Lecture 10a:
Bayesian Networks (Part 3)

CS540 4/2/19

Upcoming Project

As a group, you will build a system that takes as input
1. A blocks world start state (as in PA2)
2. An English language paragraph describing a final blocks world
   configuration and/or actions to take in blocks world.

Your program will print out a series of actions (as in PA2)
determined by the English language paragraph

*What is legal in the English language paragraph? That is up to your team. In general, the more general the better...*

Specifications

Your team will not submit code.
Instead, your team will write a 5 page (maximum) paper describing

1. Motivation
   1. What were you trying to accomplish that was interesting
      - Note: satisfy the assignment is NOT an acceptable answer
   2. There are many ways to satisfy the assignment. Why did you pick the one you did?
2. Methodology
   1. How does your system work?
   2. How does it address your motivation
3. Results
   1. What can your system do?
   2. Where does it fail?
4. Conclusion
   1. Strengths & weaknesses
   2. Possible next steps

Specifications (cont.)

Your team will also prepare a 20 minute presentation
- Using Powerpoint or similar slide presentation system
- Presented by a team member
- Either the on-campus member
- Or an off-campus member via Zoom, Skype, etc.
- Hitting the same points as the paper
- Motivation
- Methodology
- Results
- Conclusion
- And including a demonstration
  - Which implies some form of visualization

Advice

Start soon
- Pick a TA2 implementation to build on
- Pick coordination tools, learn to use them as a group
  - I use slack, trelio, zoom, git ...
- Schedule regular meeting times

General outline
- Build simulator based on TA2
- Add visualization tools, if you haven’t already built any
- Divide English language input into sentences
  - English sentences end with periods, question marks or exclamation points.
- Use the Stanford Parser to convert sentences to parse trees
- Extract frames from parse trees
  - When possible
  - Fail gracefully, otherwise
- Connect frames to TA2 planner
Announcements

Team project presentations
Tuesday, May 7th

Team papers
Due Thursday, May 9th

Class is canceled next week
I strongly recommend using the time to work on team projects.

A Bayesian Network Example

The "Asia" network:

![Bayesian Network Diagram]

Approaches to inference

**Exact inference**
- Variable elimination
- Join tree algorithm

**Approximate inference**
- Simplify the structure of the network to make exact inference efficient (variational methods, loopy belief propagation)

Probabilistic methods
- Stochastic simulation / sampling methods
- Markov chain Monte Carlo methods

Markov chains

A Markov chain is a random process (infinite sequence of random variables)
\( (X(0), X(1), ..., X(t), ...) \) that satisfies:

\[ P(X(t) \mid X(0), ..., X(t-1)) = P(X(t) \mid X(t-1)) \]

The probability of a particular state at time \( t \) depends only on the state at time \( t-1 \)

If the transition probabilities are fixed for all \( t \), the chain is called homogeneous and is characterized by a transition matrix \( T \).

\[
T = \begin{pmatrix}
0.7 & 0.3 & 0 \\
0.3 & 0.4 & 0.3 \\
0 & 0.3 & 0.7
\end{pmatrix}
\]

Markov chains

For sampling from \( P(x) \), we require that for any starting state \( x(0) \):

\[
\lim_{t \to \infty} P_t(x) = P(x)
\]

Equivalently, the stationary distribution of the Markov chain must be \( P(x) \):

\[
P_{t+1}(x') = \sum_x P_t(x)Q(x \to x')
\]

the transition probability from \( x \) to \( x' \)

\[
P(x') = \sum_x P(x)Q(x \to x')
\]

Stationary distribution

\[
Q = \begin{pmatrix}
0.7 & 0.3 & 0 \\
0.3 & 0.4 & 0.3 \\
0 & 0.3 & 0.7
\end{pmatrix}
\]

The stationary distribution of this chain is \((0.33, 0.33, 0.33)\)
Markov chains for sampling

To ensure that the chain converges to a unique stationary distribution the following conditions are sufficient:

- **Irreducibility**: every state is eventually reachable from any start state; for all \( x, y \) there exists a \( t \) such that \( P_t(y) > 0 \) when starting at \( x \).

- **Aperiodicity**: the chain doesn’t get caught in cycles.

The process is **ergodic** if it is both irreducible and aperiodic.

Detailed balance

To ensure that the stationary distribution of the Markov chain is \( P(x) \) it is sufficient for \( P \) and \( Q \) to satisfy the detailed balance (reversibility) condition:

\[
P(x)Q(x \rightarrow x') = P(x')Q(x' \rightarrow x)
\]

Given that detailed balance holds:

\[
\sum_x P(x)Q(x \rightarrow x') = \sum_x P(x')Q(x' \rightarrow x) = P(x)\sum_x Q(x' \rightarrow x) = P(x')
\]

Gibbs sampling

Idea: To transition from one state (variable assignment) to another by:

- Pick a variable \( X_j \)
- Sample its value from the conditional distribution \( P(x_j \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n) \)

In a Bayesian network \( x_j \) depends only on a subset of the variables.

Markov Blanket

The Markov blanket of a variable in a network consists of:

- The node’s immediate parents
- Its immediate children
- It’s children’s other parents

Variables are independent of everything else in the network given their Markov blanket.

So, to sample a node, only need to condition on its Markov blanket:

\[
P(x_j \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n) \approx P(x_j \mid MB(x_j))
\]

Markov Blanket Example

The Gibbs sampling algorithm

\[
\text{GIBBS}(X, e, h_n, N) \text{ returns estimate of } P(X|e)
\]

\( N[x] \) - counts the number of times each value of \( X \) was observed

\( x[j] \) - the current state of the network \( x[0] \) initialized with random values for the nonevidence variables

for \( j = 1 \) to \( N \) do

for each nonevidence variable \( X \),

sample \( X \) from \( P(X|MB(X)) \)

\( N[x] = N[x] + 1 \), where \( x \) is the value of \( X \) in \( x[j] \)
Convergence of Gibbs sampling

Gibbs sampling satisfies detailed balance:

\[ P(x|e)P(x'|\bar{x}_i, e) = P(x_i, \bar{x}_i|e)P(x'|\bar{x}_i, e) \]
\[ = P(x_i|\bar{x}_i, e)P(x'|\bar{x}_i, e) \]
\[ = P(x_i|\bar{x}_i, e)P(x_i'|\bar{x}_i, e) \]
\[ = P(x_i'|e)P(x_i|\bar{x}_i', e) \]

Practical issues

How many iterations?
When to stop?

Gibbs sampling example

Consider a 2 variable network:

Initialize randomly
Sample variables alternately