CSE 473: Artificial Intelligence

Bayes’ Nets: Inference

Steve Tanimoto

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]

Inference

- Inference: calculating some useful quantity from a joint probability distribution

Examples:
- Posterior probability
  \[ P(Q|E_1 = e_1, \ldots, E_k = e_k) \]
- Most likely explanation:
  \[ \arg\max_q P(Q = q|E_1 = e_1, \ldots) \]

Test for Infant Metabolic Defects

Blue ovals represent chromatographic peaks, grey ovals represent 20 metabolic diseases

Inference by Enumeration

- General case:
  - Evidence variables: \( E_1, \ldots, E_k = e_1, \ldots, e_k \)
  - Query* variable: \( Q \)
  - Hidden variables: \( H_1, \ldots, H_r \)

We want:

- Step 1: Select the entries consistent with the evidence
- Step 2: Sum out \( H \) to get joint of \( Q \) and evidence
- Step 3: Normalize

\[
P(Q, e_1, \ldots, e_k) = \sum_{h_1, \ldots, h_r} P(Q, h_1, \ldots, h_r, e_1, \ldots, e_k) \]

\[
Z = \sum_{v} P(Q, e_1, \ldots, e_k) = \frac{1}{Z} P(Q, e_1, \ldots, e_k)
\]

Inference by Enumeration vs. Variable Elimination

- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables
- Idea: interleave joining and marginalizing!
  - Called “Variable Elimination”
  - Still NP-hard, but usually much faster than inference by enumeration

Inference by Enumeration in Bayes’ Net

- Given unlimited time, inference in BNs is easy
- Reminder of inference by enumeration by example:

\[
P(B \mid j, +m) \propto P(B, j, +m) = \sum_{a, e, j} P(B, e, a, j, +m) = \sum_{e, a} P(B) P(a|B, e) P(j|a) P(+m|a)
\]

\[
P(B)P(-a)P(-u|B, +e)P(+j)P(+m|a) + P(B)P(+a)P(-u|B, +e)P(+j)P(+m|a) - aP(B)P(-a)P(-u|B, +e)P(+j)P(+m|a) - aP(B)P(+a)P(-u|B, +e)P(+j)P(+m|a) - a
\]

\[
P(B)P(-e)P(-a|B, -e)P(+j)P(+m|a) + P(B)P(+e)P(-a|B, -e)P(+j)P(+m|a) - aP(B)P(-e)P(-a|B, -e)P(+j)P(+m|a) - aP(B)P(+e)P(-a|B, -e)P(+j)P(+m|a) - a
\]

First we’ll need some new notation: factors
Traffic Domain

\[ P(L) = ? \]

- Inference by Enumeration
  \[ \sum \sum P(L) P(w) P(t) \]

- Variable Elimination
  \[ \sum P(w) \sum P(t) P(v) \]

Factor Zoo

Factor Zoo I

- Joint distribution: \( P(X, Y) \)
  - Entries \( P(x, y) \) for all \( x, y \)
  - Sums to 1

- Selected joint: \( P(x, Y) \)
  - A slice of the joint distribution
  - Entries \( P(x, y) \) for fixed \( x \), all \( y \)
  - Sums to \( P(x) \)

- Number of capitals = dimensionality of the table

Factor Zoo II

- Single conditional: \( P(Y | X) \)
  - Entries \( P(y | x) \) for fixed \( x \), all \( y \)
  - Sums to 1

- Family of conditionals: \( P(Y | X) \)
  - Multiple conditionals
  - Entries \( P(y | x) \) for all \( x, y \)
  - Sums to \( |Y| \)

Factor Zoo III

- Specified family: \( P(Y | X) \)
  - Entries \( P(y | x) \) for fixed \( y \), but for all \( x \)
  - Sums to ... who knows!

Factor Zoo IV

- Two dimensions

Factor Zoo V

- One dimension
Factor Zoo Summary

- In general, when we write \( P(Y_1 \ldots Y_N | X_1 \ldots X_M) \)
  - It is a "factor," a multi-dimensional array
  - Its values are \( P(y_1 \ldots y_N | x_1 \ldots x_M) \)
  - Any assigned (lower-case) \( X \) or \( Y \) is a dimension missing (selected) from the array

Example: Traffic Domain

- Random Variables
  - \( R \): Raining
  - \( T \): Traffic
  - \( L \): Late for class!

\[
P(L) = \sum_{r,t} P(r,t,L)
\]

\[
P(L) = \sum_{r,t} P(r|t) P(t|L)
\]

Inference by Enumeration: Procedural Outline

- Track objects called factors
- Initial factors are local CPTs (one per node)
- Any known values are selected
- Example: Join on \( R \)
- Procedure: Join all factors, then eliminate all hidden variables

Operation 1: Join Factors

- First basic operation: joining factors
- Combining factors:
  - Just like a database join
  - Get all factors over the joining variable
  - Build a new factor over the union of the variables involved
- Example: Join on \( R \)

\[
P(R|R) \times P(T|R) \rightarrow P(R,T)
\]

Example: Multiple Joins

- Example: Multiple Joins
Operation 2: Eliminate

- Second basic operation: marginalization
- Take a factor and sum out a variable
  - Shrinks a factor to a smaller one
  - An aggregate/project operation
- Example:

  \[
P(R, T)
  \]

  \[
  \begin{array}{c|c}
  +r & 0.38 \\
  -r & 0.62 \\
  +t & 0.50 \\
  -t & 0.50 \\
  \end{array}
  \]

  \[
  P(T)
  \]

  \[
  \begin{array}{c|c}
  +t & 0.83 \\
  -t & 0.17 \\
  \end{array}
  \]

Multiple Elimination

\[
P(R, T, L)
\]

\[
\begin{array}{c|c|c|c|c}
  R & T & L & P(R, T, L) \\
  \hline
  + & + & + & 0.024 \\
  + & + & - & 0.001 \\
  + & - & + & 0.003 \\
  + & - & - & 0.001 \\
  - & + & + & 0.018 \\
  - & + & - & 0.007 \\
  - & - & + & 0.003 \\
  - & - & - & 0.003 \\
  \end{array}
\]

\[
P(T, L)
\]

\[
\begin{array}{c|c|c|c}
  T & L & P(T, L) \\
  \hline
  + & + & 0.109 \\
  + & - & 0.884 \\
  - & + & 0.212 \\
  - & - & 0.717 \\
  \end{array}
\]

\[
P(L)
\]

\[
\begin{array}{c|c|c}
  L & P(L) \\
  \hline
  + & 0.32 \\
  - & 0.88 \\
  \end{array}
\]

Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)

Traffic Domain

\[
P(L) = ?
\]

- Inference by Enumeration
  \[
  \sum_{r} \sum_{t} P(R) P(T) = \sum_{r} \sum_{t} P(L) P(R) P(T)
  \]

- Variable Elimination
  \[
  \sum_{r} \sum_{t} P(L) P(T) = \sum_{r} \sum_{t} P(R) P(T) P(L)
  \]

Marginalizing Early (= Variable Elimination)

\[
P(R, T, L)
\]

\[
P(T)
\]

\[
P(L)
\]

Marginalizing Early! (aka VE)

\[
P(R)
\]

\[
P(T)
\]

\[
P(L)
\]
**Evidence**

- If evidence, start with factors that select that evidence
- If there is no evidence, then use these initial factors:

\[
P(R) \quad P(T) \quad P(L) \quad P(R \mid T) \quad P(T \mid R) \quad P(L \mid T)
\]

- But if given some evidence, eg +r, then select for it...
- Computing \( P(L \mid +r) \) the initial factors become:

\[
P(+r) \quad P(T \mid +r) \quad P(L \mid T)
\]

- Next do joins & eliminate, removing all vars other than query + evidence

**Evidence II**

- Result will be a selected joint of query and evidence
- E.g. for \( P(L \mid +r) \) we would end up with:

\[
P(+r, L) \quad \text{Normalize} \quad P(L \mid +r)
\]

- To get our answer, just normalize this!

- That’s it!

**General Variable Elimination**

- Query: \( P(Q \mid E_1 = e_1 \ldots E_k = e_k) \)
- Start with initial factors:
  - Local CPTs (but instantiated by evidence)
- While there are still hidden variables (not Q or evidence):
  - Choose a hidden variable \( H \)
  - Join all factors mentioning \( H \)
  - Eliminate (sum out) \( H \)
- Join all remaining factors and normalize

\[
\times \frac{1}{Z}
\]

**Example: Alarm Network**

- \( P(B \mid j, m) = ? \)

\[
\begin{array}{c|c|c}
B & P(B) & +b \quad 0.001 \\
 & -b \quad 0.999 \\
E & P(E) & +e \quad 0.002 \\
 & -e \quad 0.998 \\
A & P(A \mid B, E) & +a \quad 0.05 \\
 & -a \quad 0.95 \\
M & P(M \mid A) & +m \quad 0.01 \\
 & -m \quad 0.99 \\
J & P(J \mid A) & +j \quad 0.9 \\
 & -j \quad 0.1 \\
\end{array}
\]

**Example**

\[
P(j, m) \propto P(j, m, B, E)
\]

Choose A

\[
P(A \mid B, E) \quad P(j, m, A \mid B, E) \quad P(j, m \mid B, E)
\]

Finish with B

\[
P(B) \quad P(j, m \mid B)
\]

Choose E

\[
P(E) \quad P(E \mid j, m, B, E) \quad P(j, m \mid B)
\]

Finish with B

\[
P(B) \quad P(j, m \mid B)
\]

Normalize \( P(B \mid j, m) \)
Same Example in Equations

\[
P(B|j, m) \propto P(B, j, m)
\]

\[
P(B) = \sum_{j} P(B, j, m)
\]

\[
P(E) = \sum_{j} P(E, j, m)
\]

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

\[
P(j|A) = \sum_{m} P(j|A, m)
\]

\[
P(m|A) = \sum_{j} P(m|A, j)
\]

marginal can be obtained from joint by summing out

use Bayes’ net joint distribution expression

\[
yz = xy + xz
\]

do sum first

joining on \( a \), and then summing out gives \( f_1 \)

\[
yz = yz + xz
\]

do sum first

joining on \( e \), and then summing out gives \( f_2 \)

Simple! Exploiting \( uwy + uwz + uxy + uxz + vwy + vwz + vxy + vxz = (u+v)(w+x)(y+z) \) to reduce computation

Choices during Variable Elimination

- Query: \( P(Q|E_1 = e_1 \ldots E_k = e_k) \)
- Start with initial factors:
  - Local CPTs (but instantiated by evidence)
- While there are still hidden variables (not \( Q \) or evidence):
  - Choose a hidden variable \( H \)
  - Join all factors mentioning \( H \)
  - Eliminate (sum out) \( H \)
  - Join all remaining factors and normalize

Another Variable Elimination Example

Query: \( P(X_1|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3) \)

Alternatively, suppose we start by eliminating \( Z \):

\( f_Z(X_1, X_2, X_3) \)

What is the resulting factor?

What dimension is it?

How many entries?

Another Variable Elimination Example

Query: \( P(X_1|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3) \)

Start by eliminating evidence, which gives the following initial factors:

\( p(y_1|X_1) \)

\( p(y_2|X_2) \)

\( p(y_3|X_3) \)

What variables could we eliminate?

What dimension are \( f_1 \), \( f_2 \) & \( f_3 \)?

1
variable Elimination Example

Query: \( P(X_2|Y_1 = y_1, Y_2 = y_2, Y_1 = y_1) \)

Start by inserting evidence, which gives the following initial factors:

\( P(Y_1 = y_1, Y_2 = y_2|X_2) \) (Evidence: \( Y_1 \)), this introduces the factor \( P(Y_1 = y_1|X_2) \), and we are left with:

\( P(Y_2 = y_2|X_2) \) (Evidence: \( Y_2 \)), this introduces the factor \( P(Y_2 = y_2|X_2) \), and we are left with:

Finally, \( Z \) this introduces the factor \( P(Z|X_2) = \sum_{X_1} P(Z,X_1,X_2) \), and we are left:

No hidden variables left. Start the multiplying factors to get:

\( P(Z_1, Z_2|X_2) = P(Z_1|X_2) P(Z_2|X_2) \).

Combining over \( X_2 \) gives \( P(Z_1, Z_2) \).

Variable Elimination Ordering

- For the query \( P(X_1, X_2, \ldots, X_n) \) work through the following two different orderings as done in previous slide: \( X, X_1, \ldots, X_{n-1} \) and \( X_1, \ldots, X_{n-1}, Z \). What is the size of the maximum factor generated for each of the orderings?

- Answer: \( 2^{n+1} \) versus \( 2^n \) (assuming binary).

- In general, the ordering can greatly affect efficiency.

VE: Computational and Space Complexity

- The computational and space complexity of variable elimination is determined by the largest factor.

- The elimination ordering can greatly affect the size of the largest factor.
  - E.g., previous slide’s example \( 2^n \) vs. \( 2^2 \)

- Does there always exist an ordering that only results in small factors?
  - No!

Worst Case Complexity?

- CSP:

  \[
  \begin{align*}
  &P(X_1 = 0) = P(X_1 = 1) = 0.5 \\
  &Y_1 = X_1 \lor X_2 \lor \overline{X}_3 \\
  &Y_2 = \overline{X}_1 \lor X_2 \lor X_3 \\
  &Y_{12} = Y_1 \land Y_2 \\
  &Y_{123} = Y_1 \land Y_2 \land X_3 \\
  &Z = Y_{123} \lor Y_{12} \\
  \end{align*}
  \]

  \( P(X_1 = 0) \) if we can answer \( P(Y) \) equal to zero or not, we answered whether the 3-SAT problem has a solution.

  Hence inference in Bayes’ nets is NP-hard. No known efficient probabilistic inference in general.

Bayes’ Nets

- Representation
- Conditional Independences
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Inference is NP-complete
    - Sampling (approximate)
- Learning Bayes’ Nets from Data