Announcements

- **Section**
  - We’ll be using some software to play with Bayes nets: Bring your laptop!
  - Download necessary files (links also in the handout):
    - [http://www-inst.eecs.berkeley.edu/~cs188/sp11/bayes/bayes.jar](http://www-inst.eecs.berkeley.edu/~cs188/sp11/bayes/bayes.jar)
    - [http://www-inst.eecs.berkeley.edu/~cs188/sp11/bayes/network.xml](http://www-inst.eecs.berkeley.edu/~cs188/sp11/bayes/network.xml)

- **Assignments**
  - P4 and contest going out Monday
Outline

- Bayes net refresher:
  - Representation
  - Exact Inference
    - Enumeration
    - Variable elimination
  - Approximate inference through sampling

Bayes’ Net Semantics

- 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*
Probabilities in BNs

- For all joint distributions, we have (chain rule):
  \[ P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|x_1, \ldots, x_{i-1}) \]

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

- This lets us reconstruct any entry of the full joint
- Not every BN can represent every joint distribution
  - The topology enforces certain conditional independencies

Inference by Enumeration

- Given unlimited time, inference in BNs is easy
- Recipe:
  - State the marginal probabilities you need
  - Figure out ALL the atomic probabilities you need
  - Calculate and combine them

- Building the full joint table takes time and space exponential in the number of variables
General Variable Elimination

- Query: \( P(Q|E_1 = e_1, \ldots, E_k = e_k) \)

- Start with initial factors:
  - Local CPTs (but instantiated by evidence)

- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H

- Join all remaining factors and normalize

- Complexity is exponential in the number of variables appearing in the factors---can depend on ordering but even best ordering is often impractical

- Worst case is bad: we can encode 3-SAT with a Bayes net (NP-complete)

Approximate Inference

- Simulation has a name: sampling (e.g. predicting the weather, basketball games…)

- Basic idea:
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability
  - Show this converges to the true probability P

- Why sample?
  - Learning: get samples from a distribution you don’t know
  - Inference: getting a sample is faster than computing the right answer (e.g. with variable elimination)
Sampling

- How do you sample?
  - Simplest way is to use a random number generator to get a continuous value uniformly distributed between 0 and 1 (e.g. random() in Python)
  - Assign each value in the domain of your random variable a sub-interval of [0,1] with a size equal to its probability
    - The sub-intervals cannot overlap

Sampling Example

- Each value in the domain of $W$ has a sub-interval of [0,1] with a size equal to its probability

<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>0.1</td>
</tr>
<tr>
<td>fog</td>
<td>0.3</td>
</tr>
<tr>
<td>meteor</td>
<td>0.0</td>
</tr>
</tbody>
</table>

$u$ is a uniform random value in [0,1]
- if $0.0 \leq u < 0.6$, $w = \text{sun}$
- if $0.6 \leq u < 0.7$, $w = \text{rain}$
- if $0.7 \leq u < 1.0$, $w = \text{fog}$

E.g. if random() returns $u = 0.83$, then our sample is $w = \text{fog}$
Prior Sampling

This process generates samples with probability:

\[ S_{PS}(x_1 \ldots x_n) = \prod_{i=1}^{n} P(x_i|\text{Parents}(X_i)) = P(x_1 \ldots x_n) \]

...i.e. the BN’s joint probability

Let the number of samples of an event be \( N_{PS}(x_1 \ldots x_n) \)

Then

\[
\lim_{N \to \infty} \hat{P}(x_1, \ldots, x_n) = \lim_{N \to \infty} \frac{N_{PS}(x_1, \ldots, x_n)}{N} \\
= S_{PS}(x_1, \ldots, x_n) \\
= P(x_1 \ldots x_n)
\]

I.e., the sampling procedure is consistent
Example

- We’ll get a bunch of samples from the BN:
  - +c, -s, +r, +w
  - +c, +s, +r, +w
  - -c, +s, +r, -w
  - +c, -s, +r, +w
  - -c, -s, -r, +w

- If we want to know P(W)
  - We have counts <+w:4, -w:1>
  - Normalize to get P(W) = <+w:0.8, -w:0.2>
  - This will get closer to the true distribution with more samples
  - Can estimate anything else, too
  - What about P(C| +w)? P(C| +r, +w)? P(C| -r, -w)?
  - Fast: can use fewer samples if less time (what’s the drawback?)

Rejection Sampling

- Let’s say we want P(C)
  - No point keeping all samples around
  - Just tally counts of C as we go

- Let’s say we want P(C| +s)
  - Same thing: tally C outcomes, but ignore (reject) samples which don’t have S=+s
  - This is called rejection sampling
  - It is also consistent for conditional probabilities (i.e., correct in the limit)
Likelihood Weighting

- Problem with rejection sampling:
  - If evidence is unlikely, you reject a lot of samples
  - You don’t exploit your evidence as you sample
  - Consider $P(B|+a)$

- Idea: fix evidence variables and sample the rest

- Problem: sample distribution not consistent!
- Solution: weight by probability of evidence given parents
Likelihood Weighting

- Sampling distribution if z sampled and e fixed evidence

\[ S_{WS}(z, e) = \prod_{i=1}^{l} P(z_i|\text{Parents}(Z_i)) \]

- Now, samples have weights

\[ w(z, e) = \prod_{i=1}^{m} P(e_i|\text{Parents}(E_i)) \]

- Together, weighted sampling distribution is consistent

\[ S_{WS}(z, e) \cdot w(z, e) = \prod_{i=1}^{l} P(z_i|\text{Parents}(z_i)) \prod_{i=1}^{m} P(e_i|\text{Parents}(e_i)) = P(z, e) \]

Likelihood Weighting

- Likelihood weighting is good
  - We have taken evidence into account as we generate the sample
  - E.g. here, W’s value will get picked based on the evidence values of S, R
  - More of our samples will reflect the state of the world suggested by the evidence

- Likelihood weighting doesn’t solve all our problems
  - Evidence influences the choice of downstream variables, but not upstream ones (C isn’t more likely to get a value matching the evidence)

- We would like to consider evidence when we sample every variable
Gibbs Sampling

- **Idea:** instead of sampling from scratch, create samples that are each like the last one.

- **Procedure:** resample one variable at a time, conditioned on all the rest, but keep evidence fixed.

- **Properties:** Now samples are not independent (in fact they’re nearly identical), but sample averages are still consistent estimators!

- **What’s the point:** both upstream and downstream variables condition on evidence.

---

Gibbs Sampling

- **Say we want to sample** $P(S \mid R = +r)$

- **Step 1: Initialize**
  - Set evidence ($R = +r$)
  - Set all other variables ($S, C, W$) to random values (e.g. by prior sampling or just uniformly sampling; say $S = -s, W = +w, C = -c$)
  - Our initial sample is then: $(R = +r, S = -s, W = +w, C = -c)$

- **Steps 2+: Repeat the following for some number of iterations**
  - Choose a non-evidence variable ($S, W, or C$ in this case)
  - Sample this variable conditioned on nothing else changing
    - The first time through, if we pick $S$, we sample from $P(S \mid R = +r, W = +w, C = -c)$
    - The new sample can only be different in a single variable
Gibbs Sampling

- How is this better than sampling from the full joint?
  - In a Bayes net, sampling a variable given all the other variables (e.g. $P(R|S,C,W)$) is usually much easier than sampling from the full joint distribution
    - Only requires a join on the variable to be sampled (in this case, a join on $R$)
    - The resulting factor only depends on the variable’s parents, its children, and its children’s parents (this is often referred to as its Markov blanket)

Gibbs Sampling Example

- Want to sample from $P(R \mid +s,-c,-w)$
  - Shorthand for $P(R \mid S=+s,C=-c,W=-w)$

\[
P(R \mid +s, -c, -w) = \frac{P(R, +s, -c, -w)}{P(+s, -c, -w)}
\]

\[
= \frac{P(R, +s, -c, -w)}{\sum_r P(R=r, +s, -c, -w)}
\]

\[
= \frac{P(-c) P(+s \mid -c) P(R \mid -c) P(-w \mid +s, R)}{\sum_r P(-c) P(+s \mid -c) P(R=r \mid -c) P(-w \mid +s, R=r)}
\]

\[
= \frac{P(R \mid -c) P(-w \mid +s, R)}{\sum_r P(R=r \mid -c) P(-w \mid +s, R=r)}
\]

- Many things cancel out -- just a join on $R$!
Further Reading*

- Gibbs sampling is a special case of more general methods called Markov chain Monte Carlo (MCMC) methods
  - Metropolis-Hastings is one of the more famous MCMC methods (in fact, Gibbs sampling is a special case of Metropolis-Hastings)
- You may read about Monte Carlo methods – they’re just sampling