Announcements

- Assignments
  - W4 back today in lecture
  - Any assignments you have not picked up yet
    - In bin in 283 Soda [same room as for submission drop-off]
- Midterm
  - 3/18, 6-9pm, 0010 Evans — no lecture on Thursday
  - We have posted practice midterms (and finals)
  - One note letter-size note sheet (two sides), non-programmable calculators [strongly encouraged to compose your own!]
  - Topics go through last Thursday
- Section this week: midterm review

Bayes’ Net Semantics

- Let’s formalize the semantics of a Bayes’ net
  - A set of nodes, one per variable \(X\)
  - A directed, acyclic graph
  - A conditional distribution for each node
    - A collection of distributions over \(X\), one for each combination of parents’ values
      \[ P(X|a_1 \ldots a_n) \]
    - CPT: conditional probability table
    - Description of a noisy “causal” process

\[ A \text{ Bayes net} = \text{Topology (graph)} + \text{Local Conditional Probabilities} \]

Probabilities in BNs

- For all joint distributions, we have (chain rule):
  \[ P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | x_1, \ldots, x_{i-1}) \]
- Bayes’ nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:
    \[ P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | \text{parents}(x_i)) \]
- This lets us reconstruct any entry of the full joint
- Not every BN can represent every joint distribution
- The topology enforces certain conditional independencies

Inference

- Inference: calculating some useful quantity from a joint probability distribution
- Examples:
  - Posterior probability:
    \[ P(Q|E_1, \ldots, E_k) = \frac{1}{c_k} \]
  - Most likely explanation:
    \[ \arg \max_q P(Q = q | E_1 = c_1 \ldots) \]
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

\[
P( +b, +j, +m ) =
\]

\[
P(+b)P(+c)P(+a)P(+b, +c)P(+j)P(+a)P(+m) +
P(+b)P(+c)P(+a)P(+b, +c)P(+j)P(+a)P(+m) -
P(+b)P(-c)P(+a)P(+b, -c)P(+j)P(+a)P(+m) +
P(+b)P(-c)P(-a)P(+b, -c)P(+j)P(+a)P(+m) -
P(+b)P(-c)P(-a)P(+b, -c)P(+j)P(+a)P(+m)
\]

Inference by Enumeration?

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 \( P(x) \)

\[
\begin{array}{ccc}
T & W & P \\
\hline
\text{hot} & \text{sun} & 0.4 \\
\text{hot} & \text{rain} & 0.1 \\
\text{cold} & \text{sun} & 0.2 \\
\text{cold} & \text{rain} & 0.3 \\
\end{array}
\]

\[
\begin{array}{ccc}
T & W & P \\
\hline
\text{cold} & \text{sun} & 0.2 \\
\text{cold} & \text{rain} & 0.3 \\
\end{array}
\]

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

\[
\begin{array}{ccc}
T & W & P \\
\hline
\text{hot} & \text{sun} & 0.8 \\
\text{hot} & \text{rain} & 0.2 \\
\text{cold} & \text{sun} & 0.4 \\
\text{cold} & \text{rain} & 0.6 \\
\end{array}
\]

\[
\begin{array}{ccc}
T & W & P \\
\hline
\text{cold} & \text{sun} & 0.4 \\
\text{cold} & \text{rain} & 0.6 \\
\end{array}
\]
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 $P(Y_1 ... Y_N | X_1 ... X_M)$
  - It is a "factor," a multi-dimensional array
  - Its values are all $P(y_1 ... y_N | x_1 ... x_M)$
  - 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!

- $P(R)$

- $P(T)$

- $P(L)$

- $P(R|T)$

- $P(T|R)$

- $P(L|R)$

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 joining variable
  - Build a new factor over the union of the variables involved

- Example: Join on $R$

Example: Multiple Joins

- $P(R)$

- $P(T|R)$

- $P(L|R)$

- $P(L|T)$

Example: Multiple Joins

- $P(R)$

- $P(T)$

- $P(L)$

- $P(R, T)$

- $P(R, T, L)$
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:

\[
P(R, T) \rightarrow \text{sum } R \rightarrow P(T)
\]

\[
\begin{array}{c|c|c}
  & R^+ & R^- \\
  T^+ & 0.08 & 0.02 \\
  T^- & 0.09 & 0.81 \\
\end{array}
\]

\[
\begin{array}{c|c|c}
  & R^+ & R^- \\
  T^+ & 0.17 & 0.83 \\
  T^- & & \\
\end{array}
\]

Multiple Elimination

- Marginalization
  - Take a factor and sum out a variable
    - Shrinks a factor to a smaller one
- Example:

\[
P(R, T, L) \rightarrow \text{sum out } R \rightarrow P(T, L) \rightarrow \text{sum out } T \rightarrow P(L)
\]

\[
\begin{array}{c|c|c|c}
  & R^+ & R^- & L^+ \\
  T^+ & 0.09 & 0.01 & 0.119 \\
  T^- & 0.07 & 0.03 & 0.043 \\
  L^+ & 0.081 & 0.018 & 0.729 \\
\end{array}
\]

P(L) : Marginalizing Early!

\[
P(R) \rightarrow \text{Join } R \rightarrow P(R, T) \rightarrow \text{Sum out } R \rightarrow P(T) \rightarrow \text{Sum out } T \rightarrow P(L)
\]

\[
\begin{array}{c|c|c|c}
  & R^+ & R^- \\
  L^+ & 0.17 & 0.83 \\
  L^- & 0.051 & 0.049 \\
\end{array}
\]

Marginalizing Early (aka VE*)

- VE is variable elimination

Evidence

- If evidence, start with factors that select that evidence
  - No evidence uses these initial factors:

\[
P(R) \rightarrow P(T|R) \rightarrow P(L|R)
\]

\[
\begin{array}{c|c|c|c}
  & R^+ & R^- \\
  T^+ & 0.26 & 0.74 \\
  T^- & 0.02 & 0.98 \\
  L^+ & 0.026 & 0.074 \\
\end{array}
\]

Evidence II

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

\[
P(+r, L) \rightarrow \text{Normalize } P(L | +r)
\]

\[
\begin{array}{c|c|c|c}
  & R^+ & R^- \\
  T^+ & 0.26 & 0.74 \\
\end{array}
\]

- To get our answer, just normalize this!
- That’s it!
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

Variable Elimination Bayes Rule

```
Start / Select          Join on B          Normalize
P(B)              B              a, B
                  a
P(A|B)→P(a|B)

P(B|a, B)          a
P(B|a)

P(B|a)
```

Example

```
P(B|m, n) = P(B, m, n)
P(B) P(E) P(A|B, E) P(j|A) P(m|A)
```

Choose A

```
P(A|B, E) P(j|A) P(m|A)
P(B) P(E) P(j, m|B, E)
```

Example

```
P(B) P(E) P(j, m|B, E)
```

Choose E

```
P(E) P(j, m, E|B) P(j, m|B)
P(B) P(j, m|B)
```

Finish with B

```
P(B) P(j, m|B) Normalize P(B|j, m)
```

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)