CSE 573: Artificial Intelligence

Bayes’ Net Teaser

Gagan Bansal
(slides by Dan Weld)

[Most 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.]
Probability Recap

- **Conditional probability**
  \[ P(x|y) = \frac{P(x, y)}{P(y)} \]

- **Product rule**
  \[ P(x, y) = P(x|y)P(y) \]

- **Chain rule**
  \[ P(X_1, X_2, \ldots X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots = \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1}) \]

- **Bayes rule**
  \[ P(x|y) = \frac{P(y|x)}{P(y)}P(x) \]

- **X, Y independent if and only if:**
  \[ \forall x, y : P(x, y) = P(x)P(y) \]

- **X and Y are conditionally independent given Z:**
  \[ X \perp \!
\perp Y | Z \]
  \[ \forall x, y, z : P(x, y|z) = P(x|z)P(y|z) \]
Probabilistic Inference

- Probabilistic inference =
  "compute a desired probability from other known probabilities (e.g. conditional from joint)"

- We generally compute conditional probabilities
  - $P(\text{on time} \mid \text{no reported accidents}) = 0.90$
  - These represent the agent’s beliefs given the evidence

- Probabilities change with new evidence:
  - $P(\text{on time} \mid \text{no accidents, 5 a.m.}) = 0.95$
  - $P(\text{on time} \mid \text{no accidents, 5 a.m., raining}) = 0.80$
  - Observing new evidence causes beliefs to be updated
Inference by Enumeration

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

\[ \left\{ \begin{array}{l} X_1, X_2, \ldots X_n \\ \text{All variables} \end{array} \right\} \]

**We want:**
\[
P(Q|e_1 \ldots e_k)
\]

* Works fine with multiple query variables, too

**Step 1:** Select the entries consistent with the evidence

**Step 2:** Sum out \( H \) to get joint of Query and evidence

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

**Step 3:** Normalize

\[
Z = \sum_q P(Q, e_1 \ldots e_k)
\]

\[
P(Q|e_1 \ldots e_k) = \frac{1}{Z} P(Q, e_1 \ldots e_k)
\]
Inference by Enumeration

- Computational problems?
  - Worst-case time complexity $O(d^n)$
  - Space complexity $O(d^n)$ to store the joint distribution
The Sword of Conditional Independence!

Slay the Basilisk!

\[ X \independent Y \mid Z \]

Meaning:
\[ \forall x, y, z : P(x, y \mid z) = P(x \mid z)P(y \mid z) \]

Or, equivalently:
\[ \forall x, y, z : P(x \mid z, y) = P(x \mid z) \]
Bayes’Nets: Big Picture
Bayes’ Nets

- Representation & Semantics
- Conditional Independences
- Probabilistic Inference
- Learning Bayes’ Nets from Data
Bayes Nets = a Kind of Probabilistic Graphical Model

- Models describe how (a portion of) the world works

- Models are always simplifications
  - May not account for every variable
  - May not account for all interactions between variables
  - “All models are wrong; but some are useful.”
    – George E. P. Box

- What do we do with probabilistic models?
  - We (or our agents) need to reason about unknown variables, given evidence
  - Example: explanation (diagnostic reasoning)
  - Example: prediction (causal reasoning)
  - Example: value of information

Friction,
Air friction,
Mass of pulley,
Inelastic string, …
Bayes’ Nets: Big Picture

- Two problems with using full joint distribution tables as our probabilistic models:
  - Unless there are only a few variables, the joint is WAY too big to represent explicitly
  - Hard to learn (estimate) anything empirically about more than a few variables at a time

- Bayes’ nets: a technique for describing complex joint distributions (models) using simple, local distributions (conditional probabilities)
  - More properly ... aka probabilistic graphical model
  - We describe how variables locally interact
  - Local interactions chain together to give global, indirect interactions
  - For about 10 min, we’ll be vague about how these interactions are specified
Example Bayes’ Net: Insurance
Bayes’ Net Semantics
Bayes’ Net Semantics

- 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

\[ \begin{align*}
P(A_1) & \quad \ldots \quad P(A_n) \\
A_1 & \quad \cdots \quad A_n \\
X & \quad P(X|A_1 \ldots A_n)
\end{align*} \]

*A Bayes net = Topology (graph) + Local Conditional Probabilities*
Example: Alarm Network

<table>
<thead>
<tr>
<th>B</th>
<th>P(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+b</td>
<td>0.001</td>
</tr>
<tr>
<td>-b</td>
<td>0.999</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>P(E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+e</td>
<td>0.002</td>
</tr>
<tr>
<td>-e</td>
<td>0.998</td>
</tr>
</tbody>
</table>

| B  | E  | A   | P(A|B,E) |
|----|----|-----|---------|
| +b | +e | +a  | 0.95    |
| +b | +e | -a  | 0.05    |
| +b | -e | +a  | 0.94    |
| -b | +e | +a  | 0.29    |
| -b | +e | -a  | 0.71    |
| -b | -e | +a  | 0.001   |
| -b | -e | -a  | 0.999   |

| A  | J   | P(J|A) |
|----|-----|-------|
| +a | +j  | 0.9   |
| +a | -j  | 0.1   |
| -a | +j  | 0.05  |
| -a | -j  | 0.95  |

| A  | M   | P(M|A) |
|----|-----|-------|
| +a | +m  | 0.7   |
| +a | -m  | 0.3   |
| -a | +m  | 0.01  |
| -a | -m  | 0.99  |
Bayes Nets Implicitly Encode Joint Distribution

\[
P(+b, -e, +a, -j, +m) =
\]
Bayes Nets Implicitly Encode Joint Distribution

\[
P(+b, -e, +a, -j, +m) = P(+b)P(-e)P(+a|+b, -e)P(-j|+a)P(+m|+a) = 0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7
\]
Joint Probabilities from BNs

- Why are we guaranteed that setting results in a proper joint distribution?

Chain rule (valid for all distributions):

\[ P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i | \text{parents}(X_i)) \]

Assume conditional independences:

\[ P(x_i | x_1, \ldots x_{i-1}) = P(x_i | \text{parents}(X_i)) \]

→ Consequence:

\[ P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i | \text{parents}(X_i)) \]

- Every BN represents a joint distribution, but
- Not every distribution can be represented by a specific BN
  - The topology enforces certain conditional independencies
Causality?

- When Bayes’ nets reflect the true causal patterns:
  - Often simpler (nodes have fewer parents)
  - Often easier to think about
  - Often easier to elicit from experts

- BNs need not actually be causal
  - Sometimes no causal net exists over the domain (especially if variables are missing)
  - E.g. consider the variables Traffic and Drips
  - End up with arrows that reflect correlation, not causation

- What do the arrows really mean?
  - Topology may happen to encode causal structure
  - Topology really encodes conditional independence
    \[ P(x_i|x_1, \ldots x_{i-1}) = P(x_i|\text{parents}(X_i)) \]
Size of a Bayes’ Net

- How big is a joint distribution over $N$ Boolean variables?
  $2^N$

- How big is an $N$-node net if nodes have up to $k$ parents?
  $O(N \times 2^k)$

- Both give you the power to calculate
  $P(X_1, X_2, \ldots, X_n)$

- BNs: Huge space savings!

- Also easier to elicit local CPTs

- Also faster to answer queries (coming)
Inference in Bayes’ Net

- Many algorithms for both exact and approximate inference
- Complexity often based on
  - Structure of the network
  - Size of undirected cycles
- Usually faster than exponential in number of nodes

- **Exact inference**
  - Variable elimination
  - Junction trees and belief propagation

- **Approximate inference**
  - Loopy belief propagation
  - Sampling based methods: likelihood weighting, Markov chain Monte Carlo
  - Variational approximation
Summary: Bayes’ Net Semantics

- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over $X$, one for each combination of parents’ values
    \[ P(X|a_1 \ldots a_n) \]
- Bayes’ nets \textit{compactly} 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:
    \[ P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i|\text{parents}(X_i)) \]
Hidden Markov Models

- Defines a joint probability distribution:

\[
P(X_1, \ldots, X_n, E_1, \ldots, E_n) = \\
P(X_{1:n}, E_{1:n}) = \\
P(X_1)P(E_1|X_1) \prod_{t=2}^{N} P(X_t|X_{t-1})P(E_t|X_t)
\]
An HMM is defined by:

- Initial distribution: \( P(X_1) \)
- Transitions: \( P(X_t | X_{t-1}) \)
- Emissions: \( P(E | X) \)
HMMs have two important independence properties:

- Future independent of past given the present
Conditional Independence

HMMs have two important independence properties:

- Future independent of past given the present
- Current observation independent of all else given current state

![Diagram showing conditional independence properties of HMMs.]
HMMs have two important independence properties:
- Markov hidden process, future depends on past via the present
- Current observation independent of all else given current state

Quiz: does this mean that observations are independent given no evidence?
- [No, correlated by the hidden state]
Inference in Ghostbusters

- A ghost is in the grid somewhere
- Sensor readings tell how close a square is to the ghost
  - On the ghost: red
  - 1 or 2 away: orange
  - 3 or 4 away: yellow
  - 5+ away: green

- Sensors are noisy, but we know $P(\text{Color} \mid \text{Distance})$

<table>
<thead>
<tr>
<th></th>
<th>$P(\text{red} \mid 3)$</th>
<th>$P(\text{orange} \mid 3)$</th>
<th>$P(\text{yellow} \mid 3)$</th>
<th>$P(\text{green} \mid 3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.05</td>
<td>0.15</td>
<td>0.5</td>
<td>0.3</td>
</tr>
</tbody>
</table>
Ghostbusters HMM

- $P(X_1) = \text{uniform}$
- $P(X' | X) = \text{ghosts usually move clockwise, but sometimes move in a random direction or stay put}$
- $P(E | X) = \text{same sensor model as before: red means probably close, green means likely far away.}$

\[
P(X_1) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}
\]

\[
P(X' | X=\{1,2\}) = \begin{bmatrix} 1/6 & 1/6 & 1/2 \\ 0 & 1/6 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\]

\[
P(E | X) = \begin{bmatrix} P(\text{red} | x) & P(\text{orange} | x) & P(\text{yellow} | x) & P(\text{green} | x) \\ 2 & ... & ... & ... \\ 3 & 0.05 & 0.15 & 0.5 & 0.3 \\ 4 & ... & ... & ... & ... \end{bmatrix}
\]
HMM Examples

- Speech recognition HMMs:
  - States are specific positions in specific words (so, tens of thousands)
  - Observations are acoustic signals (continuous valued)
HMM Examples

- **POS tagging HMMs:**
  - State is the parts of speech tag for a specific word
  - Observations are words in a sentence (size of the vocabulary)
HMM Computations

- Given
  - parameters
  - evidence $E_{1:n} = e_{1:n}$

- Inference problems include:
  - **Filtering**, find $P(X_t|e_{1:t})$ for some $t$
  - **Most probable explanation**, for some $t$ find $x^*_{1:t} = \text{argmax}_{x_{1:t}} P(x_{1:t}|e_{1:t})$
  - **Smoothing**, find $P(X_t|e_{1:n})$ for some $t < n$
The task of tracking the agent’s belief state, $B(x)$, over time
- $B(x)$ is a distribution over world states – repr agent knowledge
- We start with $B(X)$ in an initial setting, usually uniform
- As time passes, or we get observations, we update $B(X)$

Many algorithms for this:
- Exact probabilistic inference
- Particle filter approximation
- Kalman filter (a method for handling continuous Real-valued random vars)
  - invented in the 60’s for Apollo Program – real-valued state, Gaussian noise
HMM Examples

- **Robot tracking:**
  - States (X) are positions on a map (continuous)
  - Observations (E) are range readings (continuous)
Filtering (aka Monitoring)

- Filtering, or monitoring, is the task of tracking the distribution $B_t(X)$ (called “the belief state”) over time.

- We start with $B_0(X)$ in an initial setting, usually uniform.

- We update $B_t(X)$
  1. As time passes, and
  2. As we get observations

  computing $B_{t+1}(X)$
  using prob model of how ghosts move
  using prob model of how noisy sensors work
Filtering: Base Cases

"Observation"

\[ P(X_1|e_1) \]

\[
P(x_1|e_1) = P(x_1, e_1)/P(e_1)
\]

\[
\propto_{X_1} P(x_1, e_1)
\]

\[
= P(x_1)P(e_1|x_1)
\]

"Passage of Time"

\[ P(X_2) \]

\[
P(x_2) = \sum_{x_1} P(x_1, x_2)
\]

\[
= \sum_{x_1} P(x_1)P(x_2|x_1)
\]
Forward Algorithm

\[ B(X_t) = P(X_t|e_{1:t}) \]

- \( t = 0 \)
- \( B(X_t) = \) initial distribution
- Repeat forever
  - \( B'(X_{t+1}) = \) Simulate passage of time from \( B(X_t) \)
  - Observe \( e_{t+1} \)
  - \( B(X_{t+1}) = \) Update \( B'(X_{t+1}) \) based on probability of \( e_{t+1} \)
Passage of Time

- Assume we have current belief \( P(X \mid \text{evidence to date}) \)
  
  \[ B(X_t) = P(X_t \mid e_{1:t}) \]

- Then, after one time step passes:
  
  \[
P(X_{t+1} \mid e_{1:t}) = \sum_{x_t} P(X_{t+1}, x_t \mid e_{1:t}) \]
  
  \[
  = \sum_{x_t} P(X_{t+1} \mid x_t, e_{1:t}) P(x_t \mid e_{1:t})
  \]
  
  \[
  = \sum_{x_t} P(X_{t+1} \mid x_t) P(x_t \mid e_{1:t})
  \]

- Basic idea: beliefs get “pushed” through the transitions
  
  - With the “B” notation, we have to be careful about what time step \( t \) the belief is about, and what evidence it includes

\[ \begin{array}{c}
  X_1 \\
  \rightarrow \\
  X_2
\end{array} \]
As time passes, uncertainty “accumulates”

(Transition model: ghosts usually go clockwise)
Observation

- Assume we have current belief $P(X | \text{previous evidence})$:
  \[
  B'(X_{t+1}) = P(X_{t+1}|e_{1:t})
  \]

- Then, after evidence comes in:
  \[
  P(X_{t+1}|e_{1:t+1}) = \frac{P(X_{t+1}, e_{t+1}|e_{1:t})}{P(e_{t+1}|e_{1:t})}
  \]
  \[
  = \frac{P(e_{t+1}|e_{1:t}, X_{t+1})P(X_{t+1}|e_{1:t})}{P(e_{t+1}|e_{1:t})}
  \]
  \[
  = \frac{P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t})}{P(e_{t+1}|e_{1:t})}
  \]

- Or, compactly:
  \[
  B(X_{t+1}) = \frac{P(e_{t+1}|X_{t+1})B'(X_{t+1})}{P(e_{t+1}|e_{1:t})}
  \]

- Basic idea: beliefs “rewighted” by likelihood of evidence
- Unlike passage of time, we have to normalize
Example: Observation

- As we get observations, beliefs get reweighted, uncertainty “decreases”

\[ B(X) \propto P(e|X)B'(X) \]
**Normalization to Account for Evidence**

<table>
<thead>
<tr>
<th>X</th>
<th>E</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>rain</td>
<td>U</td>
<td>0.4</td>
</tr>
<tr>
<td>rain</td>
<td>-</td>
<td>0.1</td>
</tr>
<tr>
<td>sun</td>
<td>U</td>
<td>0.2</td>
</tr>
<tr>
<td>sun</td>
<td>-</td>
<td>0.3</td>
</tr>
</tbody>
</table>

**SELECT** the joint probabilities matching the evidence

<table>
<thead>
<tr>
<th>X</th>
<th>E</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>rain</td>
<td>U</td>
<td>0.4</td>
</tr>
<tr>
<td>sun</td>
<td>U</td>
<td>0.2</td>
</tr>
</tbody>
</table>

**NORMALIZE** the selection (make it sum to one)

<table>
<thead>
<tr>
<th>X</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>rain</td>
<td>0.67</td>
</tr>
<tr>
<td>sun</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Since could have seen other evidence, we normalize by dividing by the probability of the evidence we *did* see (in this case dividing by 0.5)...
Pacman – Sonar (P5)

[Demo: Pacman – Sonar – No Beliefs(L14D1)]
Video of Demo Pacman – Sonar (with beliefs)
Every time step, we start with current $P(X \mid \text{evidence})$

1. We update for time:

$$P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$$

2. We update for evidence:

$$P(x_t|e_{1:t}) \propto_x P(x_{t}|e_{1:t-1}) \cdot P(e_t|x_t)$$

The forward algorithm does both at once (and doesn’t normalize)

Computational complexity?

$O(X^2 +XE)$ time & $O(X+E)$ space