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:
    \[ \arg \max_Q P(Q = q|E_1 = e_1, \ldots) \]

Inference by Enumeration

- Given unlimited time, inference in BNs is easy
- Recipe:
  - State the marginal probabilities you need
  - Figure out ALL the atomic probabilities you need
  - Calculate and combine them
- Example:
  \[
  P(+b|+j, +m) = \frac{P(+b, +j, +m)}{P(+j, +m)}
  \]

Example: Enumeration

- In this simple method, we only need the BN to synthesize the joint entries

\[
P(+b, +j, +m) = \\
P(+b)P(+e)P(+a|+b, +e)P(+j|+a)P(+m|+a) + \\
P(+b)P(+e)P(-a|+b, +e)P(+j|-a)P(+m|-a) + \\
P(+b)P(-e)P(+a|+b, -e)P(+j|+a)P(+m|+a) + \\
P(+b)P(-e)P(-a|+b, -e)P(+j|-a)P(+m|-a)
\]
Variable Elimination

- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables
  - You end up repeating a lot of work!
- Idea: interleave joining and marginalizing!
  - Called “Variable Elimination”
  - Still NP-hard, but usually much faster than inference by enumeration
- We’ll need some new notation to define VE

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)

Factor Zoo II

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

Factor Zoo III

- Specified family: P(y | X)
  - Entries P(y | x) for fixed y, but for all x
  - Sums to … who knows!
- In general, when we write P(Y₁, ..., Yₙ | X₁, ..., Xₘ)
  - It is a “factor,” a multi-dimensional array
  - Its values are all P(y₁, ..., yₙ | X₁, ..., Xₘ)
  - Any assigned X or Y is a dimension missing (selected) from the array

Example: Traffic Domain

- Random Variables
  - R: Raining
  - T: Traffic
  - L: Late for class!
- First query: P(L)

Variable Elimination Outline

- Track objects called factors
- Initial factors are local CPTs (one per node)
- Any known values are selected
  - E.g. if we know L = +l, the initial factors are
- VE: Alternately join factors and eliminate variables
Operation 1: Join Factors

- First basic operation: joining factors
- Combining factors:
  - Just like a database join
  - Get all factors over the union of the variables involved
- Example: Join on R

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

\[
\begin{array}{ccc}
\text{r} & \text{t} & \text{p} \\
0.1 & 0.8 & 0.08 \\
0.9 & 0.2 & 0.18 \\
\end{array}
\]

\[
\begin{array}{ccc}
\text{r} & \text{t} & \text{p} \\
0.1 & 0.2 & 0.02 \\
1 & 0.1 & 0.09 \\
1 & 0.9 & 0.81 \\
\end{array}
\]

- Computation for each entry: pointwise products
  \[\forall r, t : P(r, t) = P(r) \cdot P(t|r)\]

Example: Multiple Joins

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

\[
P(T|R)
\]

\[
P(L|T)
\]

Operation 2: Eliminate

- Second basic operation: marginalization
- Take a factor and sum out a variable
  - Shrinks a factor to a smaller one
  - A projection operation
- Example:

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

Multiple Elimination

P(L) : Marginalizing Early!
Marginalizing Early (aka VE*)

### Evidence
- If evidence, start with factors that select that evidence
  - No evidence uses these initial factors:
    - Computing $P(L|+r)$, the initial factors become:

### General 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):
  - Pick a hidden variable $H$
  - Join all factors mentioning $H$
  - Eliminate (sum out) $H$
- Join all remaining factors and normalize

### Example

| $P(B)$ | $P(E)$ | $P(A|B,E)$ | $P(j|A)$ | $P(m|A)$ |
|--------|--------|------------|----------|----------|
| $P(B)$ | $P(E)$ | $P(A|B,E)$ | $P(j|A)$ | $P(m|A)$ |

Choose $A$

$$P(A|B,E)$$

### Variable Elimination Bayes Rule

| $P(B)$ | $P(E)$ | $P(A|B,E)$ | $P(j|A)$ | $P(m|A)$ |
|--------|--------|------------|----------|----------|
| $P(B)$ | $P(E)$ | $P(A|B,E)$ | $P(j|A)$ | $P(m|A)$ |

Choose $A$

$$P(A|B,E)$$

### Evidence II
- Result will be a selected joint of query and evidence
  - E.g. for $P(L|+r)$, we’d end up with:

### Evidence
- If evidence, start with factors that select that evidence
  - No evidence uses these initial factors:
    - Computing $P(L|+r)$, the initial factors become:

### General 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):
  - Pick a hidden variable $H$
  - Join all factors mentioning $H$
  - Eliminate (sum out) $H$
- Join all remaining factors and normalize

### Example

| $P(B)$ | $P(E)$ | $P(A|B,E)$ | $P(j|A)$ | $P(m|A)$ |
|--------|--------|------------|----------|----------|
| $P(B)$ | $P(E)$ | $P(A|B,E)$ | $P(j|A)$ | $P(m|A)$ |

Choose $A$

$$P(A|B,E)$$

### Variable Elimination Bayes Rule

| $P(B)$ | $P(E)$ | $P(A|B,E)$ | $P(j|A)$ | $P(m|A)$ |
|--------|--------|------------|----------|----------|
| $P(B)$ | $P(E)$ | $P(A|B,E)$ | $P(j|A)$ | $P(m|A)$ |

Choose $A$

$$P(A|B,E)$$
Example

\[
\begin{array}{c|c|c}
P(B) & P(E) & P(j,m|B,E) \\
\end{array}
\]

Choose E

\[
\begin{array}{c|c|c}
P(E) & P(j,m,E|B) & P(j,m|B) \\
\end{array}
\]

\[
\begin{array}{c|c|c}
\times & P(B) & P(j,m|B) \\
\end{array}
\]

Finish with B

\[
\begin{array}{c|c|c}
P(B) & P(j,m,B) & \text{Normalize} \\
\times & P(B|j,m) \\
\end{array}
\]

Variable Elimination

- **What you need to know:**
  - Should be able to run it on small examples, understand the factor creation / reduction flow
  - Better than enumeration: saves time by marginalizing variables as soon as possible rather than at the end
- **We will see special cases of VE later**
  - On tree-structured graphs, variable elimination runs in polynomial time, like tree-structured CSPs
  - You’ll have to implement a tree-structured special case to track invisible ghosts (Project 4)