Bayesian Network

• 2018/03/27

➢ Announcement:

☑ Slides for this lecture are here


☑ Project 4 due by April 8

Slides are largely based on information from http://ai.berkeley.edu and Russel
Last time

- Hidden Markov model
  - Components of HMM
  - Independencies in HMM
  - Inference in HMM
    - Forward algorithm
    - Viterbi algorithm
  - HMM applications
    - Localization and particle filter
    - SLAM

- Required reading (red means it will be on your exams):
  - R&N: Chapter 15.3
Filtering review

**Elapse time:** compute \( P(X_t \mid e_{1:t-1}) \)

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

**Observe:** compute \( P(X_t \mid e_{1:t}) \)

\[
P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)
\]

**Belief:** \( \langle P(\text{rain}), P(\text{sun}) \rangle \)

- \( P(X_1) = \langle 0.5, 0.5 \rangle \)  
  *Prior on \( X_1 \)*

- \( P(X_1 \mid E_1 = \text{umbrella}) = \langle 0.82, 0.18 \rangle \)  
  *Observe*

- \( P(X_2 \mid E_1 = \text{umbrella}) = \langle 0.63, 0.37 \rangle \)  
  *Elapse time*

- \( P(X_2 \mid E_1 = \text{umb}, E_2 = \text{umb}) = \langle 0.88, 0.12 \rangle \)  
  *Observe*
Particle filter
Particle filtering

- **Filtering: approximate solution**
- **Sometimes \(|X|\) is too big to use exact inference**
  - \(|X|\) may be too big to even store \(B(X)\)
  - E.g. \(X\) is continuous
- **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**
- **Particle is just new name for sample**
• Our representation of \( P(X) \) is now a list of \( N \) particles (samples)
  – Generally, \( N \ll |X| \)
  – Storing map from \( X \) to counts would defeat the point

• \( P(x) \) approximated by number of particles with value \( x \)
  – So, many \( x \) may have \( P(x) = 0! \)
  – More particles, more accuracy

• For now, all particles have a weight of 1
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 probabilities
- Here, most samples move clockwise, but some move in another direction or stay in place

This captures the passage of time
- If enough samples, close to exact values before and after (consistent)
Slightly trickier:

- Similar to likelihood weighting, downweight samples based on the evidence

$$w(x) = P(e|x)$$

$$B(X) \propto P(e|X)B'(X)$$

- As before, the probabilities don’t sum to one, since all have been downweighted (in fact they now sum to (N times) an approximation of P(e))
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.
Summary of particle filtering

- Particles: track samples of states rather than an explicit distribution
Moderate number of particle
One particle
Robot localization

- In robot localization:
  - We know the map, but not the robot’s position
  - Observations may be vectors of range finder readings
  - State space is typically continuous (works basically like a very fine grid) and so we cannot store $B(X)$
  - Particle filtering is a main technique
Robot localization with sonar

Global localization with sonar sensors
SLAM

- SLAM: Simultaneous Localization And Mapping
  - We do not know the map or our location
  - State consists of position AND map!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods

DP-SLAM, Ron Parr

Ira A. Fulton Schools of Engineering
ARIZONA STATE UNIVERSITY
Laser-based SLAM with a Ground Robot

Erik Nelson, Nathan Michael

https://www.youtube.com/watch?v=iD47JWVqTCK
Most likely explanation
Most likely explanation

- HMMs defined by
  - States X
  - Observations E
  - Initial distribution \( P(X_1) \)
  - Transitions: \( P(X|X_{-1}) \)
  - Emissions: \( P(E|X) \)

- New query: most likely explanation: \( \arg \max_{x_{1:t}} P(x_{1:t}|e_{1:t}) \)

- New method: the Viterbi algorithm
State trellis

- State trellis: graph of states and transitions over time

![State trellis diagram]

- Each arc represents some transition $x_{t-1} \rightarrow x_t$
- Each arc has weight $P(x_t|x_{t-1})P(e_t|x_t)$
- Each path is a sequence of states
- The product of weights on a path is that sequence’s probability along with the evidence
- Forward algorithm computes sums of paths, Viterbi computes best paths
Viterbi algorithm

Forward Algorithm (Sum)

\[
f_t[x_t] = P(x_t, e_{1:t})
\]

\[
= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) f_{t-1}[x_{t-1}]
\]

Viterbi Algorithm (Max)

\[
m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})
\]

\[
= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]
\]
Outline for today

• Bayesian network
  o Semantics of BN
  o Probabilities in BN
  o Dynamic BN

• Required reading (red means it will be on your exams):
  o R&N: Chapter 14.1-2
Bayesian network (Bayes net)
Human-aware Robotics

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
- For about 10 min, we’ll be vague about how these interactions are specified
Probabilistic models

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

- 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

- 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
BN example: insurance (hidden variable)
BN example: car (causal reasoning)
Graphical model

• Nodes: variables (with domains)
  – Can be assigned (observed) or unassigned (unobserved)

• Arcs: interactions
  – Indicate “direct influence” between variables
  – Formally: encode conditional independence (more later)

• For now: imagine that arrows mean direct causation (in general, they don’t!)
Coin flips

N independent coin flips

$X_1 \quad X_2 \quad \ldots \quad X_n$

No interactions between variables: absolute independence
Traffic

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

Model 1: independence

Model 2: rain causes traffic

Why is an agent using model 2 better?
Let’s build a causal graphical model!

Variables

- T: Traffic
- R: It rains
- L: Low pressure
- D: Roof drips
- B: Ballgame
- C: Cavity
• Variables
  – B: Burglary
  – A: Alarm goes off
  – M: Mary calls
  – J: John calls
  – E: Earthquake!
Semantics of BN
Semantics of BN

- 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 Bayes net = Topology (graph) + Local Conditional Probabilities
Probabilities in BN

- 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:

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

- Example:

\[
P(\text{+cavity, +catch, -toothache})
\]
• Why are we guaranteed that setting
  \[ P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i|\text{parents}(X_i)) \]
  results in a proper joint distribution?

• Chain rule (valid for all distributions):
  \[ P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i|x_1 \ldots x_{i-1}) \]

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

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

• Not every BN can represent every joint distribution
  – The topology enforces certain conditional independencies
Only distributions whose variables are absolutely independent can be represented by a Bayes’ net with no arcs.
Traffic

\[ P(R) \]

\begin{array}{c|c}
+r & 1/4 \\
-r & 3/4 \\
\end{array}

\[ P(T|R) \]

\begin{array}{c|c|c}
+r & +t & 3/4 \\
+r & -t & 1/4 \\
-r & +t & 1/2 \\
-r & -t & 1/2 \\
\end{array}

\[ P(T, R) \]

\begin{array}{c|c|c}
+r & +t & 3/16 \\
+r & -t & 1/16 \\
-r & +t & 6/16 \\
-r & -t & 6/16 \\
\end{array}

\[ P(+r, -t) = \]

\[ P(T, R) \]
Reverse Traffic?

Reverse causal direction

\[ P(T) \]
\[ \begin{array}{c|c} 
+ t & 9/16 \\
- t & 7/16 \\
\end{array} \]

\[ P(R|T) \]
\[ \begin{array}{c|c|c} 
+ t & + r & 1/3 \\
- r & 2/3 \\
- t & + r & 1/7 \\
- r & 6/7 \\
\end{array} \]

\[ P(T, R) \]
\[ \begin{array}{c|c|c} 
+ r & + t & 3/16 \\
+ r & - t & 1/16 \\
- r & + t & 6/16 \\
- r & - t & 6/16 \\
\end{array} \]
Alarm

<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>

| 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|

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

- So far: how a Bayes’ net encodes a joint distribution
- Next: how to answer queries about that distribution
  - Today:
    - First assembled BNs using an intuitive notion of conditional independence as causality
    - Then saw that key property is conditional independence
  - Main goal: answer queries about conditional independence and influence
- After that: how to answer numerical queries (inference)
Dynamic Bayesian network
Joint distribution of HMM

\[ P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1)P(E_2|X_2)P(X_3|X_2)P(E_3|X_3) \]

- Joint distribution:

\[ P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1) \prod_{t=2}^{T} P(X_t|X_{t-1})P(E_t|X_t) \]

- More generally:
Dynamic Bayesian network

- 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 $t-1$

- Dynamic Bayes nets are a generalization of HMMs
Particle filter

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: $P(E_1^a | G_1^a) * P(E_1^b | G_1^b)$

**Resample**: Select prior samples (tuples of values) in proportion to their likelihood
Outline for today

• Bayesian network
  o Semantics of BN
  o Probabilities in BN
  o Dynamic BN

• Required reading (red means it will be on your exams):
  o R&N: Chapter 14.1-2