b) $\{X_5, X_8, X_7\}$. Only a) is blocked by X_5, and b) is blocked by the absence of X_7.
3. Yes. There are multiple paths between X_2 and X_8 listed below and they are all blocked by X_5.
   (a) $\{X_2, X_4, X_5, X_8\}$
   (b) $\{X_2, X_5, X_8\}$
   (c) $\{X_2, X_5, X_6, X_7, X_8\}$
4. No. There are multiple paths between X_1 and X_9 and it is not necessary to list them all. Notice that two of such paths must pass through X_8, and since it is given, those paths are no longer blocked.

1.2 What happens when you have sets of variables?

If you have X, Y and Z which are three disjoint subsets of nodes, then Y d-separates X and Z if every undirected path from the sets X and Z is blocked by Y. Consider the following example.

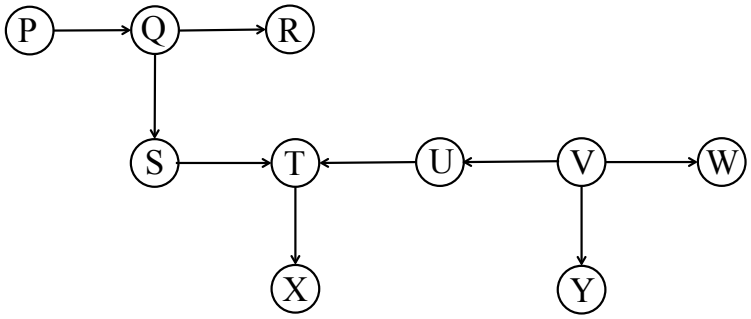

Figure 3: Bayesian network example for sets.

Do the following independence relations hold?

1. $\{P, Q, R, S\} \perp \{U, V, W, Y\}|T$
2. $\{P, Q, R, S\} \perp \{U, V, W, Y\}|\emptyset$
3. $\{P, Q, R, S\} \perp \{U, V, W, Y\}|X$
4. $\{S, T, X\} \perp \{W\}|Y$

Solution

1. No, this is a converging connection, given T would make the two sets dependent.
2. Yes, conditioning on empty will make the sets independent.
3. No, conditioning on T or any of it's descendants such as X would make the two sets dependent.
4. No, conditioning on Y does not help, it needs to be U or V.

2 Inference

Inference is about answering some queries. You can build a distribution as a database of probabilistic dependencies and conditional distributions and you want to answer some queries. One query is to find the likelihood of the data and the second query is to find posteriori belief. Without a graphical model, the only thing you can do is nested summations that lead to exponential complexity. However, using a graphical model, we can exploit the properties of the graph to make it more efficient.

Given a distribution P over a set of random variables $\boldsymbol{X}$, there are several queries we are interested in.

1. $P(\boldsymbol{e})$ - **Likelihood of evidence $\boldsymbol{e}$**. This is the marginal probability of a subset of variables $\boldsymbol{e}$, the evidence, in the distributions.

$$P(\boldsymbol{e}) = \sum_{X_1} \cdots \sum_{X_k} P(X_1, \ldots, X_k, \boldsymbol{e}) \tag{1}$$

2. $P(\boldsymbol{X}|\boldsymbol{e})$ - **Posteriori Belief** The is the conditional probability distribution of some query nodes conditioned on the evidence.

$$P(\boldsymbol{X}|\boldsymbol{e}) = \frac{P(\boldsymbol{X}, \boldsymbol{e})}{\sum_{\boldsymbol{X}} P(\boldsymbol{X}, \boldsymbol{e})} \tag{2}$$

   We can also answer conditional probability of a subset of variables $\boldsymbol{Y} \subseteq \boldsymbol{X}$ given the evidence $\boldsymbol{e}$:

$$P(\boldsymbol{Y}|\boldsymbol{e}) = \sum_{\boldsymbol{Z}} P(\boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{e}) \tag{3}$$

   The process of summing out $\boldsymbol{Z}$ is called marginalization.

2.1 Variable Elimination

Variable elimination is an exact inference method. The key insight is that given the query and evidence nodes in a Bayesian network, finding the posteriori belief does not involve all the variables in the joint distributions. Thus, we want to avoid marginalization involving naive summation over an exponential number of terms and use the graph structure to help us make this more efficient.

2.1.1 Variation Elimination on a Chain

We are given a distribution $P(A, B, C, D, E)$ which has graph structure depicted in Figure 4.

Figure 4: Bayesian network of five random variables A, B, C, D, E where E is observed.

We want to compute $P(e)$. The first thought is by summing out all other variables in the joint distribution:

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

However, the number of summations is equal to number of all configurations of $A \sim D$, 2^4 or $O(n^4)$, which isn't ideal.

We can be more clever and exploit the graph structure to simplify our computation. Specifically, we can factorize the joint probability according to the graph, and move the summation in such that it only covers terms that are affected.

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

Note that we **grouped all terms that have variable A together**, i.e., $P(a)$ and $P(b|a)$, and did the summation only on the product of these terms.

We can continue doing this until only e is left:

$$\begin{aligned} P(e) &= \sum_b \sum_c \sum_d P(c|b)P(d|c)P(e|d) \sum_a P(a)P(b|a) \\ &= \sum_b \sum_c \sum_d P(c|b)P(d|c)P(e|d)\tau_B(b) \\ &= \sum_c \sum_d P(d|c)P(e|d) \sum_b P(c|b)\tau_B(b) \\ &= \sum_c \sum_d P(d|c)P(e|d)\tau_C(c) \\ &= \sum_d P(e|d) \sum_c P(d|c)\tau_C(c) \\ &= \sum_d P(e|d)\tau_D(d) \\ &= \tau_E(e) \end{aligned}$$

In this example, the intermediate results, $\tau_X(x)$ is actually equal to $P(x)$ (for example, $\tau_B(b) = P(b)$), and that's because of the nice graph structure. In some other graphs, and in undirected graphs, they may not be meaningful, and that's the reason we use the τ notation here.

Just by being clever in the way we do the summation, you can get fewer than 16 summations. The **complexity** of the whole computation is $O(k|Val(X_i)| \cdot |Val(X_{i+1})|) = O(kn^2)$, which is linear in the number of variables. Compared with the original exponential complexity, $O(n^k)$, this is a huge improvement.

2.1.2 Variation Elimination on a Graph

Consider the more complicated graphical model shown in Figure 5. The complete sets of random variables is $\mathcal{X} = A, B, C, D, E, F, G, H$. We want to evaluate the query $P(A|H = h)$.

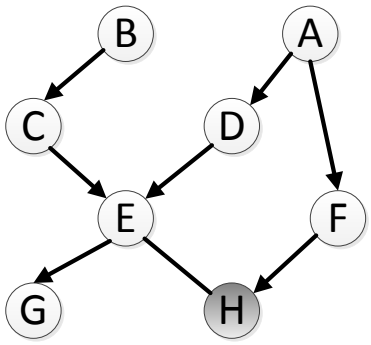

Figure 5: Graph for variable elimination example, where H is observed. (Credit: Eric Xing)

Query $P(A|h)$

First, let's factorize the joint distribution. By doing so, you get the following initial factors.

$P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$

Choose an elimination ordering and then push in the sums.

$$P(a)\sum_b P(b)\sum_c P(c|b)\sum_d P(d|a)\sum_e P(e|c,d)\sum_f P(f|a)\sum_g P(g|e)\sum_h P(h|e,f)$$

Now let's break down the variable elimination process step by step.

Step 1: Fix evidence
Fix the evidence node h on its observed value: $h = \bar{h}$.

$m_h(e,f) = \sum_h P(h|e,f)\delta(h = \bar{h})$
$P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$

Step 2: Eliminate G
Compute $m_g(e) = \sum_g P(g|e) = 1$
$P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_g(e)m_h(e,f)$
$P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$

Step 3: Eliminate F
Compute $m_f(e,a) = \sum_f P(f|a)m_h(e,f)$
$P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$
$P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(a,e)$

Step 4: Eliminate E
Compute $m_e(e,a,d) = \sum_e P(e|c,d)m_f(a,e)$
$P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(a,e)$
$P(a)P(b)P(c|b)P(d|a)m_e(a,c,d)$

Step 5: Eliminate D
Compute $m_d(a,c) = \sum_d P(d|a)m_e(a,c,d)$
$P(a)P(b)P(c|b)P(d|a)m_e(a,c,d)$
$P(a)P(b)P(c|b)m_d(a,c)$

Step 6: Eliminate C
Compute $m_c(a,b) = \sum_c P(c|b)m_e(a,c)$
$P(a)P(b)P(c|b)m_d(a,c)$
$P(a)P(b)m_c(a,b)$

Step 7: Eliminate B
Compute $m_b(a) = \sum_b P(b)m_c(a,b)$
$P(a)P(b)m_c(a,b)$
$P(a)m_b(a)$

Step 8: Finish

$P(a,\bar{h}) = P(a)m_b(a)$
$P(\bar{h}) = \sum_a P(a)m_b(a)$

$P(a|\bar{h}) = \frac{P(a)m_b(a)}{\sum_a P(a)m_b(a)}$

Now we should analyze the general complexity of this algorithm. We have a summation operation and the product operation. Below is the calculation of the complexity.

Total number of additions:

$$= |Val(X)| \cdot \prod_i |Val(Y_{C_i})|$$

Total number of multiplications:

$$= k|Val(X)| \cdot \prod_i |Val(Y_{C_i})|$$

In the above equations, k is the number of factors and Y_{C_i} is the i^{th} clique defined on Y. The complexity is

polynomial to the number of components in the factor. This is very different from exponential complexity. The complexity of the algorithm is dependent on the intermediate entity that you create, the term that gets put back in the stack. That's where the graph is attractive because it allows you to visualize what intermediate entity you created.

However, this means that the selection of ordering is very important, because the order determines the size of the factors. Take a look at Figure 6 and you will see the factors that we created via the variable elimination process. In addition, notice the undirected structure of the factors. This leads us to graph elimination.

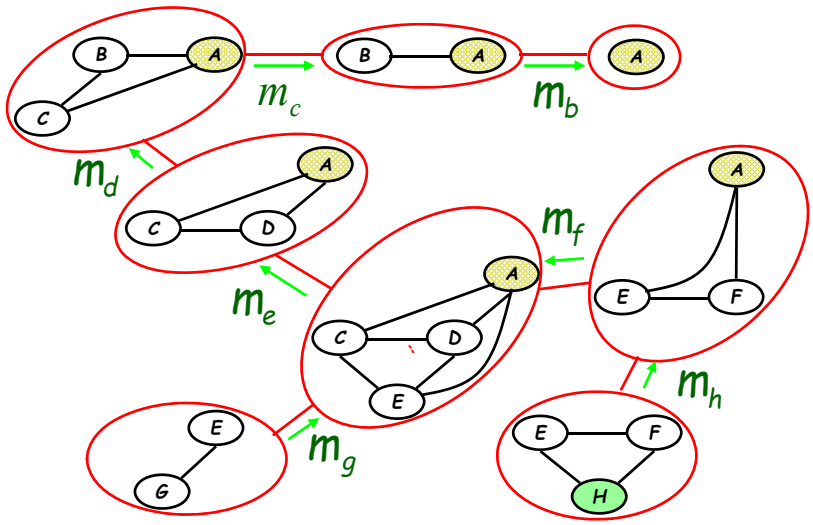

Figure 6: The factors that we created in variable elimination. (Credit: Eric Xing)

2.2 Graph Elimination

Since the chosen ordering when we perform variable elimination is so important, it would be great if there was an easier way to visualize the factors we create and thus try different orderings. Well, there is! The procedure is called graph elimination and we start by moralizing the graph. To moralize means to take every node and connect it's parents. The result is an undirected graph that we can do graph elimination on. Now, when we eliminate one node, if the neighbours of this node are unconnected, then we connect them. Figure 7 shows what we get if we perform graph elimination on the example we just considered.

Graph elimination gives you some interesting data structures. What's interesting in graph elimination are the elimination cliques. The cliques include the variable and its neighbours that were connected. Each clique corresponds to the intermediate terms that get generated from the equation. Why is this interesting? Graph elimination does not give you any extra power but now you can now visualize. For a star graph, you can eliminate the external nodes, creating intermediate terms with 2 variables. Or, you can start eliminating from the center, in which case, the intermediate terms are the entire clique. A function of this has a huge complexity. This is called the tree width of a graphical model, the bottleneck complexity of the biggest clique. For a tree structure, you can go from the leaves to the top or the opposite without creating huge factors. Thus, a tree is a structure that people are happy to do inference on.

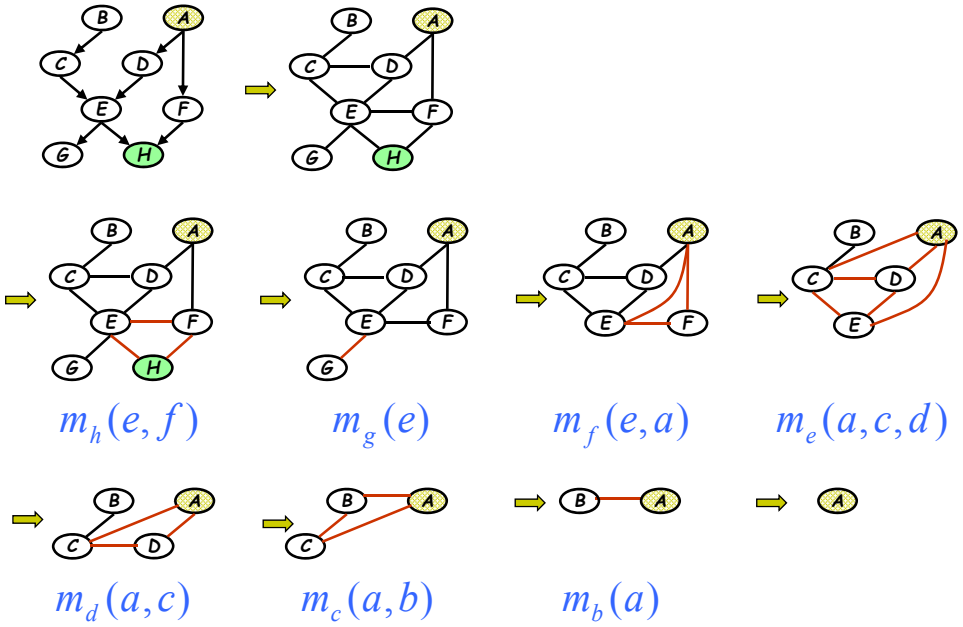

Figure 7: Depicted is the process of graph elimination. (Credit: Eric Xing)

2.3 Inference: Worked Example

Figure 8: Westeros map with the associated Bayes net. (Credit: `http://www.optionated.com/wp-content/uploads/2012/04/Westeros-and-Essos-new-map.jpg`)

Winter is coming, and the White Walkers are ravaging the countryside. We will model the pillaging by the White Walkers with a Bayesian network, shown in Figure 8. The major cities in Westeros are represented with binary random variables *Winterfell (W), Iron Islands (I), Riverrun (R), King's Landing (K), Highgarden (H), Dorne (D), Shivering Sea (S), Bravos (B), Pentos (P) and Myr (M)*. Each of these binary variables has state winter (w) or summer (s) corresponding to whether winter has come to the city or whether it is still summer there.

1. We want to calculate $Pr(M = w)$, i.e., the probability that Myr is in winter state. What is the subset $\mathcal{X}$ of the variables/cities in the network that we have to consider in order to calculate this probability?

   Solution We only need to consider S, B, and P: these are the only parents of M, and without any evidence, every node's probability distribution depends only on the distributions of its parents.

2. How many possible elimination orderings are there for the variables $x \in \mathcal{X}$ for calculating $Pr(M = w)$?

   Solution There are $3! = 6$ possible elimination orderings.

3. We will now consider a simpler graph to perform inference on. We will use the subgraph consisting of the cities *Winterfell (W)*, *Riverrun (R)*, *Iron Islands (I)*, and *King's Landing (K)*, shown in Figure 9 with its conditional probability tables.

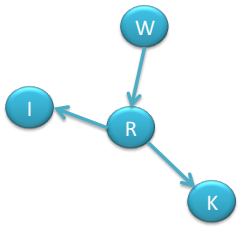

Figure 9: A simpler Bayesian network (a subgraph of Figure 8).

W=w	W=s
0.9	0.1

	R=w	R=s
W=w	0.8	0.2
W=s	0.3	0.7

	I=w	I=s
R=w	0.8	0.2
R=s	0.3	0.7

	K=w	K=s
R=w	0.8	0.2
R=s	0.3	0.7

Find the state of King's Landing given that winter has come to Winterfell.

Solution

First thing is to come up with an ordering. For complicated graphs, you would use graph elimination to visualize the best ordering. But here, it is relatively simple, and with the exception of picking R first, all the orderings are pretty much equivalent, so we choose I, W, R, and then K.

Then, we factor the joint distribution and push the sums in and perform variable elimination.

$$
\begin{aligned}
P(K, W = w) &= \sum_{I,R,W} P(K|R)P(R|W)P(W = w)P(I|R) \\
&= \sum_{R} P(K|R) \sum_{W} P(R|W)P(W = w) \sum_{I} P(I|R) \\
&= \sum_{R} P(K|R) \sum_{W} P(R|W)P(W = w) \cdot 1 \\
&= \sum_{R} P(K|R)P(R|W = w)P(W = w) \\
&= \begin{bmatrix} 0.8 & 0.3 \\ 0.2 & 0.7 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix} \cdot 0.9 \\
&= \begin{bmatrix} 0.63 \\ 0.27 \end{bmatrix}
\end{aligned}
$$

Now, we just computed the joint, to get the conditional probability, we need to divide by the probability of evidence.

$$
\begin{aligned}
P(K|W = w) &= \frac{P(K, W = w)}{P(W = w)} && (4) \\
&= \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} && (5)
\end{aligned}
$$