Outline

- Probabilistic sequence models (and inference)
  - (Review) Hidden Markov Models
  - (Review) Particle Filters
  - (Postponed) Most Probable Explanations
  - Dynamic Bayesian networks
  - Bayesian Networks (BNs)
  - Independence in BNs
Announcements

- We are still grading PS3
- PS4 out, due next Monday
- Mini-project guidelines out this week
- Exam next Thursday
  - In class, closed book, one page of notes
- Look at Berkley exams for practice:
  - http://inst.eecs.berkeley.edu/~cs188/fa10/midterm.html
Recap: Reasoning Over Time

- **Stationary Markov models**
  \[ P(X_1) \quad P(X|X_{-1}) \]

- **Hidden Markov models**

<table>
<thead>
<tr>
<th>X</th>
<th>E</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>rain</td>
<td>umbrella</td>
<td>0.9</td>
</tr>
<tr>
<td>rain</td>
<td>no umbrella</td>
<td>0.1</td>
</tr>
<tr>
<td>sun</td>
<td>umbrella</td>
<td>0.2</td>
</tr>
<tr>
<td>sun</td>
<td>no umbrella</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Recap: Hidden Markov Models

- Defines a joint probability distribution:

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

- Filtering is the inference process of finding a distribution over $X_T$ given $e_1$ through $e_T$: $P( X_T | e_{1:t} )$
- We first compute $P( X_1 | e_1 )$: $P(x_1|e_1) \propto P(x_1) \cdot P(e_1|x_1)$
- For each $t$ from 2 to $T$, we have $P( X_{t-1} | e_{1:t-1} )$
  - **Elapse time:** compute $P( X_t | e_{1:t-1} )$
    \[
P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})
    \]
  - **Observe:** compute $P(X_t | e_{1:t-1}, e_t) = P( X_t | e_{1:t} )$
    \[
P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)
    \]
An HMM is defined by:

- Initial distribution:
  \[ P(X_1) \]
  \[ P(X_t | X_{t-1}) \]

- Transitions:
  \[ P(E | X) \]
Recap: Filtering Example

The diagram illustrates a filtering example with the following nodes and edges:

- $Rain_0$
- $Rain_1$
- $Rain_2$
- $Umbrella_1$
- $Umbrella_2$

The probabilities associated with these nodes are:

- $P(Rain_0 = \text{True}) = 0.500$
- $P(Rain_0 = \text{False}) = 0.500$
- $P(Rain_1 = \text{True} | Rain_0 = \text{True}) = 0.500$
- $P(Rain_1 = \text{False} | Rain_0 = \text{True}) = 0.500$
- $P(Rain_2 = \text{True} | Rain_1 = \text{True}) = 0.627$
- $P(Rain_2 = \text{False} | Rain_1 = \text{True}) = 0.373$
- $P(Rain_2 = \text{True} | Rain_1 = \text{False}) = 0.883$
- $P(Rain_2 = \text{False} | Rain_1 = \text{False}) = 0.117$

The edges represent the conditional probabilities between the nodes.
Example Pac-man
Recap: Particle Filtering

- Sometimes $|X|$ is too big to use exact inference
  - $|X|$ may be too big to even store $B(X)$
  - E.g. $X$ is continuous
  - $|X|^2$ may be too big to do updates

- Solution: approximate inference
  - Track samples of $X$, not all values
  - Samples are called particles
  - Time per step is linear in the number of samples
  - But: number needed may be large
  - In memory: list of particles, not states

- This is how robot localization works in practice
Recap: Particle Filtering

At each time step $t$, we have a set of $N$ particles / samples

- Initialization: Sample from prior, reweight and resample
- Three step procedure, to move to time $t+1$:
  1. **Sample transitions**: for each each particle $x$, sample next state
     \[ x' = \text{sample}(P(X' | x)) \]
  2. **Reweight**: for each particle, compute its weight given the actual observation $e$
     \[ w(x) = P(e | x) \]
  3. **Resample**: normalize the weights, and sample $N$ new particles from the resulting distribution over states
Our representation of $P(X)$ is now a list of $N$ particles (samples)
- Generally, $N << |X|$
- Storing map from $X$ to counts would defeat the point

$P(x)$ approximated by number of particles with value $x$
- So, many $x$ will have $P(x) = 0!$
- More particles, more accuracy

For now, all particles have a weight of 1
Particle Filtering: Elapse Time

- Each particle is moved by sampling its next position from the transition model

\[ x' = \text{sample}(P(X'|x)) \]

- This is like prior sampling – samples’ frequencies reflect the transition probs
- Here, most samples move clockwise, but some move in another direction or stay in place

- This captures the passage of time
  - If we have enough samples, close to the exact values before and after (consistent)
Particle Filtering: Observe

- Slightly trickier:
  - We don’t sample the observation, we fix it
  - We weight our samples based on the evidence

\[ w(x) = P(e|x) \]

\[ B(X) \propto P(e|X)B'(X) \]

- Note that, as before, the weights/probabilities don’t sum to one, since most have been downweighted (in fact they sum to an approximation of $P(e)$)
Particle Filtering: Resample

- Rather than tracking weighted samples, we resample.
- N times, we choose from our weighted sample distribution (i.e. draw with replacement).
- This is equivalent to renormalizing the distribution.
- Now the update is complete for this time step, continue with the next one.

Old Particles:
- (3,3) w=0.1
- (2,1) w=0.9
- (2,1) w=0.9
- (3,1) w=0.4
- (3,2) w=0.3
- (2,2) w=0.4
- (1,1) w=0.4
- (3,1) w=0.4
- (2,1) w=0.9
- (3,2) w=0.3

New Particles:
- (2,1) w=1
- (2,1) w=1
- (2,1) w=1
- (3,2) w=1
- (2,2) w=1
- (2,1) w=1
- (1,1) w=1
- (3,1) w=1
- (2,1) w=1
- (1,1) w=1
Recap: Particle Filtering

At each time step $t$, we have a set of $N$ particles / samples

- Initialization: Sample from prior, reweight and resample
- Three step procedure, to move to time $t+1$:
  1. Sample transitions: for each each particle $x$, sample next state
     \[ x' = \text{sample}(P(X'|x)) \]
  2. Reweight: for each particle, compute its weight given the actual observation $e$
     \[ w(x) = P(e|x) \]
  3. Resample: normalize the weights, and sample $N$ new particles from the resulting distribution over states
Which Algorithm?

Particle filter, uniform initial belief, 300 particles
PS4: Ghostbusters

- **Plot:** Pacman's grandfather, Grandpac, learned to hunt ghosts for sport.

- He was blinded by his power, but could hear the ghosts’ banging and clanging.

- **Transition Model:** All ghosts move randomly, but are sometimes biased

- **Emission Model:** Pacman knows a “noisy” distance to each ghost
Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence.
- Idea: Repeat a fixed Bayes net structure at each time.
- Variables from time $t$ can condition on those from $t-1$.

- Discrete valued dynamic Bayes nets are also HMMs.
DBN Particle Filters

- A particle is a complete sample for a time step
- **Initialize**: Generate prior samples for the $t=1$ Bayes net
  - Example particle: $G_1^a = (3,3)$ $G_1^b = (5,3)$

- **Elapse time**: Sample a successor for each particle
  - Example successor: $G_2^a = (2,3)$ $G_2^b = (6,3)$

- **Observe**: Weight each entire sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: $P(E_1^a | G_1^a) \times P(E_1^b | G_1^b)$

- **Resample**: Select samples (tuples of values) in proportion to their likelihood weights
Model for Ghostbusters

- Reminder: ghost is hidden, sensors are noisy
- T: Top sensor is red
  B: Bottom sensor is red
  G: Ghost is in the top
- Queries:
  \( P( +g) = ?? \)
  \( P( +g \mid +t) = ?? \)
  \( P( +g \mid +t, -b) = ?? \)
- Problem: joint distribution too large / complex

Joint Distribution

<table>
<thead>
<tr>
<th>T</th>
<th>B</th>
<th>G</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>+t</td>
<td>+b</td>
<td>+g</td>
<td>0.16</td>
</tr>
<tr>
<td>+t</td>
<td>+b</td>
<td>-g</td>
<td>0.16</td>
</tr>
<tr>
<td>+t</td>
<td>-b</td>
<td>+g</td>
<td>0.24</td>
</tr>
<tr>
<td>+t</td>
<td>-b</td>
<td>-g</td>
<td>0.04</td>
</tr>
<tr>
<td>-t</td>
<td>+b</td>
<td>+g</td>
<td>0.04</td>
</tr>
<tr>
<td>-t</td>
<td>+b</td>
<td>-g</td>
<td>0.24</td>
</tr>
<tr>
<td>-t</td>
<td>-b</td>
<td>+g</td>
<td>0.06</td>
</tr>
<tr>
<td>-t</td>
<td>-b</td>
<td>-g</td>
<td>0.06</td>
</tr>
</tbody>
</table>
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 called graphical models
  - We describe how variables locally interact
  - Local interactions chain together to give global, indirect interactions
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

A Bayes net = Topology (graph) + Local Conditional Probabilities
Example Bayes’ Net: Car
Probabilities in BNs

- 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 independence assumptions
  - Compare to the exact decomposition according to the chain rule!
Example Bayes’ Net: Insurance
Example: Independence

- \(N\) fair, independent coin flips:

\[
\begin{array}{c|c}
\text{h} & 0.5 \\
\text{t} & 0.5 \\
\end{array}
\]

\[
\begin{array}{c|c}
\text{h} & 0.5 \\
\text{t} & 0.5 \\
\end{array}
\]

... 

\[
\begin{array}{c|c}
\text{h} & 0.5 \\
\text{t} & 0.5 \\
\end{array}
\]

\[
P(X_1, X_2, \ldots, X_n)
\]

\[
2^n
\]
Example: Coin Flips

- \( N \) independent coin flips

\[ X_1 \quad X_2 \quad \ldots \quad X_n \]

- No interactions between variables: absolute independence
Independence

- Two variables are independent if:

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

  - This says that their joint distribution factors into a product two simpler distributions
  - Another form:

\[ \forall x, y : P(x|y) = P(x) \]

  - We write: \( X \perp Y \)

- Independence is a simplifying modeling assumption
  - Empirical joint distributions: at best “close” to independent
  - What could we assume for \{Weather, Traffic, Cavity, Toothache\}?
Example: Independence?

\[ P_1(T, W) \]

<table>
<thead>
<tr>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>warm</td>
<td>sun</td>
<td>0.4</td>
</tr>
<tr>
<td>warm</td>
<td>rain</td>
<td>0.1</td>
</tr>
<tr>
<td>cold</td>
<td>sun</td>
<td>0.2</td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

\[ P(T) \]

<table>
<thead>
<tr>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>warm</td>
<td>0.5</td>
</tr>
<tr>
<td>cold</td>
<td>0.5</td>
</tr>
</tbody>
</table>

\[ P(W) \]

<table>
<thead>
<tr>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.6</td>
</tr>
<tr>
<td>rain</td>
<td>0.4</td>
</tr>
</tbody>
</table>

\[ P_2(T, W) \]

<table>
<thead>
<tr>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>warm</td>
<td>sun</td>
<td>0.3</td>
</tr>
<tr>
<td>warm</td>
<td>rain</td>
<td>0.2</td>
</tr>
<tr>
<td>cold</td>
<td>sun</td>
<td>0.3</td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Conditional Independence

- $P(\text{Toothache, Cavity, Catch})$

- If I have a cavity, the probability that the probe catches in it doesn't depend on whether I have a toothache:
  - $P(+\text{catch} \mid +\text{toothache}, +\text{cavity}) = P(+\text{catch} \mid +\text{cavity})$

- The same independence holds if I don’t have a cavity:
  - $P(+\text{catch} \mid +\text{toothache}, \neg\text{cavity}) = P(+\text{catch} \mid \neg\text{cavity})$

- Catch is conditionally independent of Toothache given Cavity:
  - $P(\text{Catch} \mid \text{Toothache, Cavity}) = P(\text{Catch} \mid \text{Cavity})$

- Equivalent statements:
  - $P(\text{Toothache} \mid \text{Catch, Cavity}) = P(\text{Toothache} \mid \text{Cavity})$
  - $P(\text{Toothache, Catch} \mid \text{Cavity}) = P(\text{Toothache} \mid \text{Cavity}) \cdot P(\text{Catch} \mid \text{Cavity})$
  - One can be derived from the other easily
Conditional Independence

- Unconditional (absolute) independence very rare (why?)

- *Conditional independence* is our most basic and robust form of knowledge about uncertain environments:

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

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

- What about this domain:
  - Traffic
  - Umbrella
  - Raining

- What about fire, smoke, alarm?
Each sensor depends only on where the ghost is.

That means, the two sensors are conditionally independent, given the ghost position.

T: Top square is red
B: Bottom square is red
G: Ghost is in the top

Can assume:
\[ P( +g ) = 0.5 \]
\[ P( +t \mid +g ) = 0.8 \]
\[ P( +t \mid -g ) = 0.4 \]
\[ P( +b \mid +g ) = 0.4 \]
\[ P( +b \mid -g ) = 0.8 \]

\[ P(T,B,G) = P(G) P(T\mid G) P(B\mid G) \]

<table>
<thead>
<tr>
<th>T</th>
<th>B</th>
<th>G</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>+t</td>
<td>+b</td>
<td>+g</td>
<td>0.16</td>
</tr>
<tr>
<td>+t</td>
<td>+b</td>
<td>-g</td>
<td>0.16</td>
</tr>
<tr>
<td>+t</td>
<td>-b</td>
<td>+g</td>
<td>0.24</td>
</tr>
<tr>
<td>+t</td>
<td>-b</td>
<td>-g</td>
<td>0.04</td>
</tr>
<tr>
<td>-t</td>
<td>+b</td>
<td>+g</td>
<td>0.04</td>
</tr>
<tr>
<td>-t</td>
<td>+b</td>
<td>-g</td>
<td>0.24</td>
</tr>
<tr>
<td>-t</td>
<td>-b</td>
<td>+g</td>
<td>0.06</td>
</tr>
<tr>
<td>-t</td>
<td>-b</td>
<td>-g</td>
<td>0.06</td>
</tr>
</tbody>
</table>
Example: Traffic

- Variables:
  - R: It rains
  - T: There is traffic

- Model 1: independence

- Model 2: rain is conditioned on traffic
  - Why is an agent using model 2 better?

- Model 3: traffic is conditioned on rain
  - Is this better than model 2?
Example: Alarm Network

- **Variables**
  - B: Burglary
  - A: Alarm goes off
  - M: Mary calls
  - J: John calls
  - E: Earthquake!
### Example: Alarm Network

**Burglary**

<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>

**Earthqk**

<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>

**Alarm**

- John calls
- Mary calls

**P(J|A)**

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

**P(M|A)**

| A  | M  | P(M|A) |
|----|----|-------|
| +a | +m | 0.7   |
| +a | ¬m | 0.3   |
| ¬a | +m | 0.01  |
| ¬a | ¬m | 0.99  |

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

| 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 | 0.06    |
| ¬b | +e | +a | 0.29    |
| ¬b | +e | ¬a | 0.71    |
| ¬b | ¬e | +a | 0.001   |
| ¬b | ¬e | ¬a | 0.999   |
Example: Traffic II

- Let’s build a causal graphical model

- Variables
  - T: Traffic
  - R: It rains
  - L: Low pressure
  - D: Roof drips
  - B: Ballgame
  - C: Cavity
Example: Independence

- For this graph, you can fiddle with $\theta$ (the CPTs) all you want, but you won’t be able to represent any distribution in which the flips are dependent!

<table>
<thead>
<tr>
<th></th>
<th>$P(X_1)$</th>
<th></th>
<th>$P(X_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>0.5</td>
<td>$h$</td>
<td>0.5</td>
</tr>
<tr>
<td>$t$</td>
<td>0.5</td>
<td>$t$</td>
<td>0.5</td>
</tr>
</tbody>
</table>

$X_1 \perp X_2$

All distributions
Topology Limits Distributions

- Given some graph topology $G$, only certain joint distributions can be encoded.
- The graph structure guarantees certain (conditional) independences.
- (There might be more independence.)
- Adding arcs increases the set of distributions, but has several costs.
- Full conditioning can encode any distribution.
Important question about a BN:
- Are two nodes independent given certain evidence?
- If yes, can prove using algebra (tedious in general)
- If no, can prove with a counter example

Example:

\[ \text{X} \overset{\text{Y}}{\rightarrow} \text{Z} \]

Question: are X and Z necessarily independent?
- Answer: no. Example: low pressure causes rain, which causes traffic.
- X can influence Z, Z can influence X (via Y)
- Addendum: they *could* be independent: how?
Causal Chains

- This configuration is a “causal chain”

\[ P(x, y, z) = P(x)P(y|x)P(z|y) \]

- Is X independent of Z given Y?

\[ P(z|x, y) = \frac{P(x, y, z)}{P(x, y)} = \frac{P(x)P(y|x)P(z|y)}{P(x)P(y|x)} = P(z|y) \]

Yes!

- Evidence along the chain “blocks” the influence
Another basic configuration: two effects of the same cause

- Are X and Z independent?
- Are X and Z independent given Y?

\[
P(z|x, y) = \frac{P(x, y, z)}{P(x, y)} = \frac{P(y)P(x|y)P(z|y)}{P(y)P(x|y)} = P(z|y)
\]

Yes!

- Observing the cause blocks influence between effects.

Y: Project due
X: Newsgroup busy
Z: Lab full
Common Effect

- Last configuration: two causes of one effect (v-structures)
  - Are X and Z independent?
    - Yes: the ballgame and the rain cause traffic, but they are not correlated
    - Still need to prove they must be (try it!)
  - Are X and Z independent given Y?
    - No: seeing traffic puts the rain and the ballgame in competition as explanation?
  - This is backwards from the other cases
    - Observing an effect activates influence between possible causes.
The General Case

- Any complex example can be analyzed using these three canonical cases

- General question: in a given BN, are two variables independent (given evidence)?

- Solution: analyze the graph
Reachability

- **Recipe**: shade evidence nodes
- **Attempt 1**: if two nodes are connected by an undirected path not blocked by a shaded node, they are conditionally independent
- **Almost works, but not quite**
  - Where does it break?
  - Answer: the v-structure at T doesn’t count as a link in a path unless “active”
Reachability (D-Separation)

- **Question**: Are $X$ and $Y$ conditionally independent given evidence vars $\{Z\}$?
  - Yes, if $X$ and $Y$ “separated” by $Z$
  - Look for active paths from $X$ to $Y$
  - No active paths = independence!

- **A path is active if each triple is active**:  
  - Causal chain $A \rightarrow B \rightarrow C$ where $B$ is unobserved (either direction)
  - Common cause $A \leftarrow B \rightarrow C$ where $B$ is unobserved
  - Common effect (aka v-structure) $A \rightarrow B \leftarrow C$ where $B$ or one of its descendents is observed

- All it takes to block a path is a single inactive segment
Example: Independent?

\[ R \perp B \]
\[ R \perp B | T \]
\[ R \perp B | T' \]

Yes

Diagram:

- Node \( R \) connected to \( T \)
- Node \( B \) connected to \( T \)
- Node \( T \) connected to \( T' \)
Example: Independent?

\[
\begin{align*}
L \perp T' | T & \quad \text{Yes} \\
L \perp B & \quad \text{Yes} \\
L \perp B | T & \\
L \perp B | T' & \\
L \perp B | T, R & \quad \text{Yes}
\end{align*}
\]
Example

- **Variables:**
  - R: Raining
  - T: Traffic
  - D: Roof drips
  - S: I’m sad

- **Questions:**

  \[ T \perp D \]

  \[ T \perp D | R \]

  Yes

  \[ T \perp D | R, S \]
The same joint distribution can be encoded in many different Bayes’ nets

Analysis question: given some edges, what other edges do you need to add?

- One answer: fully connect the graph
- Better answer: don’t make any false conditional independence assumptions
Example: Coins

- Extra arcs don’t prevent representing independence, just allow non-independence

- Adding unneeded arcs isn’t wrong, it’s just inefficient
Summary

- Bayes nets compactly encode joint distributions
- Guaranteed independencies of distributions can be deduced from BN graph structure
- D-separation gives precise conditional independence guarantees from graph alone
- A Bayes’ net’s joint distribution may have further (conditional) independence that is not detectable until you inspect its specific distribution