Inference in Bayesian networks

Devika Subramanian
Comp 440
Lecture 7

Exact inference in Bayesian networks

- Inference by enumeration
- The variable elimination algorithm
Probabilistic inference using Bayesian networks

- A single mechanism can account for a wide range of inferences under uncertainty.
  - Diagnostic inference (from effects to causes).
    - Example: $P(B|J) = 0.016$
  - Causal inference (from causes to effects).
    - Example: $P(J|B) = 0.875$

Probabilistic inference

- Inter-causal inference (between causes of a common effect).
  - Example: $P(B|A) = 0.376$ and $P(B|A,E) = 0.003$.
  Even though B and E are independent, the presence of one makes the other unlikely. This phenomenon is called “explaining away”. Cannot be captured in logic which is monotonic.
Probabilistic inference

- **Mixed inferences** (combining two or more of the above).
  - Example: $P(A|J, \text{not } E) = 0.03$ is a simultaneous use of diagnostic and causal inference.

Inference by enumeration

- To compute the probability of variable $Q$ given evidence $E$ ($E_1, \ldots, E_k$), we use the rule of conditional independence:

  $$P(Q|E) = \frac{P(Q,E)}{P(E)}$$

  Each of these terms can be computed by summing terms from the full joint distribution.
Example

- \( P(B|J) = \frac{P(B,J)}{P(J)} \)

\[
P(B,J) = \sum_{A \in E, M} P(B,J,A,E,M) = \sum_{A \in E, M} P(B)P(J | A)P(A | B,E)P(E)P(M | A)
\]

\[
= P(B) \sum_{E} P(E) \sum_{A} P(A | B,E)P(J | A) \sum_{M} P(M | A)
\]

\[
= P(B) \sum_{E} P(E) \sum_{A} P(A | B,E)P(J | A)
\]

\[
= P(B)[P(E)(P(A | B,E)P(J | A) + P(\overline{A} | B,E)P(J | \overline{A})) + \\
  P(\overline{E})(P(A | B,\overline{E})P(J | A) + P(\overline{A} | B,\overline{E})P(J | \overline{A})]
\]

\[
= 0.001[0.002(0.95*0.9 + 0.05*0.05) + 0.998(0.95*0.9 + 0.05*0.05)]
\]

= 0.0008575

(c) Devika Subramanian, 2006

The expression tree

Bottom up computation

(c) Devika Subramanian, 2006
Example (contd.)

\[ P(J) = P(B, J) + P(\overline{B}, J) \]
\[ P(\overline{B}, J) = 0.999[0.002(0.29 \times 0.9 + 0.71 \times 0.05) + 0.998(0.001 \times 0.9 + 0.999 \times 0.05)] \]
\[ = 0.0513413 \]
\[ P(B \mid J) = 0.0164 \]

Any query can be answered by computing sums of products of conditional probabilities from the network.

Inter-causal inference

\[ P(B \mid A) = \frac{P(B, A)}{P(A)} \]
\[ P(B, A) = P(B) \sum_{E} P(E)P(A \mid B, E) \sum_{J} P(J \mid A) \sum_{M} P(M \mid A) \]
\[ = P(B) \sum_{E} P(E)P(A \mid B, E) \sum_{J} P(J \mid A) \]
\[ = P(B) \sum_{E} P(E)P(A \mid B, E) \]
\[ = P(B)[P(E)P(A \mid B, E) + P(\overline{E})P(A \mid B, \overline{E})] \]
\[ = 0.001[0.002 \times 0.95 + 0.998 \times 0.95] \]
\[ = 0.00095 \]
Inter-causal inference

\[ P(B, A) = P(B)[P(E)P(A \mid B, E) + P(\overline{E})P(A \mid B, \overline{E})] \]
\[ = 0.999[0.002 \times 0.29 + 0.998 \times 0.001] \]
\[ = 0.00158 \]
\[ P(B \mid A) = 0.0095 / (0.0095 + 0.00158) = 0.376 \]

Inter-causal inference

\[ P(B \mid A, E) = P(B, A, E) / P(A, E) \]
\[ P(B, A, E) = P(B)P(E)P(A \mid B, E) \sum_{J} P(J \mid A) \sum_{M} P(M \mid A) \]
\[ = P(B)P(E)P(A \mid B, E) \]
\[ = 0.001 \times 0.002 \times 0.95 = 0.0000019 \]
\[ P(B, A, E) = P(B)P(E)P(A \mid B, E) \]
\[ = 0.999 \times 0.002 \times 0.29 = 0.000579 \]
\[ P(B \mid A, E) = 0.0000019 / (0.0000019 + 0.000579) \]
\[ = 0.003 \]
Variable elimination

- Basic inference: a 3-chain

\[
P(C = c) = \sum_b P(C = c \mid B = b) \sum_a P(B = b \mid A = a) P(A = a)
\]

If there are \( k \) values for \( A \) and \( B \), what is the complexity of this calculation?

---

Variable elimination

- Basic inference: a 3-chain

\[
P(C = c) = \sum_b P(C = c \mid B = b) \sum_a P(B = b \mid A = a) P(A = a)
\]

\[
P(C = c) = \sum_b P(C = c \mid B = b) P(B = b)
\]

\[
P(B = b) = \sum_a P(B = b \mid A = a) P(A = a)
\]

Store and do not recompute!
Variable elimination

- Basic inference: an n-chain takes $O(nk^2)$ computation

$$P(D = d) = \sum_c P(D = d \mid C = c) \sum_b P(C = c \mid B = b) \sum_a P(B = b \mid A = a)P(A = a)$$

$$f_1(b) = \sum_a P(B = b \mid A = a)P(A = a)$$

$$f_2(c) = \sum_b P(C = c \mid B = b)f_1(b)$$

$$f_3(d) = \sum_c P(D = d \mid C = c)f_2(c)$$

Singly connected networks

- Singly connected networks (exactly one undirected path between any two nodes in network)

What is $P(X \mid Z, Y_1, ..., Y_n)$?  

Causal support for $X$  

Evidential support for $X$
Multiply connected networks

\[ P(C) = 0.5 \]
\[ P(R|C)=0.8 \]
\[ P(R|\text{not } C)=0.2 \]
\[ P(S|C)=0.1 \]
\[ P(S|\text{not } C)=0.5 \]
\[ P(W|R,S)=0.99 \]
\[ P(W|\text{not } R,S)=0.90 \]
\[ P(W|R,\text{not } S)=0.90 \]
\[ P(W|\text{not } R, \text{not } S)=0.00 \]

Compute \( P(\text{Wet grass}) \) using this network.

Variable elimination again

\[
P(W) = \sum_{R,S} P(W | R,S) \sum_{C} P(R | C) P(S | C) P(C)
\]
\[
f_1(R,S) = \sum_{C} P(R | C) P(S | C) P(C)
\]
\[
f_2(W) = \sum_{R,S} P(W | R,S) f_1(R,S)
\]

We exploit the structure of the belief network and break up the computation into two pieces as above. We use a form of dynamic programming to remember values of the inner sum to prevent re-computing it.
The variable elimination algorithm

- To evaluate \( \sum_{X_1} \sum_{X_2} \ldots \sum_{X_n} \prod_{j} P(X_j | \text{Parents}(X_j)) \)

- For \( i = m \) to 1 do
  - Group the terms in which \( X_i \) occurs and construct a factor \( f \) that sums over \( X_i \). \( f \) will be indexed by all other variables that occur in those terms.
  - Replace the sum in the original expression by the factor constructed above.

Example

\[
P(W) = \sum_{R,S,C} P(W | R,S)P(R | C)P(S | C)P(C)
= \sum_{R,S} P(W | R,S) \sum_{C} P(S | C)P(R | C)P(C)
\]

Eliminate \( C \) first, then \( R \) and \( S \)

\[
f_1(R,S) = \sum_{C} P(S | C)P(R | C)P(C)
\]

\[
P(W) = \sum_{R,S} P(W | R,S)f_1(R,S)
\]

Eliminate \( R \) and \( S \) first, then \( C \).

\[
f_2(C) = \sum_{R,S} P(W | R,S)P(S | C)P(R | C)
\]

\[
P(W) = \sum_{C} P(C)f_2(C)
\]
Complexity of variable elimination

- For singly connected networks, time and space complexity of inference is linear in the size of the network.
- For multiply connected networks, time and space complexity is exponential in the size of the network in the worst case. Exact inference is \#P-hard.
- Practically, choosing a good order to eliminate variables in, makes process tractable.

Clustering

- Idea: transform network to a singly connected network by combining nodes.

\begin{align*}
P(C) &= 0.5 \\
P(R|C) &= 0.08 \\
P(R, not S|C) &= 0.72 \\
P(not R, S|C) &= 0.02 \\
P(not R, not S|C) &= 0.18 \\
P(R, S|not C) &= 0.1 \\
P(R, not S|not C) &= 0.1 \\
P(not R, S|not C) &= 0.4 \\
P(not R, not S|not C) &= 0.4 \\
P(C) &= 0.5 \\
P(W|R,S) &= 0.99 \\
P(W|not R,S) &= 0.90 \\
P(W|R, not S) &= 0.90 \\
P(W|not R, not S) &= 0.00 \\
\end{align*}
Properties of clustering

- Exact method for evaluating belief networks that are not singly connected.
- Choosing good nodes to cluster is difficult; the same issues as in determining good variable ordering. Further, CPTs at clustered nodes are exponential in size (in the number of nodes clustered there).

Summary

- Belief networks are a compact encoding of the full joint probability distribution over n variables that makes conditional independence assumptions between these variables explicit.
- We can use belief networks to exactly compute any probability of interest over the given variables.
- Exact inference is intractable for multiply connected networks.
Approximate inference

- Direct sampling
- Rejection sampling
- Likelihood weighting
- Markov Chain Monte Carlo (MCMC)

Direct sampling

- Idea: generate samples \((x_1, \ldots, x_n)\) from the joint distribution specified by the network.
- For each \(x_i\), draw a sample according to \(P(x_i | \text{parents}(x_i))\).

\[
S_{DS}(x_1, \ldots, x_n) = \prod P(x_i | \text{Parents}(x_i))
\]

- The probability of \((x_1, \ldots, x_n)\) is simply the number of times of \((x_1, \ldots, x_n)\) is generated divided by the total number of samples.
Direct sampling method

- Choose a value for the root nodes weighted by their priors.
- Use CPTs for direct descendants of roots (given values of root nodes) to choose values for them.
- Do previous step all the way down to the leaves.

Direct Sampling

- $P(C) = 0.5$
- $P(R|C) = 0.8$, $P(R|\neg C) = 0.2$
- $P(S|C) = 0.1$, $P(S|\neg C) = 0.5$
- $P(W|R,S) = 0.99$, $P(W|\neg R,S) = 0.90$, $P(W|R,\neg S) = 0.90$, $P(W|\neg R, \neg S) = 0.00$
Direct Sampling

\[
P(C) = 0.5
\]

\[
P(R|C) = 0.8 \\
P(R|\text{not } C) = 0.2
\]

\[
P(S|C) = 0.1 \\
P(S|\text{not } C) = 0.5
\]

\[
P(W|R,S) = 0.99 \\
P(W|\text{not } R,S) = 0.90 \\
P(W|R,\text{not } S) = 0.90 \\
P(W|\text{not } R, \text{not } S) = 0.00
\]
Direct Sampling

\[ P(C) = 0.5 \]

\[ P(R|C) = 0.8 \quad P(R|\text{not } C) = 0.2 \]

\[ P(S|C) = 0.1 \quad P(S|\text{not } C) = 0.5 \]

\[ P(W|R, S) = 0.99 \quad P(W|\text{not } R, S) = 0.90 \]
\[ P(W|R, \text{not } S) = 0.90 \quad P(W|\text{not } R, \text{not } S) = 0.00 \]
Example

- To compute \( P(W) \) in sprinkler network
  - Choose a value for \( C \) with prior \( P(C) = 0.5 \); assume we pick \( C = \text{false} \).
  - Choose a value for \( S \): \( P(S|\text{not } C) = 0.5 \); assume we pick \( S = \text{true} \). Choose a value for \( R \): \( P(R|\text{not } C) = 0.2 \); assume we pick \( R = \text{false} \).
  - Choose a value for \( W \) drawn according to \( P(W|S, \text{not } R) = 0.9 \); assume we pick \( W = \text{true} \).
  - We have generated the event (not \( C \), \( S \), not \( R \), \( W \)).
  - Repeat this process and calculate the fraction of events in which \( W \) is true.

Rejection sampling

- Used to estimate \( P(X|e) \) in Bayesian networks.
- Generate samples from the full joint distribution using direct sampling.
- Eliminate all samples that don’t match \( e \).
- Estimate \( P(X|e) \) as the fraction of samples with \( X = x \) from the remaining samples.
**Rejection sampling example**

- To find \( P(\text{Rain}|\text{Sprinkler} = \text{true}) \), generate 100 samples by direct sampling of the network.
- Say, 27 of them have \( \text{Sprinkler} = \text{true} \).
- 8 of the 27 have \( \text{Rain} = \text{true} \).
- So we estimate above conditional probability as \( 8/27 = 0.296 \)

---

**Likelihood weighting**

- Avoids inefficiency of rejection sampling, by only generating events that are consistent with evidence \( e \).
- To find \( P(X|e) \), the algorithm fixes \( e \), and samples \( X \) and the remaining variables \( Y \) in the network.
- Each event is weighted by the probability of its occurrence.
Likelihood weighting

- $P(C) = 0.5$
- $P(R|C) = 0.8$
- $P(R|\text{not } C) = 0.2$
- $P(S|C) = 0.1$
- $P(S|\text{not } C) = 0.5$
- $P(W|R,S) = 0.99$
- $P(W|\text{not } R,S) = 0.90$
- $P(W|R,\text{not } S) = 0.90$
- $P(W|\text{not } R, \text{not } S) = 0.00$

Let $w = 1.0$

$P(R|S,W) = ?$
Likelihood weighting

\[ P(C) = 0.5 \]

\[ P(R|C) = 0.8 \quad P(R|\text{not } C) = 0.2 \]

\[ P(S|C) = 0.1 \quad P(S|\text{not } C) = 0.5 \]

\[ P(W|R,S) = 0.99 \quad P(W|\text{not } R,S) = 0.90 \quad P(W|R,\text{not } S) = 0.90 \quad P(W|\text{not } R, \text{not } S) = 0.00 \]

\[ w = 1.0 \times 0.1 \]

\[ P(R|S,W) = ? \]
Likelihood weighting example

- \( P(\text{Rain}|\text{Sprinkler}, \text{WetGrass}) = ? \)
- \( w = 1.0 \)
- Sample from \( P(\text{Cloudy}) = <0.5, 0.5> \); suppose it returns \( \text{Cloudy} = \text{true} \).
- Sprinkler is an evidence variable with value true. We set \( w = w \times P(\text{Sprinkler}|\text{Cloudy}) = 0.1 \)
- Sample from \( P(\text{Rain}|\text{Cloudy}) = <0.8, 0.2> \). Let say we get \( \text{Rain} = \text{true} \).
- WetGrass is an evidence variable, so we set \( w = w \times P(\text{WetGrass}|\text{Sprinkler}, \text{Rain})=0.099 \)
- We have generated the event \( (\text{Cloudy}, \text{Sprinkler}, \text{Rain}, \text{WetGrass}) \) with weight 0.099.
Why likelihood weighting works

- Sampling distribution $S_W$ is
  \[ S_W(z,e) = \prod_i P(z_i \mid \text{Parents}(Z_i)) \]

- Likelihood weighting $w(z,e)$
  \[ w(z,e) = \prod_i P(e_i \mid \text{parents}(E_i)) \]

Note that $S_W(z,e) \times w(z,e) = P(z,e)$

Markov processes: a quick intro

- We are interested in predicting weather, and for the purposes of this example, weather can take on one of three values: \{sunny, rainy, cloudy\}.
- The weather on a given day is dependent only on the weather on the previous day.
  \[ P(w_i \mid w_{i-1}, \ldots, w_1) = P(w_i \mid w_{i-1}) \]

This is the Markov property.
Markov process example

- We have knowledge of the transition probabilities between the various states: $q(s,s')$. $q$ is called the transition kernel.

\[
\begin{bmatrix}
0.50 & 0.25 & 0.25 \\
0.50 & 0.00 & 0.50 \\
0.25 & 0.25 & 0.50 \\
\end{bmatrix}
\]

Prediction

- Suppose day 1 is rainy. We will represent this as a vector of probabilities over the three values.

\[
\pi(1) = [0 \ 1 \ 0];
\]

- How do we predict the weather for day 2 given $\pi(1)$ and the transition kernel $q$?

- From the transition kernel, we can see that the probability of day 2 being sunny is .5, and that the probabilities for being cloudy or rainy are 0.25 each.

\[
\pi(2) = \pi(1) \ast q = [0.5 \ 0 \ 0.5];
\]
Prediction (contd.)

- We can calculate the distribution of weather at time $t+1$ given the distribution for time $t$.

$$
\pi(t + 1) = \pi(t) \cdot q
= (\pi(t-1) \cdot q) \cdot q
= \pi(1) \cdot q^t
$$

Prediction

- What’s the weather going to be like on the 3rd, 5th, 7th and 9th days?

$$
\pi(3) = \pi(1) \cdot q^2 = [0.375 \ 0.25 \ 0.375]
\pi(5) = \pi(1) \cdot q^4 = [0.3984 \ 0.2031 \ 0.3984]
\pi(7) = \pi(1) \cdot q^6 = [0.3999 \ 0.2002 \ 0.3999]
\pi(9) = \pi(1) \cdot q^8 = [0.4 \ 0.2 \ 0.4]
\pi(t) = [0.4 \ 0.2 \ 0.4], \text{ for all } t \geq 9
$$
A new start state

- Let the weather on day 1 be sunny.
- How does the distribution of weather change with time?

\[ \pi(1) = [1 \ 0 \ 0] \]
\[ \pi(3) = \pi(1) * q^2 = [0.4375 \ 0.1875 \ 0.375] \]
\[ \pi(5) = \pi(1) * q^4 = [0.4023 \ 0.1992 \ 0.3984] \]
\[ \pi(7) = \pi(1) * q^6 = [0.4001 \ 0.2000 \ 0.3999] \]
\[ \pi(9) = \pi(1) * q^8 = [0.4 \ 0.2 \ 0.4] \]
\[ \pi(t) = [0.4 \ 0.2 \ 0.4], \text{ for all } t \geq 9 \]

Stationary distribution

- Independent of the start state, this Markov process converges to a stationary distribution \([0.4 \ 0.2 \ 0.4]\) in the limit.
- The stationary distribution \(p^*\) is the solution to the equation \(p^* \cdot q = p^*\).
Sampling from a Markov chain

- We can sample from the discrete distribution [0.4 0.2 0.4] as follows
  - Start the Markov chain at a random state at time 1.
  - Use the transition kernel \( q \) to generate a state at time \( t+1 \), given the value of the state at time \( t \).
  - Keep repeating above step to generate a long chain.
  - After eliminating an initial prefix of the chain (burn-in), use the rest as samples from the above distribution.

When does this work?

- As \( t \to \infty \), a Markov chain converges to a unique stationary distribution if it is
  - Irreducible (every state in the state space is reachable from every other state).
  - Has no periodic cycles
  - The stationary distribution \( \pi \) satisfies
    \[
    \pi(s') = \sum_s \pi(s)q(s, s')
    \]
  - Such a Markov chain is called ergodic, and the above theorem is called the ergodicity theorem.
Detailed balance equation

\[ \pi(s)q(s,s') = \pi(s')q(s',s) \]

The detailed balance equation implies stationarity.

\[ \sum_s \pi(s)q(s,s') = \sum_s \pi(s')q(s',s) = \pi(s') \]

Designing an MCMC sampler

- Need to find \( q(s,s') \) that satisfies the detailed balance equation with respect to the probability of interest, say \( P(x|e) \).
Gibbs sampler

- Each variable is sampled conditionally on the current values of all the other variables.
- Example: sampling from a 2d distribution by sampling first coordinate from a 1d conditional distribution, and then sampling the second coordinate from another 1d conditional distribution.

Gibbs sampling and Bayesian networks

- Let $X_i$ be the variable to be sampled and let $Y$ be all the hidden variables other than $X_i$. Let their current values be $x_i$ and $y$ respectively. We will sample a new value for $X_i$ conditioned on all the other variables including the evidence.
- $q(x,x')=q((x_i,y),(x_i',y))=P(x_i'|y,e)$
Gibbs sampling works!

\[ \pi(x)q(x, x') = P(x \mid e)P(x_i' \mid y, e) = P(x_i, y \mid e)P(x_i' \mid y, e) \]
\[ = P(x_i \mid y, e)P(y \mid e)P(x_i' \mid y, e) \]
\[ = P(x_i \mid y, e)P(x_i', y \mid e) \]
\[ = \pi(x')q(x', x) \]

Detailed balance equation is satisfied with \( P(x \mid e) \) as the stationary distribution.

Inference by MCMC

- Instead of generating events from scratch, MCMC generates events by making a random change to the preceding event.
- Start the network at a random state (assignment of values to all its nodes).
- Generate the next state by randomly sampling a value for one of the non-evidence variables, \( X \) conditioned on the current values of the variables in the Markov blanket of \( X \).
- After a burn-in period, each visited state is a sample from the desired distribution.
Example

Query: P(R|SW)

Sample from P(C|RS)

not C RSW

Sample from P(R|not C SW)

not C RSW

Sample from P(C|RS)

CRSW

MCMC example

- Query: P(Rain|Sprinkler, WetGrass)
- Start at:
  - (Cloudy, not Rain, Sprinkler, WetGrass)
- Sample from P(Cloudy|Sprinkler, not Rain), suppose Cloudy = false.
  - (not Cloudy, not Rain, Sprinkler, WetGrass)
- Sample from P(Rain|not Cloudy, Sprinkler, WetGrass), suppose we get Rain = true.
  - (not Cloudy, Rain, Sprinkler, WetGrass)

Repeat these two steps.
Sampling step

- How do we sample Cloudy according to \( P(\text{Cloudy}|\text{Sprinkler, not Rain}) \)?
- Use the network!

\[
P(C|S, \overline{R}) = \frac{P(C, S, \overline{R})}{P(C, S, \overline{R}) + P(\overline{C}, S, \overline{R})} = \frac{P(C)P(S|C)P(\overline{R}|C)}{P(C)P(S|C)P(\overline{R}|C) + P(\overline{C})P(S|\overline{C})P(\overline{R}|\overline{C})}
\]

MCMC

Query: \( P(R|S,W) \)

Construct the transition kernel explicitly and verify that its fix point is in fact: \( P(R|S,W) \)
Why does MCMC work?

- The sampling process settles into a dynamic equilibrium in which the long run fraction of time spent in each state is exactly proportional to the probability $P(X|e)$.

Gibbs sampling in Bayesian networks

- Query: $P(X|e)$; $Z$ is the set of all variables in network.
- Start with a random value $z$ for $Z$.
- Pick a variable $X_i$ in $Z-e$; generate new value $v$ according to $P(X_i|MB(x_i))$
- Move to new state which is $z$ with value of $X_i$ replaced by $v$. 
Knowledge engineering for building Bayesian networks

- Decide what to talk about
  - Identify relevant factors
  - Identify relevant relationships between them.
- Decide on a vocabulary of random variables
  - What are the variables that represent the factors?
  - What are the values they take on?
  - Should a variable be treated as continuous or should it be discretized?

Knowledge engineering (contd.)

- Encode knowledge about dependence between variables
  - Qualitative part: identify topology of belief network.
  - Quantitative part: specify the CPTs
- Encode description of specific problem instance
  - Translate problem givens into values of nodes in the belief network
- Pose queries to inference procedure and get answers
  - Formulate quantity of interest as a conditional probability.
PathFinder System

  - Diagnostic system for lymph node diseases.
  - 60 diseases and 100 symptoms and test results.
  - 14000 probabilities in Bayesian network.
- Effort to build system:
  - 8 hours to determine nodes.
  - 35 hours to determine topology
  - 45 hours to determine probability values
- Performs better than expert doctors.