Bayesian Networks

Problem

At a certain time $t$, the KB of an agent is some collection of beliefs
At time $t$ the agent’s sensors make an observation that changes the strength of one of its beliefs
How should the agent update the strength of its other beliefs?

Purpose of Bayesian Networks

Facilitate the description of a collection of beliefs by making explicit causality relations and conditional independence among beliefs
Provide a more efficient way (than by using joint distribution tables) to update belief strengths when new evidence is observed

Other Names

Belief networks
Probabilistic networks
Causal networks

Bayesian Networks

A simple, graphical notation for conditional independence assertions resulting in a compact representation for the full joint distribution

Syntax:
- a set of nodes, one per variable
- a directed, acyclic graph (link = ‘direct influences’)
- a conditional distribution for each node given its parents: $P(X|\text{Parents}(X))$
Example

Topology of network encodes conditional independence assertions:

- Weather is independent of other variables.
- Toothache and Catch are independent given Cavity.

Example

I’m at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn’t call. Sometimes it’s set off by a minor earthquake. Is there a burglar?

Variables: Burglar, Earthquake, Alarm, JohnCalls, MaryCalls

Network topology reflects “causal” knowledge:
- A burglar can set the alarm off.
- An earthquake can set the alarm off.
- The alarm can cause Mary to call.
- The alarm can cause John to call.

A Simple Belief Network

Node meanings:
- Burglary
- Earthquake
- Alarm
- JohnCalls
- MaryCalls

Directed acyclic graph (DAG)

Intuitive meaning of arrow from x to y: “x has direct influence on y.”

Nodes are random variables

Assigning Probabilities to Roots

Assigning probabilities to roots:
- Burglary
- Earthquake

Conditional Probability Tables

Size of the CPT for a node with k parents: $2^k$.
What the BN Means

Calculation of Joint Probability

What The BN Encodes

What The BN Encodes

Structure of BN

Construction of BN
Markov Assumption

- We now make this independence assumption more precise for directed acyclic graphs (DAGs).
- Each random variable \( X \) is independent of its non-descendants, given its parents \( \text{Pa}(X) \).
- Formally, \( I(X; \text{NonDesc}(X) \mid \text{Pa}(X)) \).

Inference In BN

- Set \( E \) of evidence variables that are observed, e.g., \( \{\text{JohnCalls, MaryCalls}\} \).
- Query variable \( X \), e.g., Burglary, for which we would like to know the posterior probability distribution \( P(X \mid E) \).

Inference Patterns

- Basic use of a BN: Given new observations, compute the new strengths of some (or all) beliefs.
- Other use: Given the strength of a belief, which observation should we gather to make the greatest change in this belief’s strength.

Singly Connected BN

- A BN is singly connected if there is at most one undirected path between any two nodes.

Types Of Nodes On A Path

Independence Relations In BN

Given a set \( E \) of evidence nodes, two beliefs connected by an undirected path are independent if one of the following three conditions holds:
1. A node on the path is linear and in \( E \).
2. A node on the path is diverging and in \( E \).
3. A node on the path is converging and neither this node, nor any descendant is in \( E \).
Given a set $E$ of evidence nodes, two beliefs connected by an undirected path are independent if one of the following three conditions holds:
1. A node on the path is linear and in $E$
2. A node on the path is diverging and in $E$
3. A node on the path is converging and neither this node, nor any descendant is in $E$

Gas and Radio are independent given evidence on SparkPlugs.

Gas and Radio are independent given evidence on Battery.

Gas and Radio are independent given no evidence, but they are dependent given evidence on Starts or Moves.

What is time complexity to compute $P(X_n)$?
What is time complexity if we computed the full joint?
Variable Elimination

General idea:
- Write query in the form
  \[ P(X, e) = \sum_x \sum_y \prod_i P(x_i | p_{a_i}) \]
- Iteratively
  - Move all irrelevant terms outside of innermost sum
  - Perform innermost sum, getting a new term
  - Insert the new term into the product

\[ \sum \sum \sum \prod = \]

A More Complex Example

“Asia” network:

We want to compute \( P(d) \)
- Need to eliminate: \( v, s, x, t, l, a, b \)

Initial factors

We want to compute \( P(d) \)
- Need to eliminate: \( v, s, x, t, l, a, b \)

Initial factors

Eliminate:

Note: \( f_v(t) = P(t) \)
In general, result of elimination is not necessarily a probability term

Compute:

Summing on \( s \) results in a factor with two arguments \( f_s(b, l) \)
In general, result of elimination may be a function of several variables

We want to compute \( P(d) \)
- Need to eliminate: \( x, t, l, a, b \)

Initial factors

We want to compute \( P(d) \)
- Need to eliminate: \( x, t, l, a, b \)

Initial factors

Eliminate:

Note: \( f_a(s) = 1 \) for all values of \( a \)
Compute: $\sum_{a} \ell_{(b)} \ell_{(d)} P(d | a, b)$

Eliminate: $a, b$

Initial factors

$P(v) P(s | v) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

$
\Rightarrow f(t) P(v) P(s | v) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)
\Rightarrow f(t) f(b) P(a | t, v) P(x | a) P(d | a, b)
\Rightarrow f(t) f(b) f(a) P(a | t, v) P(x | a) P(d | a, b)
\Rightarrow f(t) f(b) f(a) P(d | a, b) \Rightarrow f(d) 
$

We want to compute $P(d)$

We now understand variable elimination as a sequence of rewriting operations

Actual computation is done in elimination step

Computation depends on order of elimination

Dealing with evidence

How do we deal with evidence?

Suppose get evidence $V = t, S = f, D = t$

We want to compute $P(L, V = t, S = f, D = t)$

Variable Elimination

We want to compute $P(d)$

Need to eliminate: $a, b$

Initial factors

$P(v) P(s | v) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) P(s) P(v | s) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) f(a) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) f(a) P(d | a, b) \Rightarrow f(d)$

We want to compute $P(d)$

Need to eliminate: $a, b$

Initial factors

$P(v) P(s | v) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) P(v) P(s | v) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) f(a) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) f(a) P(d | a, b) \Rightarrow f(d)$

Variable Elimination

We want to compute $P(d)$

Need to eliminate: $a, b$

Initial factors

$P(v) P(s | v) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) P(s) P(v | s) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) f(a) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) f(a) P(d | a, b) \Rightarrow f(d)$

We want to compute $P(d)$

Need to eliminate: $a, b$

Initial factors

$P(v) P(s | v) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) P(s) P(v | s) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) f(a) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) f(a) P(d | a, b) \Rightarrow f(d)$

We want to compute $P(d)$

Need to eliminate: $a, b$

Initial factors

$P(v) P(s | v) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) P(s) P(v | s) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) f(a) P(a | t, v) P(x | a) P(d | a, b)$

$\Rightarrow f(t) f(b) f(a) P(d | a, b) \Rightarrow f(d)$

Dealing with Evidence

We start by writing the factors:

$P(v) P(s | v) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

Since we know that $V = t$, we don't need to eliminate $V$

Instead, we can replace the factors $P(V)$ and $P(T | V)$ with $f_{(V)} = P(V = t)$ and $f_{(T|V)} = P(T | V = t)$

These “select” the appropriate parts of the original factors given the evidence

Note that $f_{(V)}$ is a constant, and thus does not appear in elimination of other variables

Dealing with Evidence

We start by writing the factors:

$P(v) P(s | v) P(t | s) P(b | s) P(a | t, v) P(x | a) P(d | a, b)$

Since we know that $V = t$, we don't need to eliminate $V$

Instead, we can replace the factors $P(V)$ and $P(T | V)$ with $f_{(V)} = P(V = t)$ and $f_{(T|V)} = P(T | V = t)$

These “select” the appropriate parts of the original factors given the evidence

Note that $f_{(V)}$ is a constant, and thus does not appear in elimination of other variables
Dealing with Evidence

- Given evidence \( V = t, S = f, D = t \)
- Compute \( P(L, V = t, S = f, D = t) \)
- Initial factors, after setting evidence:

\[
f_{v_{11}}(t)f_{v_{12}}(t)f_{s_{12}}(b)p(a | t, l)p(x | a)f_{d_{12}}(o)(a, b)
\]

- Eliminating \( x \), we get

\[
f_{v_{11}}(t)f_{v_{12}}(t)f_{s_{12}}(b)p(a | t, l)f_x(a)f_{d_{12}}(o)(a, b)
\]

- Eliminating \( t \), we get

\[
f_{v_{11}}(t)f_{s_{12}}(b)f_x(a)f_x(a)f_{d_{12}}(o)(a, b)
\]

- Eliminating \( a \), we get

\[
f_{v_{11}}(t)f_{s_{12}}(b)f_x(b, l)
\]

Variable Elimination Algorithm

- Let \( X_1, \ldots, X_m \) be an ordering on the non-query variables

\[
\sum_{X_1} \cdots \sum_{X_m} \prod_{(X_i \rightarrow \text{Parents } (X_j))} P(X_i | \text{Parents } (X_j))
\]

- For \( i = m, \ldots, 1 \)
  - Leave in the summation for \( X_i \) only factors mentioning \( X_i \)
  - Multiply the factors, getting a factor that contains a number for each value of the variables mentioned, including \( X_i \)
  - Sum out \( X_i \), getting a factor \( f \) that contains a number for each value of the variables mentioned, not including \( X_i \)
  - Replace the multiplied factor in the summation
Complexity of variable elimination

Suppose in one elimination step we compute
\[ f'_x(y_1, \ldots, y_k) = \sum_s f_s(x, y_1, \ldots, y_k) \]
\[ f'_x(y_1, \ldots, y_k) = \prod_{i=1}^m f_i(x, y_1, \ldots, y_k) \]
This requires
\[ m|\text{Val}(X)| \cdot \prod |\text{Val}(X)| \text{ multiplications} \]
- For each value of \(x, y_1, \ldots, y_k\), we do \(m\) multiplications.
\[ |\text{Val}(X)| \cdot \prod |\text{Val}(X)| \text{ additions} \]
- For each value of \(y_1, \ldots, y_k\), we do \(|\text{Val}(X)|\) additions.

Complexity is exponential in number of variables in the intermediate factor!

Understanding Variable Elimination

We want to select "good" elimination orderings that reduce complexity.

This can be done by examining a graph theoretic property of the "induced" graph; we will not cover this in class.

This reduces the problem of finding good ordering to graph-theoretic operation that is well-understood—unfortunately computing it is NP-hard!

Approaches to inference

- Exact inference
  - Inference in Simple Chains
  - Variable elimination
  - Clustering / join tree algorithms
- Approximate inference
  - Stochastic simulation / sampling methods
  - Markov chain Monte Carlo methods

Stochastic simulation - direct

Suppose you are given values for some subset of the variables, \(G\), and want to infer values for unknown variables, \(U\).
- Randomly generate a very large number of instantiations from the BN.
- Generate instantiations for all variables—start at root variables and work your way "forward."
- Rejection Sampling: keep those instantiations that are consistent with the values for \(G\).
- Use the frequency of values for \(U\) to get estimated probabilities.
- Accuracy of the results depends on the size of the sample (asymptotically approaches exact results).

Direct Stochastic Simulation

\( P(\text{WetGrass}|\text{Cloudy})? \)

\( P(\text{WetGrass}|\text{Cloudy}) = P(\text{WetGrass} \land \text{Cloudy}) / P(\text{Cloudy}) \)

1. Repeat \(N\) times:
   1.1. Guess Cloudy at random
   1.2. For each guess of Cloudy, guess Sprinkler and Rain, then WetGrass
2. Compute the ratio of the \# runs where WetGrass and Cloudy are True over the \# runs where Cloudy is True

Exercise: Direct sampling

Topological order = ...

Random number generator: .35, .76, .51, .44, .08, .28, .03, .92, .02, .42
Likelihood weighting

- Idea: Don’t generate samples that need to be rejected in the first place!
- Sample only from the unknown variables Z
- Weight each sample according to the likelihood that it would occur, given the evidence E

Markov chain Monte Carlo algorithm

- So called because
  - Markov chain - each instance generated in the sample is dependent on the previous instance
  - Monte Carlo – statistical sampling method
- Perform a random walk through variable assignment space, collecting statistics as you go
  - Start with a random instantiation, consistent with evidence variables
  - At each step, for some nonevidence variable, randomly sample its value, consistent with the other current assignments
- Given enough samples, MCMC gives an accurate estimate of the true distribution of values

Applications

- Medical diagnosis, e.g., lymph-node diseases
- Fraud/uncollectible debt detection
- Troubleshooting of hardware/software systems

Applications

http://excalibur.brc.uconn.edu/~baynet/researchApps.html