Statistical Inference in Bayesian Networks

Wenhu Chen

Lecture 9

Readings: RN 13.4. PM 8.4.

Outline

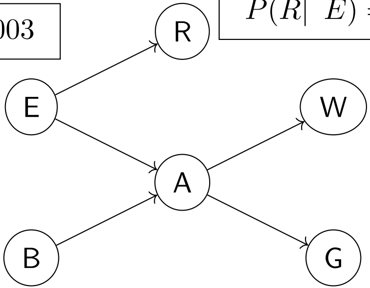
Learning Goals

- Explain how we can perform probabilistic inference more efficiently using the variable elimination algorithm.
- Define factors. Manipulate factors using operations restrict, sum out, multiply and normalize.
- Describe/trace/implement the variable elimination algorithm for calculating a prior or a posterior probability given a Bayesian network.
- Explain how the elimination ordering affects the complexity of the variable elimination algorithm.
- Know about approximate statistical inference.

A Bayesian Network for the Holmes Scenario

$$P(E) = 0.0003$$

$$P(B) = 0.0001$$



$$P(R|\neg E) = 0.0002$$
$$P(R|\ E) = 0.9$$

$$P(W|\neg A) = 0.4$$
$$P(W|\ A) = 0.8$$

$$P(G|\neg A) = 0.04$$
$$P(G|\ A) = 0.4$$

$$P(A|\neg B \wedge \neg E) = 0.01$$
$$P(A|\neg B \wedge\ E) = 0.2$$
$$P(A|\ B \wedge \neg E) = 0.95$$
$$P(A|\ B \wedge\ E) = 0.96$$

Answering a Question

What is the probability that a burglary is happening given that Dr. Watson and Mrs. Gibbon both call?

$$P(B|w \land g)$$

- Query variables: B
- Evidence variables: W and G
- Hidden variables: E, A, and R.

Notice the new notation: capital letters for random variables and lowercase letters for values.

Answering the query using the joint distribution

Evaluate $P(B|w \wedge g)$ in terms of known distributions from the Bayesian network.

Answering the query using the joint distribution

Evaluate $P(B|w \wedge g)$ in terms of known distributions from the Bayesian network.

$\rightarrow$ Following the approach from Lecture 6:

$$
\begin{aligned}
P(B|w \wedge g) &= \frac{P(B \wedge w \wedge g)}{P(w \wedge g)} \\
&= \frac{P(B \wedge w \wedge g)}{P(b \wedge w \wedge g) + P(\neg b \wedge w \wedge g)} \\
&\propto P(B \wedge w \wedge g) \\
&\propto \sum_{e} \sum_{a} \sum_{r} P(B \wedge e \wedge a \wedge w \wedge g \wedge r) \\
&\propto \sum_{e} \sum_{a} \sum_{r} P(B)P(e)P(a|B \wedge e)P(w|a)P(g|a)P(r|e)
\end{aligned}
$$

Q #1: Number of operations using the joint distribution

How many addition and multiplication operations do we need to calculate the expression below?

$$\sum_{e}\sum_{a}\sum_{r} P(B)P(e)P(a|B \land e)P(w|a)P(g|a)P(r|e)$$

(A) ≤ 10

(B) 11-20

(C) 21-40

(D) 41-60

(E) ≥ 61

Q #1: Number of operations using the joint distribution

How many addition and multiplication operations do we need to calculate the expression below?

$$\sum_e \sum_a \sum_r P(B)P(e)P(a|B \land e)P(w|a)P(g|a)P(r|e)$$

(A) ≤ 10

(B) 11-20

(C) 21-40

(D) 41-60

(E) ≥ 61

$\rightarrow$ Correct answer is (D). 47 operations.

Is there any duplication?

- $P(B)$ appears in every single term, but it doesn't involve any of the three hidden variables.
- $P(B)$ can be thought of as a constant number.
- If you pull $P(B)$ outside of the summation, you decrease 8 multiplication to 1 multiplication
- Push summation to as right as possible.

Answering the query using variable elimination algorithm

$$\sum_e \sum_a \sum_r P(B)P(e)P(a|B \wedge e)P(w|a)P(g|a)P(r|e)$$

$\rightarrow$ We should move the summations as much to the right of the expression as possible to reduce the overall # of operations.

$$= \sum_e \sum_a \sum_r P(B)P(e)P(a|B \wedge e)P(w|a)P(g|a)P(r|e)$$

$$= P(B) \sum_e P(e) \sum_a P(a|B \wedge e)P(w|a)P(g|a) \sum_r P(r|e)$$

$$= P(B) \sum_e P(e) \sum_a P(a|B \wedge e)P(w|a)P(g|a)$$

Q #2: Number of operations via the variable elimination algorithm

How many addition and multiplication operations do we need to calculate the expression below?

$$P(B) \sum_e P(e) \sum_a P(a|B \wedge e) P(w|a) P(g|a)$$

(A) ≤ 10

(B) 11-20

(C) 21-40

(D) 41-60

(E) ≥ 61

Q #2: Number of operations via the variable elimination algorithm

How many addition and multiplication operations do we need to calculate the expression below?

$$P(B) \sum_e P(e) \sum_a P(a|B \wedge e) P(w|a) P(g|a)$$

(A) $\leq$ 10

(B) 11-20

(C) 21-40

(D) 41-60

(E) $\geq$ 61

$\rightarrow$ The inner term requires 1 add + 4 mul.
It requires 1 mul + 1 add + (1 mul + 5 opss) * 2 = 14 ops

Q #2: Number of operations via the variable elimination algorithm

How many addition and multiplication operations do we need to calculate the expression below?

$$P(B) \sum_{e} P(e) \sum_{a} P(a|B \wedge e) P(w|a) P(g|a)$$

(A) ≤ 10

(B) 11-20

(C) 21-40

(D) 41-60

(E) ≥ 61

$\rightarrow$ Correct answer is (B) 11-20. 14 operations in total.

Efficient Inference Algorithm

- To compute the posterior or prior distribution from a given Bayesian Network.
- Different orders will lead to different complexity.
- 14 operations vs 47 operations.
- We prefer to choose the most efficient inference algorithm.

Introducing the Variable Elimination Algorithm

- Performing probabilistic inference is challenging.

  → Computing the posterior distribution of one or more query variables given some evidence is #NP. Estimate the posterior probability in a Bayesian network within an absolute error is already NP-hard. No general efficient implementation.

- Exact and approximate inferences.

  → Compute the probabilities exactly.
  Naive approach: enumerate all the worlds consistent with the evidence. Do better below.

- The variable elimination algorithm uses dynamic programming and exploits the conditional independence.

  → Do the calculations once and save the results for later.

  Factors and operations.

Introducing the Variable Elimination Algorithm

- High-Level Idea: Reuse intermediate computation and exploits the conditional independence present in the Bayesian Network.
- Define Factors.
- Restrict Factors to reflect the evidence.
- Multiply factors with shared variables.
- Sum out hidden variables.
- Normalize to obtain probability.

Factors

- A function from some random variables to a number.
- $f(X_1, \ldots, X_j)$: a factor f on variables $X_1, \ldots, X_j$.
- A factor can represent a joint or a conditional distribution. For example, $f(X_1, X_2)$ can represent $P(X_1 \land X_2)$, $P(X_1|X_2)$ or $P(X_1 \land X_3 = v_3|X_2)$.
- Define a factor for every conditional probability distribution in the Bayes net.

$\rightarrow$ Every conditional probability distribution in the Bayes net is a factor.

$f(B), f(E), f(A, B, E), f(R, E), f(W, A), f(G, A)$

$P(B), P(E), P(A|B \land E), P(R|E), P(W|A), P(G|A)$

$$P(B \land w \land g)$$
$$= P(B) \sum_a P(w|a)P(g|a) \sum_e P(e)P(a|B \land e)$$

Restrict a variable

- To eliminate hidden variables, we need to find all factors containing the variable.
- We want to restrict the factor to the case where $X_1 = v_1$.
- This operation produces a new factor which only contains the variables without being restricted.

Restrict a factor

Restrict a factor.

- Assign each evidence variable to its observed value.
- Restricting $f(X_1, X_2, \ldots, X_j)$ to $X_1 = v_1$,
  produces a new factor $f(X_1 = v_1, X_2, \ldots, X_j)$ on $X_2, \ldots, X_j$.
- $f(X_1 = v_1, X_2 = v_2, \ldots, X_j = v_j)$ is a number.

$\rightarrow$ Restrict $f(W, A)$ to $W = t$. Restrict $f(G, A)$ to $G = t$.

$$
\begin{aligned}
& P(B \wedge w \wedge g) \\
& = P(B) \sum_a P(w|a) P(g|a) \sum_e P(e) P(a|B \wedge e)
\end{aligned}
$$

Restrict a factor

$f_1(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

$\rightarrow$

$f_2(Y, Z)$:

Y	Z	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

$f_3(Y)$:

Y	val
t	0.9
f	0.8

$f_4() = 0.8$

- What is $f_2(Y, Z) = f_1(x, Y, Z)$?
- What is $f_3(Y) = f_2(Y, \neg z)$?
- What is $f_4() = f_3(\neg y)$?

Sum out a variable

- To eliminate hidden variables, we need to find all factors containing the variable.
- Multiply the factors together, and sum out the variable from the product.
- The sum out operation is similar to the sum rule of probability.
- The rule will derive a new factor without the sum-out variable.

Sum out a variable

Sum out a variable.

Summing out X_1 with domain $\{v_1, \ldots, v_k\}$ from factor $f(X_1, \ldots, X_j)$, produces a factor on $X_2, \ldots, X_j$ defined by:

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

→ Sum out a and e.

$$P(B \wedge w \wedge g)$$
$$= P(B) \sum_a P(w|a)P(g|a) \sum_e P(e)P(a|B \wedge e)$$

Sum out a variable

$f_1(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

$\rightarrow$ $f_2(X, Z)$:

X	Z	val
t	t	0.57
t	f	0.43
f	t	0.54
f	f	0.46

What is $f_2(X, Z) = \sum_Y f_1(X, Y, Z)$?

Multiplying factors

- To eliminate hidden variables, we need to find all factors containing the variable.
- Multiply the factors together, and sum out the variable from the product.
- Multiplication is the first step of the procedure.
- The new factor is a union of the two sets.

Multiplying factors

Multiply two factors together.

The **product** of factors $f_1(X, Y)$ 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:

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

$\rightarrow$

$$\begin{aligned} & P(B \land w \land g) \\ & = P(B) \sum_a P(w|a) P(g|a) \sum_e P(e) P(a|B \land e) \end{aligned}$$

Multiplying factors

f_1:

X	Y	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

f_2:

Y	Z	val
t	t	0.3
t	f	0.7
f	t	0.6
f	f	0.4

$\rightarrow f_1 \times f_2$:

X	Y	Z	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

What is $f_1(X, Y) \times f_2(Y, Z)$?

Normalize a factor

- The purpose of normalization is to convert some numbers into a valid probability distribution
- Normalize is the last step of the variable elimination algorithm
- After normalizing the values, they will sum to 1 and represent valid probabilities

Normalize a factor

- Convert it to a probability distribution.
- Divide each value by the sum of all the values.

f_1:

Y	val
t	0.2
f	0.6

Variable elimination algorithm

To compute $P(X_q | X_{o_1} = v_1 \wedge \ldots \wedge X_{o_j} = v_j)$:

- **Construct a factor** for each conditional probability distribution.
- **Restrict** the observed variables to their observed values.
- Eliminate each hidden variable X_{h_j}.
  - **Multiply** all the factors that contain X_{h_j} to get new factor g_j.
  - **Sum out** the variable X_{h_j} from the factor g_j.
- **Multiply** the remaining factors.
- **Normalize** the resulting factor.

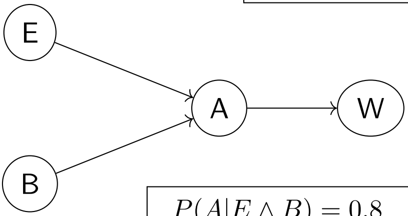
Example of VEA

Given a portion of the Holmes network below,
calculate $P(B|\neg a)$ using the variable elimination algorithm.

Eliminate the hidden variables in reverse alphabetical order.

$$P(E) = 0.1$$

$$P(W|A) = 0.8$$
$$P(W|\neg A) = 0.4$$



$$P(B) = 0.3$$

$$P(A|E \wedge B) = 0.8$$
$$P(A|E \wedge \neg B) = 0.2$$
$$P(A|\neg E \wedge B) = 0.7$$
$$P(A|\neg E \wedge \neg B) = 0.1$$

Example of VEA

- B is the query variable, and A is the evidence variable.
- To calculate $P(B|\neg a)$, it suffices to compute the joint distribution of $P(B, \neg a)$
- Step 0: Define all the factors.
- Step 1: Restrict $A = false$.
- Step 2: Sum out E and W.
- Step 3: Multiply remaining factors.
- Step 4: Normalizing the resulting factor.

More Efficiency

The joint distribution can be written as:

$$P(B \wedge \neg a) = \sum_e \sum_w P(B)p(e)P(\neg a|B \wedge e)P(w| \wedge \neg a)$$

Simplify this term:

$$P(B \wedge \neg a) = P(B)(\sum_e p(e)P(\neg a|B \wedge e))(\sum_w P(w| \wedge \neg a))$$

Define factors

$$P(B), P(E), P(A|B \wedge E), P(W|A)$$

$$\rightarrow f_1(B), f_2(E), f_3(A, B, E), f_4(W, A)$$

$f_1(B)$:

B	val
t	0.3
f	0.7

$f_2(E)$:

E	val
t	0.1
f	0.9

$f_3(A, B, E)$:

A	B	E	val
t	t	t	0.8
t	t	f	0.7
t	f	t	0.2
t	f	f	0.1
f	t	t	0.2
f	t	f	0.3
f	f	t	0.8
f	f	f	0.9

$f_4(W, A)$:

W	A	val
t	t	0.8
t	f	0.4
f	t	0.2
f	f	0.6

Restrict factors

$$f_1(B), f_2(E), f_3(\neg a, B, E), f_4(W, \neg a)$$

$$\to f_1(B), f_2(E), f_5(B, E), f_6(W)$$

$f_1(B)$:

B	val
t	0.3
f	0.7

$f_2(E)$:

E	val
t	0.1
f	0.9

$f_5(B, E)$:

B	E	val
t	t	0.2
t	f	0.3
f	t	0.8
f	f	0.9

$f_6(W)$:

W	val
t	0.4
f	0.6

Sum out E and W

$$f_1(B), f_2(E), f_5(B, E), f_6(W)$$

$$\to f_1(B), f_2(E), f_5(B, E), f_7() = 1.0$$

$f_1(B)$:

B	val
t	0.3
f	0.7

$f_2(E)$:

E	val
t	0.1
f	0.9

$f_5(B, E)$:

B	E	val
t	t	0.2
t	f	0.3
f	t	0.8
f	f	0.9

$f_7()$:

val
1.0

Sum out E and W

$$f_1(B), f_2(E), f_5(B,E), f_7() = 1.0$$
$$\rightarrow f_1(B), f_8(B,E), f_7() = 1.0$$

$f_1(B)$:

B	val
t	0.3
f	0.7

$f_8(B,E) = f_5(B,E) \times f_2(E)$:

B	E	val
t	t	0.2 * 0.1 = 0.02
t	f	0.3 * 0.9 = 0.27
f	t	0.8 * 0.1 = 0.08
f	f	0.9 * 0.9 = 0.81

$f_7()$:

val
1.0

Sum out E and W

$$f_1(B), f_8(B,E), f_7() = 1.0$$
$$\rightarrow f_1(B), f_9(B), f_7() = 1.0$$

$f_1(B)$:

B	val
t	0.3
f	0.7

$f_9(B) = \sum_E f_8(B,E)$:

B	val
t	$0.02 + 0.27 = 0.29$
f	$0.81 + 0.08 = 0.89$

$f_7()$:

val
1.0

Multiply all factors

$$f_1(B), f_9(B), f_7() = 1.0$$

$$\to f_{10}(B)$$

$f_{10}(B) = f_9(B) \times f_1(B) \times f_7()$:

B	val
t	0.29 * 0.3 = 0.087
f	0.89 * 0.7 = 0.623

Normalization

Normalization for the factor:

$p(B|\neg a) = f_{10}(B)$:

B	val
t	0.087 / (0.087 + 0.623) = 0.1225
f	0.623 / (0.087 + 0.623) = 0.8775

Effect of The Elimination Ordering

In general, VEA is exponential in space and time.

The complexity of VEA depends on:

- The size of the CPT in the Bayesian network.
- The size of the largest factor during algorithm execution.

Effect of the elimination ordering on algorithm complexity:

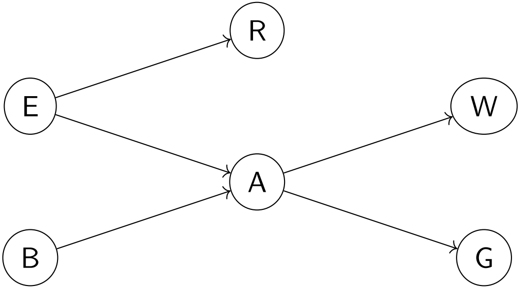
- Every order yields a valid algorithm.
- Different orderings yields different intermediate factors.

Examples of Good and Bad Orderings

Suppose that we want to calculate $P(G)$.
What factors do we produce if we

- Eliminate $R \to W \to E \to B \to A$?
- Eliminate $A \to B \to E \to R \to W$?

Which ordering leads to worse complexity for VEA?



Examples of Good and Bad Orderings

Eliminate $R \to W \to E \to B \to A$?

$f_1(E), f_2(E, R), f_3(B), f_4(E, B, A), f_5(W, A), f_6(G, A)$

Examples of Good and Bad Orderings

Eliminate $R \to W \to E \to B \to A$?

$f_1(E), f_2(E,R), f_3(B), f_4(E,B,A), f_5(W,A), f_6(G,A)$

- $f_2(E,R) \to f_7(E)$: 1 * 2 = 2 ops
- $f_5(W,A) \to f_8(A)$: 1 * 2 = 2 ops
- $f_1(E)f_7(E)f_4(E,B,A) \to f_9(B,A)$: (4 + 1) * 4 = 20 ops
- $f_9(B,A)f_8(A) \to f_{10}(A)$: (2 + 1) * 2 = 6 ops
- $f_{10}(A)f_6(G,A) \to f_{11}(G)$: (2 + 1) * 2 = 6 ops
- Total: 36 ops including additions and multiplication

Examples of Good and Bad Orderings

Eliminate $A \to B \to E \to R \to W$

$f_1(E), f_2(E, R), f_3(B), f_4(E, B, A), f_5(W, A), f_6(G, A)$

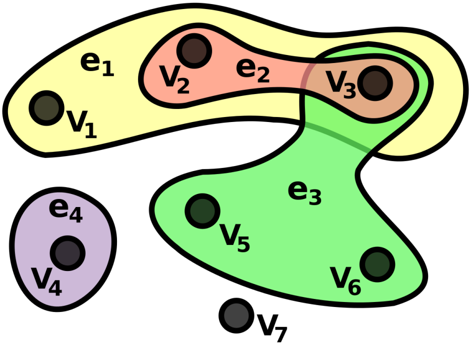
Examples of Good and Bad Orderings

Eliminate $A \to B \to E \to R \to W$

$f_1(E), f_2(E, R), f_3(B), f_4(E, B, A), f_5(W, A), f_6(G, A)$

- $f_4(E, B, A) f_5(W, A) f_6(G, A) \to f_7(E, B, W, G)$: (4 + 1) * 16 = 80 ops
- $f_7(E, B, W, G) f_3(B) \to f_8(E, W, G)$: (2 + 1) * 8 = 24 ops
- $f_1(E) f_8(E, W, G) f_2(E, R) \to f_9(W, G, R)$: (4 + 1) * 8 = 24 ops
- $f_9(W, G, R) \to f_{10}(W, G)$: 1 * 4 = 4 ops
- $f_{10}(W, G) \to f_{11}(G)$: 1 * 2 = 2 ops
- Total: 134 ops including additions and multiplication

Hypergraph



- A generalization graph, which contains vertices and a set of hyperedges. A hyperedge connects multiple vertices.
- A hyperedge is a clique of several vertices.
- A Bayesian Network can be seen as a hypergraph, where each factor is basically a hyperedge.

Elimination Width

- Given an ordering π of the variables and an initial hypergraph $\mathcal{H}$, eliminating these variables yields a sequence of hypergraphs:

$$\mathcal{H} \rightarrow H_0, H_1, \cdots, H_n$$

- where H_n contains only one vertex
- The elimination width is the maximum size of any hyperedge in any of the hypergraphs $H_0, H_1, \cdots, H_n$.
- The elimination width is 4 for the hypergraph $f_7(E, B, W, G)$.

Elimination Width

- If the elimination width of an ordering π is k, then the complexity of VE using that order is $2^{O(k)}$
- Elimination width k means that at some stage in the elimination process, a factor involving k variables was generated.
- That factor will require $2^{O(k)}$ space to store.
- And it will require $2^{O(k)}$ operations to process.
- The time and space complexity of VE are both exponential w.r.t elimination width.

Tree Width

- Given a hypergraph $\mathcal{H}$ with vertices $\{X_1, X_2, \cdots, X_n\}$ the tree width of $\mathcal{H}$ is the MINIMUM elimination width of any orderings π of these variables.
- Thus VE has best case complexity of $2^{O(\omega)}$ where ω is the tree width of the initial Bayes Net.
- In the worst case the treewidth can be equal to the number of variables.

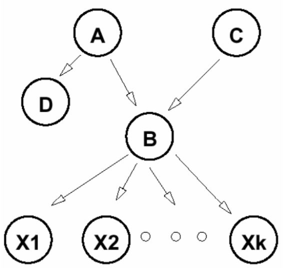
Tree Width

- VE complexity is exponential in the treewidth.
- Finding an ordering that has an elimination width equal to treewidth is NP-hard.
- Heuristics are used to find good orderings with low elimination widths.
- In practice, this can be very successful. Elimination widths can be often relatively small, 8-10 even when the network has 1000s of variables.
- Thus VE can be much more efficient than simply summing up the probability of all the possible events.

Finding Good Orderings

- A **polytree** is a single-connected network in which there is at most one undirected path between any two nodes.
- A node can have multiple parents, but they have no cycles.
- Good orderings are easy to find for polytrees.
  - At each stage, eliminate a singly connected node.
  - Because we have a polytree, we are assured that a singly connected node will always exist.
  - The size of the factors never increases.

Finding Good Orderings



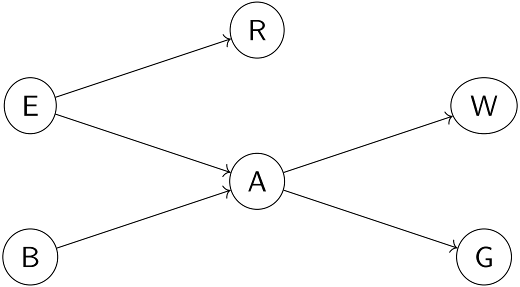
- Treewidth of the polytree is equal to the maximum number of parents among all the nodes.
- Eliminating singly connected nodes allows VE to run the time linear in size of the network.
  - e.g. Eliminating D, A, C, X1, etc.
  - Results: No factor ever larger than original CPTs.

Approximate Inference

We cannot make exact inferences in many cases where the graph is too large. Instead, we use approximate inferences with sampling to understand the distribution of different variables.

- Forward Sampling
- Rejection Sampling

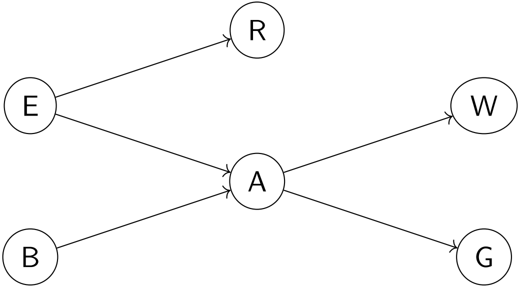
Forward Sampling



Estimating $p(G)$:

- Loop for i in $1 \cdots n$, do
  - Sample $e_i \sim P(E)$
  - Sample $b_i \sim P(B)$
  - Sample $a_i \sim P(A|E = e, B = b)$
  - Sample $g_i \sim p(G|A = a)$
- $p(G = True) = \frac{\sum_i g_i}{n}$

Rejection Sampling



Estimating $p(G|\neg b)$

- Loop for i in $1 \cdots n$, do
  - Sample e_i, b_i, a_i, g_i as before.
  - Reject $b_i =$ True, only take accepted $\tilde{g}_i$.
- $p(G = True) = \frac{\sum_i \tilde{g}_i}{n}$

Revisiting Learning Goals

- Explain how we can perform probabilistic inference more efficiently using the variable elimination algorithm.
- Define factors. Manipulate factors using operations restrict, sum out, multiply and normalize.
- Describe/trace/implement the variable elimination algorithm for calculating a prior or a posterior probability given a Bayesian network.
- Explain how the elimination ordering affects the complexity of the variable elimination algorithm.
- Know about approximate statistical inference.