Lecture10b: Gibbs Sampling

CS540 3/29/18

Announcements

On-campus students:

- Make sure I am wearing the microphone
- Check the clip!

Reading Assignment (for Tuesday)

- Pedro Felzenswalb & Daniel Huttenlocher, *Pictorial Structures for Object Recognition,* International Journal of Computer Vision 61(1): 55-79, 2005

Project #2

In project #1, most groups either

- Compared search strategies to find simple paths
- Created a two level system
  - Subgoals
  - Simple paths

For project #2

- Same simulator
- Same format for specifying problems
- But I specify the problems
- You have two weeks to improve your code
  - Anticipate hard problems
  - Parse problems into subgoals
  - Plan at level of subgoals
  - Plan paths to satisfy each subgoal

Project #2 (cont)

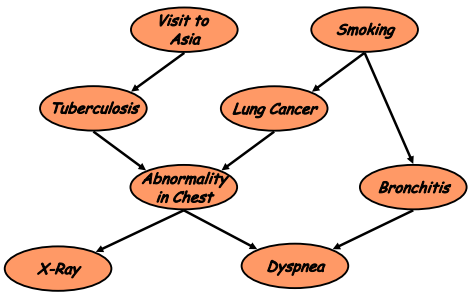
Two weeks from today (April 10)

- Freeze your code
- I will release my state and goal files

You have one more week (until April 17) to:

- Test your system on my problems/goals
- Write a 4 page paper (max) describing the results
  - This is a challenge paper. No motivation or literature section is required. However, it does need an introduction that summarizes what a reader should learn, a results section, an analysis of the results, and a conclusion.
- Prepare a 5 minute in-class presentation (April 17)
- Every team member writes one separate column describing their contribution

How do we do inference on a complex Bayesian network?



Inference by sampling

Suppose we can sample instances $\langle X_1,\ldots,X_n\rangle$ according to $P(X_1,\ldots,X_n)$

Want to compute $P(e)$

The probability that a random sample $\langle X_1,\ldots,X_n\rangle$ satisfies e is approximately $P(e)$

We can view each sample as tossing a biased coin with probability $P(e)$ of "Heads"

BN sampling

Let $X_1,\ldots,X_n$ be order of variables consistent with arc direction

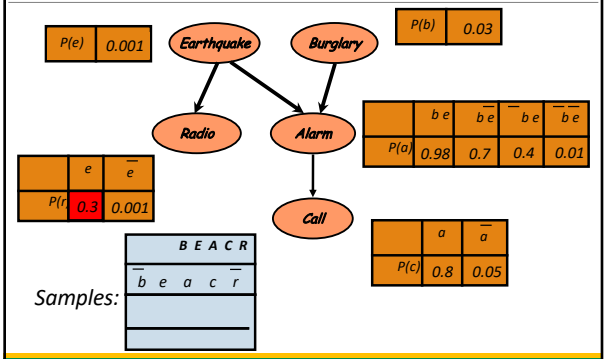
for $i = 1,\ldots,n$ do

sample x_i from $P(X_i \mid Pa(X_i))$

(Note: since $Pa(X_i) \subseteq \{X_1,\ldots,X_{i-1}\}$, we already assigned values to them)

return $x_1,\ldots,x_n$

BN sampling



BN sampling

Sampling a complete instance is linear in number of variables

- Regardless of structure of the network

However, if $P(\boldsymbol{e})$ is small, we need many samples to get a decent estimate

Can we sample from $P(X_1,\ldots,X_n \mid \boldsymbol{e})$?

If evidence is in roots of network, easily

If evidence is in leaves of network, we have a problem

- Our sampling method proceeds according to order of nodes in graph

Rejection sampling: keep those instantiations that are consistent with the values of the evidence variables

Estimate $P(X|\boldsymbol{e})$ by $N(X,\boldsymbol{e}) / N(\boldsymbol{e})$ where $N(.)$ counts the number of times an event was sampled.

Likelihood Weighting

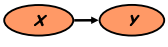
Can we ensure that all of our samples satisfy $\boldsymbol{e}$?

One simple solution:

- When we need to sample a variable that is assigned value by $\boldsymbol{e}$, use the specified value

For example: we know $Y = 1$

- Sample X from $P(X)$
- Then take $Y = 1$



Is this a sample from P($X, Y \mid Y = 1$)?

Likelihood Weighting

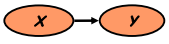
Problem: these samples of X are from $P(X)$

Solution:

- Penalize samples in which $P(Y=1|X)$ is small

We now sample as follows:

- Let x[i] be a sample from $P(X)$
- Let w[i] be $P(Y = 1|X = x\,[i])$



$$P(X = x \mid \boldsymbol{Y}=1) \approx \frac{\sum_i w[i] P(X = x|\boldsymbol{x}[i])}{\sum_i w[i]}$$

Markov chain Monte Carlo sampling

Generates events by making random changes to the state variable.

The next state is generated by sampling a value for one of the nonevidence variables conditioned on the current values.

Markov chains

A Markov chain is a random process (infinite sequence of random variables)

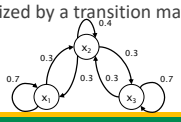
$(X(0), X(1), \ldots, X(t), \ldots)$ that satisfies:

$P(X(t) \mid X(0), \ldots X(t\text{-}1)) = P(X(t) \mid X(t\text{-}1))$

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.

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



Sampling via 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 $P(x)$:

$$P_{t+1}(\mathbf{x}') = \sum_{\mathbf{x}} P_t(\mathbf{x}) Q(\mathbf{x} \to \mathbf{x}')$$

the transition probability from $\boldsymbol{x}$ to $\boldsymbol{x}'$

$$P(\mathbf{x}') = \sum_{\mathbf{x}} P(\mathbf{x}) Q(\mathbf{x} \to \mathbf{x}')$$

Using a Markov chain to sample

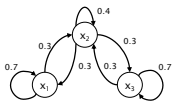
If a Markov chain converges to is probability distribution:

- Start from an arbitrary state
- Do a long random walk
  - According to the transition probabilities
- Count how often you are in each state x
- Divide the occurances of x by t to get p(x)

The resulting state will be sampled from $P(x)$.

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:

$$P(\mathbf{x})Q(\mathbf{x} \rightarrow \mathbf{x}') = P(\mathbf{x}')Q(\mathbf{x}' \rightarrow \mathbf{x})$$

Given that detailed balance holds:

$$\begin{aligned} \sum_{\mathbf{x}} P(\mathbf{x})Q(\mathbf{x} \rightarrow \mathbf{x}') &= \sum_{\mathbf{x}} P(\mathbf{x}')Q(\mathbf{x}' \rightarrow \mathbf{x}) \\ &= P(\mathbf{x}') \sum_{\mathbf{x}} Q(\mathbf{x}' \rightarrow \mathbf{x}) \\ &= P(\mathbf{x}') \end{aligned}$$

Back to Bayesian Networks

What does this have to do with inference on a Bayesian Network?

- Bayesian nets have random variables (nodes)
- Bayesian nets have relations (edges)
- Think of a complete assignment of values to random variables at a time t as a state x(t) in our Markov chain.
- The relations tell us how to probabilistically move from one state to another

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

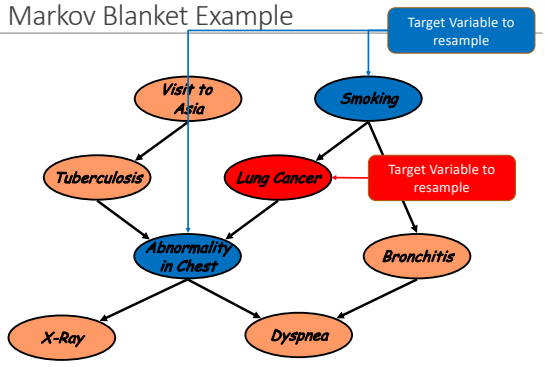
Variables are independent of their non-descendents given their parents

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:

$P(x_j \mid x_1, \ldots x_{j-1}, x_{j+1}, \ldots x_n) =$
$P(x_j \mid MB(x_j))$

Markov Blanket Example



The Gibbs sampling algorithm

GIBBS$(X, \mathbf{e}, bn, N)$ returns estimate of $P(X|\mathbf{e})$

$\mathbf{N}[x]$ - counts the number of times each value of X was observed

$\mathbf{x}[j]$ - the current state of the network $\mathbf{x}[0]$ initialized with random values for the nonevidence variables

for $j = 1$ *to* N do

for each nonevidence variable X_i

sample X_i from $P(X_i|MB(X_i))$

$\mathbf{N}[x] = \mathbf{N}[x] + 1$, where x is the value of X in $\mathbf{x}[j]$

Convergence of Gibbs sampling

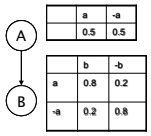
Gibbs sampling satisfies detailed balance:

$$
\begin{aligned}
P(\mathbf{x}|\mathbf{e})P(x_i'|\bar{\mathbf{x}}_i,\mathbf{e}) &= P(x_i,\bar{\mathbf{x}}_i|\mathbf{e})P(x_i'|\bar{\mathbf{x}}_i,\mathbf{e}) \\
&= P(x_i|\bar{\mathbf{x}}_i,\mathbf{e})P(\bar{\mathbf{x}}_i,\mathbf{e})P(x_i'|\bar{\mathbf{x}}_i,\mathbf{e}) \\
&= P(x_i|\bar{\mathbf{x}}_i,\mathbf{e})P(x_i',\bar{\mathbf{x}}_i|\mathbf{e}) \\
&= P(\mathbf{x}'|\mathbf{e})P(x_i|\bar{\mathbf{x}}_i',\mathbf{e})
\end{aligned}
$$

$\bar{\mathbf{x}}_i$

Gibbs sampling example

Consider a 2 variable network:



Initialize randomly

Sample variables alternately

Practical issues

How many iterations?

When to stop?