Bayesian Networks

+ Reinforcement Learning: Markov Decision Processes
Reminders

• Homework 7: HMMs
  – Out: Wed, Nov. 03
  – Due: Fri, Nov. 12 at 11:59pm
Q&A

Q: Lecture: Would you be so kind as to end lecture on time?

A:
Q&A

Q: Lecture: The larger-than-life in-class demonstrations are absolutely amazing. Could you do more of them?

A: Honestly, as the core material becomes increasingly complex it will be quite difficult, but we sure can try!

Q: Lecture: The larger-than-life in-class demonstrations are really boring and take up a lot of time. Could you do less of them?

A: Honestly, that would make our lives a lot easier, so we sure can try!
| Q: **Lectures**: Could you upload the slides a day ahead of time? | A: Yes, we can do that.  
(Just a heads up that the slides might change slightly after that first upload.) |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Q: <strong>Homework</strong>: Some of the multiple choice homework questions are ambiguous or you end up changing the questions later</td>
<td>A: We are trying to improve our own testing to try to catch these sorts of bugs early. They tend to come up specifically in these heavily constrained multiple choice problems.</td>
</tr>
</tbody>
</table>
**Q&A**

<table>
<thead>
<tr>
<th>Q: Recitation: Some of the TAs handwriting is even worse than yours (some is much better), could you all work on that?</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: Ah. We hadn’t thought of that – sorry! We’ve just instituted some digital handwriting practice for those who haven’t had much. (We used to use chalkboards, but don’t have those this semester.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q: Recitation: It’d be great if recitations left more time for students to solve the problems.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: Sorry about that. We’ve been trying to pack more and more in and rushing a bit through the interactive-problem-solving parts as a result.</td>
</tr>
</tbody>
</table>
GRAPHICAL MODELS:
DETERMINING CONDITIONAL INDEPENDENCIES
What Independencies does a Bayes Net Model?

• In order for a Bayesian network to model a probability distribution, the following must be true:
  
  Each variable is conditionally independent of all its non-descendants in the graph given the value of all its parents.

• This follows from

\[ P(X_1 \ldots X_n) = \prod_{i=1}^{n} P(X_i \mid \text{parents}(X_i)) \]

  \[ = \prod_{i=1}^{n} P(X_i \mid X_1 \ldots X_{i-1}) \]

• But what else does it imply?
What Independencies does a Bayes Net Model?

Three cases of interest...

Cascade

Common Parent

V-Structure
What Independencies does a Bayes Net Model?

Three cases of interest...

**Cascade**

\[ X \perp Z \mid Y \]

Knowing \( Y \) **decouples** \( X \) and \( Z \)

**Common Parent**

\[ X \perp Z \mid Y \]

Knowing \( Y \) **couples** \( X \) and \( Z \)

**V-Structure**

\[ X \nparallel Z \mid Y \]

Knowing \( Y \) **couples** \( X \) and \( Z \)
Proof of conditional independence

\( X \perp Z \mid Y \)

(The other two cases can be shown just as easily.)
The “Burglar Alarm” example

- Your house has a twitchy burglar alarm that is also sometimes triggered by earthquakes.
- Earth arguably doesn’t care whether your house is currently being burgled.
- While you are on vacation, one of your neighbors calls and tells you your home’s burglar alarm is ringing. Uh oh!

Quiz: True or False?

$\text{Burglar} \perp \text{Earthquake} \mid \text{Phone Call}$
Question 1

A

B

C
Markov Blanket

**Def:** the **co-parents** of a node are the parents of its children.

**Def:** the **Markov Blanket** of a node is the set containing the node’s parents, children, and co-parents.
**Markov Blanket**

**Def:** the co-parents of a node are the parents of its children.

**Def:** the Markov Blanket of a node is the set containing the node’s parents, children, and co-parents.

**Example:** The Markov Blanket of $X_6$ is

\[ \{X_3, X_4, X_5, X_8, X_9, X_{10}\} \]
Markov Blanket

**Def:** the co-parents of a node are the parents of its children.

**Def:** the Markov Blanket of a node is the set containing the node’s parents, children, and co-parents.

**Theorem:** a node is conditionally independent of every other node in the graph given its Markov blanket.

**Example:** The Markov Blanket of $X_6$ is $\{X_3, X_4, X_5, X_8, X_9, X_{10}\}$.
D-Separation

**Definition #1:**
Variables X and Z are *d-separated* given a set of evidence variables E (variables that are observed) iff every path from X to Z is “blocked”.

A path is “blocked” whenever:
1.  $\exists Y$ on path s.t. $Y \in E$ and Y is a “common parent”

   ![Diagram 1](#)

2.  $\exists Y$ on path s.t. $Y \in E$ and Y is in a “cascade”

   ![Diagram 2](#)

3.  $\exists Y$ on path s.t. $\{Y, \text{descendants}(Y)\} \not\in E$ and Y is in a “v-structure”

   ![Diagram 3](#)

**If** variables X and Z are *d-separated* given a set of variables E

**Then** X and Z are *conditionally independent* given the set E
D-Separation

If variables X and Z are \textit{d-separated} given a set of variables E
Then X and Z are \textit{conditionally independent} given the set E

\textbf{Definition \#2:}
Variables X and Z are \textit{d-separated} given a set of evidence variables E iff there does \textbf{not} exist a path between X and Z in the \textit{undirected ancestral moral} graph with E \textbf{removed}.

1. **Ancestral graph**: keep only X, Z, E and their ancestors
2. **Moral graph**: add undirected edge between all pairs of each node’s parents
3. **Undirected graph**: convert all directed edges to undirected
4. **Givens Removed**: delete any nodes in E

\textbf{Example Query}: \(A \perp B \mid \{D, E\}\)

\begin{align*}
\text{Original:} & \quad \text{Ancestral:} & \quad \text{Moral:} & \quad \text{Undirected:} & \quad \text{Givens Removed:} \\
\begin{tikzpicture}[->, thick, node distance=2cm]
    \node (A) {$A$};
    \node (B) [right of=A] {$B$};
    \node (C) [below of=A] {$C$};
    \node (D) [left of=C] {$D$};
    \node (E) [right of=C] {$E$};
    \node (F) [below of=E] {$F$};
    \path
    (A) edge (B)
    (A) edge (C)
    (C) edge (D)
    (C) edge (E);
\end{tikzpicture}
\begin{tikzpicture}[->, thick, node distance=2cm]
    \node (A) {$A$};
    \node (B) [right of=A] {$B$};
    \node (C) [below of=A] {$C$};
    \node (D) [left of=C] {$D$};
    \node (E) [right of=C] {$E$};
    \path
    (A) edge (B)
    (A) edge (C)
    (C) edge (D)
    (C) edge (E);
\end{tikzpicture}
\begin{tikzpicture}[->, thick, node distance=2cm]
    \node (A) {$A$};
    \node (B) [right of=A] {$B$};
    \node (C) [below of=A] {$C$};
    \node (D) [left of=C] {$D$};
    \node (E) [right of=C] {$E$};
    \path
    (A) edge (B)
    (A) edge (C)
    (C) edge (D)
    (C) edge (E);
\end{tikzpicture}
\begin{tikzpicture}[->, thick, node distance=2cm]
    \node (A) {$A$};
    \node (B) [right of=A] {$B$};
    \node (C) [below of=A] {$C$};
    \node (D) [left of=C] {$D$};
    \node (E) [right of=C] {$E$};
    \path
    (A) edge (B)
    (A) edge (C)
    (C) edge (D)
    (C) edge (E);
\end{tikzpicture}
\begin{tikzpicture}[->, thick, node distance=2cm]
    \node (A) {$A$};
    \node (B) [right of=A] {$B$};
    \node (C) [below of=A] {$C$};
    \node (D) [left of=C] {$D$};
    \node (E) [right of=C] {$E$};
    \path
    (A) edge (B)
    (A) edge (C)
    (C) edge (D)
    (C) edge (E);
\end{tikzpicture}
\end{align*}

\(\Rightarrow A \text{ and } B \text{ connected}
\Rightarrow \text{ not d-separated} \)
SUPERVISED LEARNING FOR BAYES NETS
Recipe for Closed-form MLE

1. Assume data was generated i.i.d. from some model (i.e. write the generative story)
   \[ x^{(i)} \sim p(x|\theta) \]

2. Write log-likelihood
   \[ \ell(\theta) = \log p(x^{(1)}|\theta) + \ldots + \log p(x^{(N)}|\theta) \]

3. Compute partial derivatives (i.e. gradient)
   \[ \frac{\partial \ell(\theta)}{\partial \theta_1} = \ldots \]
   \[ \frac{\partial \ell(\theta)}{\partial \theta_2} = \ldots \]
   \[ \ldots \]
   \[ \frac{\partial \ell(\theta)}{\partial \theta_M} = \ldots \]

4. Set derivatives to zero and solve for \( \theta \)
   \[ \frac{\partial \ell(\theta)}{\partial \theta_m} = 0 \text{ for all } m \in \{1, \ldots, M\} \]
   \[ \theta^{\text{MLE}} = \text{solution to system of } M \text{ equations and } M \text{ variables} \]

5. Compute the second derivative and check that \( \ell(\theta) \) is concave down
   at \( \theta^{\text{MLE}} \)
Machine Learning

The **data** inspires the structures we want to predict.

*Inference* finds \{best structure, marginals, partition function\} for a new observation.

(Inference is usually called as a subroutine in learning.)

Our **model** defines a score for each structure.

It also tells us what to optimize.

**Learning** tunes the parameters of the model.

---

Domain Knowledge

Mathematical Modeling

Combinatorial Optimization

Optimization

ML (Inference is usually called as a subroutine in learning)
Machine Learning

Data

(Inference is usually called as a subroutine in learning)

Model

Objective

Learning

Inference

Machine Learning
Learning Fully Observed BNs

\[
p(X_1, X_2, X_3, X_4, X_5) = p(X_5|X_3)p(X_4|X_2, X_3) \\
p(X_3)p(X_2|X_1)p(X_1)
\]
Learning Fully Observed BNs

\[ p(X_1, X_2, X_3, X_4, X_5) = \]
\[ p(X_5 | X_3)p(X_4 | X_2, X_3) \]
\[ p(X_3)p(X_2 | X_1)p(X_1) \]
Learning Fully Observed BNs

\[ p(X_1, X_2, X_3, X_4, X_5) = \\
\quad p(X_5|X_3)p(X_4|X_2, X_3) \\
\quad p(X_3)p(X_2|X_1)p(X_1) \]

How do we learn these conditional and marginal distributions for a Bayes Net?
Learning Fully Observed BNs

Learning this fully observed Bayesian Network is **equivalent** to learning five (small / simple) independent networks from the same data.

\[
p(X_1, X_2, X_3, X_4, X_5) = \\
p(X_5|X_3)p(X_4|X_2, X_3) \\
p(X_3)p(X_2|X_1)p(X_1)
\]
Learning Fully Observed BNs

How do we **learn** these **conditional** and **marginal** distributions for a Bayes Net?

\[ \mathbf{\theta}^* = \underset{\theta}{\text{argmax}} \log p(X_1, X_2, X_3, X_4, X_5) \]

\[ = \underset{\theta}{\text{argmax}} \log p(X_5 | X_3, \theta_5) + \log p(X_4 | X_2, X_3, \theta_4) \]

\[ + \log p(X_3 | \theta_3) + \log p(X_2 | X_1, \theta_2) \]

\[ + \log p(X_1 | \theta_1) \]

\[ \theta_1^* = \underset{\theta_1}{\text{argmax}} \log p(X_1 | \theta_1) \]

\[ \theta_2^* = \underset{\theta_2}{\text{argmax}} \log p(X_2 | X_1, \theta_2) \]

\[ \theta_3^* = \underset{\theta_3}{\text{argmax}} \log p(X_3 | \theta_3) \]

\[ \theta_4^* = \underset{\theta_4}{\text{argmax}} \log p(X_4 | X_2, X_3, \theta_4) \]

\[ \theta_5^* = \underset{\theta_5}{\text{argmax}} \log p(X_5 | X_3, \theta_5) \]
Example: Tornado Alarms

1. Imagine that you work at the 911 call center in Dallas.
2. You receive six calls informing you that the Emergency Weather Sirens are going off.
3. What do you conclude?
Example: Tornado Alarms

1. Imagine that you work at the 911 call center in Dallas.
2. You receive six calls informing you that the Emergency Weather Sirens are going off.
3. What do you conclude?

Figure from https://www.nytimes.com/2017/04/08/us/dallas-emergency-sirens-hacking.html
Learning Fully Observed BNs

Ex: Tornado Alarms

\[\begin{array}{c}
\text{Tornado} \\
\text{Hurricane} \\
\text{Alarm} \\
\text{911 PhoneCall}
\end{array}\]

\[\begin{array}{c}
\text{H} \sim \text{Bernoulli}(\eta) \\
\text{T} \sim \text{Bernoulli}(\zeta) \\
\text{A} \sim \text{Bernoulli}(\alpha_{\text{A},\text{T}}) \\
\text{C} \sim \text{Uniform}(1, \ldots, 63) + A \times \text{Uniform}(1, \ldots, 63)
\end{array}\]

\[\text{MLE's in Closed Form}\]

\[\hat{\eta}, \hat{\zeta}, \hat{\alpha} = \arg\max_{\eta, \zeta, \alpha} \mathcal{L}(\eta, \zeta, \alpha, x)\]

\[\hat{\eta} = \arg\max_{\eta} \frac{1}{N} \sum_{i=1}^{N} \log p(h^{(i)} | \eta) = \#(T=1) / N\]

\[\hat{\zeta} = \arg\max_{\zeta} \frac{1}{N} \sum_{i=1}^{N} \log p(t^{(i)} | \zeta) = \#(H=1) / N\]

\[\hat{\alpha} = \arg\max_{\alpha} \frac{1}{N} \sum_{i=1}^{N} \log p(a^{(i)} | t^{(i)}, h^{(i)}, \alpha)\]

\[\hat{\alpha}_{h_t} = \frac{\#(A=1, T=t, H=h)}{\#(T=t, H=h)}\]

What are the MLEs?

\[\hat{\eta} = \frac{1}{3}\]

\[\hat{\zeta} = \frac{1}{2}\]

\[\hat{\alpha} = \begin{array}{c}
T=0 \\
T=1
\end{array}\]

\[\begin{array}{c|c|c}
& T=0 & T=1 \\
\hline
A=0 & 1 & 1 \\
A=1 & 0 & 0
\end{array}\]
INFERENCE FOR BAYESIAN NETWORKS
A Few Problems for Bayes Nets

Suppose we already have the parameters of a Bayesian Network...

1. How do we compute the probability of a specific assignment to the variables?
   \[ P(T=t, H=h, A=a, C=c) \]

2. How do we draw a sample from the joint distribution?
   \[ t, h, a, c \sim P(T, H, A, C) \]

3. How do we compute marginal probabilities?
   \[ P(A) = \ldots \]

4. How do we draw samples from a conditional distribution?
   \[ t, h, a \sim P(T, H, A | C = c) \]

5. How do we compute conditional marginal probabilities?
   \[ P(H | C = c) = \ldots \]
Gibbs Sampling

\[ p(x) \]

\[ p(x_1 | x_2^{(t)}) \]

\[ x(t+1) \]

\[ x(t) \]
Gibbs Sampling

\[ p(x) \]

\[ x_1 \]

\[ x_2 \]

\[ x(t) \]

\[ x(t+1) \]

\[ x(t+2) \]

\[ p(x_2 | x_1^{(t+1)}) \]
Gibbs Sampling

\(p(x)\)

\(x_1\)

\(x_2\)

\(x(t)\)

\(x(t+1)\)

\(x(t+2)\)

\(x(t+3)\)

\(x(t+4)\)
Gibbs Sampling

**Question:**
How do we draw samples from a conditional distribution?

\[ y_1, y_2, \ldots, y_J \sim p(y_1, y_2, \ldots, y_J \mid x_1, x_2, \ldots, x_J) \]

**(Approximate) Solution:**

- Initialize \( y_1^{(0)}, y_2^{(0)}, \ldots, y_J^{(0)} \) to arbitrary values
- For \( t = 1, 2, \ldots \):
  - \( y_1^{(t+1)} \sim p(y_1 \mid y_2^{(t)}, \ldots, y_J^{(t)}, x_1, x_2, \ldots, x_J) \)
  - \( y_2^{(t+1)} \sim p(y_2 \mid y_1^{(t+1)}, y_3^{(t)}, \ldots, y_J^{(t)}, x_1, x_2, \ldots, x_J) \)
  - \( y_3^{(t+1)} \sim p(y_3 \mid y_1^{(t+1)}, y_2^{(t+1)}, y_4^{(t)}, \ldots, y_J^{(t)}, x_1, x_2, \ldots, x_J) \)
  - \( \ldots \)
  - \( y_J^{(t+1)} \sim p(y_J \mid y_1^{(t+1)}, y_2^{(t+1)}, \ldots, y_{J-1}^{(t+1)}, x_1, x_2, \ldots, x_J) \)

**Properties:**

- This will eventually yield samples from \( p(y_1, y_2, \ldots, y_J \mid x_1, x_2, \ldots, x_J) \)
- But it might take a long time -- just like other Markov Chain Monte Carlo methods
Gibbs Sampling

Full conditionals only need to condition on the Markov Blanket

- Must be “easy” to sample from conditionals
- Many conditionals are log-concave and are amenable to adaptive rejection sampling
Learning Objectives

Bayesian Networks

You should be able to...

1. Identify the conditional independence assumptions given by a generative story or a specification of a joint distribution
2. Draw a Bayesian network given a set of conditional independence assumptions
3. Define the joint distribution specified by a Bayesian network
4. Use domain knowledge to construct a (simple) Bayesian network for a real-world modeling problem
5. Depict familiar models as Bayesian networks
6. Use d-separation to prove the existence of conditional independence in a Bayesian network
7. Employ a Markov blanket to identify conditional independence assumptions of a graphical model
8. Develop a supervised learning algorithm for a Bayesian network
9. Use samples from a joint distribution to compute marginal probabilities
10. Sample from the joint distribution specified by a generative story
11. Implement a Gibbs sampler for a Bayesian network
Reinforcement Learning
Learning Paradigms

• Supervised Learning
  • Training data is (input, output)
  • Variants: active learning and online learning

• Unsupervised Learning
  • Training data is (input)

• Reinforcement Learning
  • Training data is (input, action, reward)
Reinforcement Learning (RL)

Source: https://www.xkcd.com/242/
RL: Examples

Source: https://techobserver.net/2019/06/argo-ai-self-driving-car-research-center/
Source: https://www.wired.com/2012/02/high-speed-trading/

Source: https://twitter.com/alphagomovie
• The algorithm has to gather its own training data

• The outcome of taking some action is often stochastic or unknown until after the fact

• Decisions can have a delayed effect on future outcomes (exploration-exploitation tradeoff)
• Problem formulation
  • Time discounted cumulative reward
  • Markov decision processes (MDPs)

• Algorithms:
  • Value iteration and policy iteration
    (dynamic programming)
  • (Deep) Q-learning (temporal difference learning)
• State space, \( S \)
• Action space, \( \mathcal{A} \)
• Reward function, \( R : S \times \mathcal{A} \to \mathbb{R} \)
• Transition probabilities, \( p(s' | s, a) \)
  • Deterministic transitions:
    \[
p(s' | s, a) = \begin{cases} 
    1 & \text{if } \delta(s, a) = s' \\
    0 & \text{otherwise}
  \end{cases}
\]
  where \( \delta(s, a) \) is a transition function
• Policy, \( \pi : S \to \mathcal{A} \)
• Value function, \( V^\pi : S \to \mathbb{R} \)
  • Measures the expected total payoff of starting in some state \( s \) and executing policy \( \pi \)
RL: Toy Example

$S = \text{all empty squares in the grid}$

$\mathcal{A} = \{ \text{up, down, left, right} \}$
RL: Poll Q2

Is this policy optimal?
When poll is active, respond at pollev.com/10301601polls

Question 2

A

B

C
Justify your answer to the previous question

Join by Web

1. Go to PollEv.com
2. Enter 10301601POLLS
3. Respond to activity

1 Instructions not active. Log in to activate
RL: Toy Example

Optimal policy given reward of -2 for each step
RL: Toy Example

Optimal policy given reward of -0.1 for each step
• Find a policy \( \pi^* = \arg\max_\pi V^\pi(s) \ \forall \ s \in S \)

• \( V^\pi(s) = \mathbb{E}[\text{discounted total reward of starting in state } s \ \text{and executing policy } \pi \text{ forever}] \)
  
  \[
  = \mathbb{E}_{p(s' | s, a)}[R(s_0 = s, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \cdots ]
  = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{p(s' | s, a)}[R(s_t, \pi(s_t))] 
  \]

  where \( 0 < \gamma < 1 \) is some discount factor for future rewards
• In RL, the model for our data is an MDP:
  1. Start in some initial state $s_0$
  2. For time step $t$:
     1. Agent observes state $s_t$
     2. Agent takes action $a_t = \pi(s_t)$
     3. Agent receives reward $r_t = R(s_t, a_t)$
     4. Agent transitions to state $s_{t+1} \sim p(s' | s_t, a_t)$
  3. Total reward is $\sum_{t=0}^{\infty} \gamma^t r_t$

• Makes the same Markov assumption we used for HMMs! The next state only depends on the current state and action.