More Bayesian Networks

Arnav Gupta

November 27, 2024

Contents

1 Bayesian/Belief Networks

To represent a domain in a belief network, consider:

- what the relevant variables are
  - what might be observed → **evidence**
  - what information is desired → **query**
  - what features make the model simpler → other variables
- what values the variables will take

- what relationships exist between variables, expressed in terms of **local influence**
- how the values of each variable depend on the parents, expressed in terms of conditional probabilities

1.1 Bayes' Rule

Agent has a prior belief in a hypothesis h, $P(h)$.

Agent observes some evidence e that has a **likelihood** given the hypothesis $P(e \mid h)$.

The agent's **posterior belief** about h after observing e, $P(h \mid e)$ is given by Bayes' Rule

$$P(h \mid e) = \frac{P(e \mid h)P(h)}{P(e)} = \frac{P(e \mid h)P(h)}{\sum_h P(e \mid h)P(u)}$$

Often agents have **causal knowledge** but want to do **evidential reasoning**.

1.2 Simple Forward Inference (Chain)

Computing marginal requires simple forward propagation of probabilities.

Consider the following Bayesian Network:

Marginalization

$$P(B) = \sum_{m,c} P(M = m, C = c, B)$$

Chain Rule

$$P(B) = \sum_{m,c} P(B \mid m, c)P(m \mid c)P(c)$$

Independence

$$P(B) = \sum_{m,c} P(B \mid m, c)P(m)P(c)$$

Distribution of Produce over Sum

$$P(B) = \sum_{m} P(m) \sum_{c} P(c)P(B \mid m, c)$$

All terms on the last line are CPTs in the Bayes' Network. Only ancestors of B are considered.

With multiple parents, the evidence can be pooled. This also works with upstream evidence.

1.3 Simple Backward Inference

When evidence is downstream of query, reason backwards, which requires Bayes' rule.

2 Variable Elimination

Intuitions above are the **polytree** algorithm. This works for simple networks without loops.

A more general algorithm is **Variable Elimination**:

- applies sum-out rule repeatedly
- distributes sums

Factor: representation of a function from a tuple of random variables into a number

Write a factor f on variables $X_1, \ldots, X_j$ as $f(X_1, \ldots, X_j)$. Assigning some or all of the variables of a factor is **restricting** a factor:

- $f(X_1 = v_1, X_2, \ldots, X_j)$ where $v_1 \in \text{dom}(X_1)$ is a factor on $X_2, \ldots, X_j$
- $f(X_1 = v_1, X_2 = v_2, \ldots, X_j = v_j)$ is a number that is the value of f when each X_i has value v_i

The **product** of factor $f_1(X, Y)$ and $f_2(Y, Z)$ where Y are the variables in common, is the factor $(f_1 \times f_2)(X, Y, Z)$ defined by

$$(f_1 \times f_2)(X, Y, Z) = f_1(X, Y) f_2(Y, Z)$$

To **sum out** a variable X_1 with domain $\{v_1, \ldots, v_k\}$, from factor $f(X_1, \ldots, X_j)$ resulting in a factor on $X_2, \ldots, X_j$ defined by:

$$\left(\sum_{X_1} f\right)(X_2, \ldots, X_j) = f(X_1 = v_1, \ldots, X_j) + \cdots + f(X_1 = v_k, \ldots, X_j)$$

2.1 Evidence

To compute the posterior probability of Z given evidence $Y_1 = v_1 \wedge \cdots \wedge Y_j = v_j$:

$$P(Z \mid Y_1 = v_1, \ldots, Y_j = v_j) = \frac{P(Z, Y_1 = v_1, \ldots, Y_j = v_j)}{\sum_Z P(Z, Y_1 = v_1, \ldots, Y_j = v_j)}$$

This computation reduces to the joint probability of $P(Z, Y_1 = v_1, \ldots, Y_j = v_j)$ normalized at the end.

The query value can also be restricted, for example $Z = z$.

2.2 Probability of a Conjunction

Suppose the variables of the belief network are $X_1, \ldots, X_n$. To compute $P(Z, Y_1 = v_1, \ldots, Yj = v_j)$, sum out the variables other than query Z and evidence Y

$$Z_1, \ldots, Z_k = \{X_1, \ldots, X_n\} - \{Z\} - \{Y_1, \ldots, Y_j\}$$

Order Z_i into an elimination ordering $Z_1, \ldots, Z_k$

$$P(Z, Y_1 = v_1, \ldots, Y_j = v_j) = \sum_{Z_k} \cdots \sum_{Z_1} \prod_{i=1}^{n} P(X_i \mid \text{parents}(X_i))_{Y_1 = v_1, \ldots, Y_j = v_j}$$

2.3 Computing Sums of Products

Computation in belief networks reduces to computing the sums of products:

- to compute $ab + ac$ efficiently
  - distribute to $a(b + c)$
- to compute $\sum_{Z_1} \prod_{i=1}^{n} P(X_i \mid \text{parents}(X_i))$ efficiently
  - distribute out the factors that don't involve Z_1

2.4 Algorithm

To compute $P(Z \mid Y_1 = v_1 \wedge \cdots \wedge Y_j = v_j)$:

- construct a factor for each conditional probability
- restrict the observed variables to their observed values

- sum out each of the other variables according to some **elimination ordering** for each Z_i in order starting from $i = 1$
  - collect all factors that contain Z_i
  - multiple together and sum out Z_i
  - add resulting new factor back to the pol
- multiply the remaining factors
- normalize by dividing the resulting factor $f(Z)$ by $\sum_Z f(Z)$

2.5 Summing out a Variable

To sum out a variable Z_j from a product $f_1, \ldots, f_k$ of factors:

- partition the factors into
  - those that don't contain Z_j, let these be $f_1, \ldots, f_i$
  - those that contain Z_j, let these be $f_{i+1}, \ldots, f_k$

This gives

$$\sum_{Z_j} f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \times \left(\sum_{Z_j} f_{i+1} \times \cdots \times f_k \right)$$

Explicitly construct a representation of the rightmost factor and replace the factors by the new factor.

2.6 Elimination Ordering

Complexity is linear in the number of variables and exponential in the size of the largest factor. Creating new factors blows this up, but this depends on the **elimination ordering**.

For polytrees, work outside in. For general BNs, this can be hard. Specifically, finding the optimal elimination ordering is NP-hard for general BNs.

In general, inference is NP-hard.

2.6.1 Polytrees

Eliminate singly connected nodes first. Then, no factor is ever larger than original CPTs.

If the most connected nodes are eliminated first, a large factor is created with the singly connected nodes.

2.6.2 Relevance

Certain variables have no impact.

Can restrict attention to relevant variables. Given query Q and evidence $\mathbf{E}$, complete approximation is:

- Q is relevant
- if any node is relevant, its parents are relevant
- if $E \in \mathbf{E}$ is a descendent of a relevant variable, then E is relevant

Irrelevant variable: a node that is not an ancestor of a query or evidence variable

This will only remove irrelevant variables, but may not remove them all.