CS 188: Artificial Intelligence
Spring 2010

Lecture 18: Bayes Nets V
3/30/2010

Pieter Abbeel – UC Berkeley
Many slides over this course adapted from Dan Klein, Stuart Russell, Andrew Moore

Announcements

- Midterms
  - In glookup

- Assignments
  - W5 due Thursday
  - W6 going out Thursday

- Midterm course evaluations in your email soon
Outline

- Bayes net refresher:
  - Representation
  - Inference
    - Enumeration
    - Variable elimination
- Approximate inference through sampling
- Value of information

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
- Description of a noisy “causal” process

\[
P(X|A_1 \ldots A_n)
\]

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

- 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

Approximate Inference

- **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)
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

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

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

- 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

```
P(C)
+ c 0.5
- c 0.5

P(S|C)
+ c + s 0.1
+ c - s 0.9
- c + s 0.5
- c - s 0.5

P(W|S, R)
+ s + r + w 0.99
- s + r 0.01
+ s - w 0.90
+ s - w 0.10
- s + r 0.90
- s - w 0.10
- s - w 0.01
- s - w 0.99

P(R|C)
+ c + r 0.8
+ c - r 0.2
- c + r 0.2
- c - r 0.8

Samples:
+ c, + s, + r, + w

w = 1.0 \times 0.1 \times 0.99
```

```
P(C)
+ c 0.5
- c 0.5

P(S|C)
+ c + s 0.1
+ c - s 0.9
- c + s 0.5
- c - s 0.5

P(W|S, R)
+ s + r + w 0.99
- s + r 0.01
+ s - w 0.90
+ s - w 0.10
- s + r 0.90
- s - w 0.10
- s - w 0.01
- s - w 0.99

P(R|C)
+ c + r 0.8
+ c - r 0.2
- c + r 0.2
- c - r 0.8

Samples:
+ c, + s, + r, + w

w = 1.0 \times 0.1 \times 0.99
```

```
P(C)
+ c 0.5
- c 0.5

P(S|C)
+ c + s 0.1
+ c - s 0.9
- c + s 0.5
- c - s 0.5

P(W|S, R)
+ s + r + w 0.99
- s + r 0.01
+ s - w 0.90
+ s - w 0.10
- s + r 0.90
- s - w 0.10
- s - w 0.01
- s - w 0.99

P(R|C)
+ c + r 0.8
+ c - r 0.2
- c + r 0.2
- c - r 0.8

Samples:
+ c, + s, + r, + w

w = 1.0 \times 0.1 \times 0.99
```

```
P(C)
+ c 0.5
- c 0.5

P(S|C)
+ c + s 0.1
+ c - s 0.9
- c + s 0.5
- c - s 0.5

P(W|S, R)
+ s + r + w 0.99
- s + r 0.01
+ s - w 0.90
+ s - w 0.10
- s + r 0.90
- s - w 0.10
- s - w 0.01
- s - w 0.99

P(R|C)
+ c + r 0.8
+ c - r 0.2
- c + r 0.2
- c - r 0.8

Samples:
+ c, + s, + r, + w

w = 1.0 \times 0.1 \times 0.99
```

```
P(C)
+ c 0.5
- c 0.5

P(S|C)
+ c + s 0.1
+ c - s 0.9
- c + s 0.5
- c - s 0.5

P(W|S, R)
+ s + r + w 0.99
- s + r 0.01
+ s - w 0.90
+ s - w 0.10
- s + r 0.90
- s - w 0.10
- s - w 0.01
- s - w 0.99

P(R|C)
+ c + r 0.8
+ c - r 0.2
- c + r 0.2
- c - r 0.8

Samples:
+ c, + s, + r, + w

w = 1.0 \times 0.1 \times 0.99
```
Now, samples have weights

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

Now, samples have weights

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

Together, weighted sampling distribution is consistent

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

\[ = P(z, e) \]

Likelihood weighting 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
Markov Chain Monte Carlo*

- **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. E.g., for P(b|c):

```
+ b -> a -> c
+ b -> a -> c
+ b -> a -> c
```

- **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.

---

Decision Networks

- **MEU:** choose the action which maximizes the expected utility given the evidence
- **Can directly operationalize this with decision networks**
  - Bayes nets with nodes for utility and actions
  - Lets us calculate the expected utility for each action
- **New node types:**
  - Chance nodes (just like BNs)
  - Actions (rectangles, cannot have parents, act as observed evidence)
  - Utility node (diamond, depends on action and chance nodes)
Decision Networks

- **Action selection:**
  - Instantiate all evidence
  - Set action node(s) each possible way
  - Calculate posterior for all parents of utility node, given the evidence
  - Calculate expected utility for each action
  - Choose maximizing action

Example: Decision Networks

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>

Umbrella = leave

\[
EU(\text{leave}) = \sum_w P(w)U(\text{leave}, w) \\
= 0.7 \cdot 100 + 0.3 \cdot 0 = 70
\]

Umbrella = take

\[
EU(\text{take}) = \sum_w P(w)U(\text{take}, w) \\
= 0.7 \cdot 20 + 0.3 \cdot 70 = 35
\]

Optimal decision = leave

\[
\text{MEU}(a) = \max_a EU(a) = 70
\]
Evidence in Decision Networks

Find \( P(W|F=\text{bad}) \)
- Select for evidence
- First we join \( P(W) \) and \( P(\text{bad}|W) \)
- Then we normalize

<table>
<thead>
<tr>
<th>W</th>
<th>P(W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>0.7</td>
</tr>
<tr>
<td>rain</td>
<td>0.3</td>
</tr>
</tbody>
</table>

| F     | P(F|W) |
|-------|--------|
| sun   | 0.8    |
| rain  | 0.2    |

| F     | P(F|W) |
|-------|--------|
| sun   | 0.1    |
| rain  | 0.9    |

\( P(W) \) \hspace{1cm} \( P(\text{bad}|W) \)

\[ P(W, \text{bad}) = \frac{P(W) \cdot P(\text{bad}|W)}{P(W)} = P(\text{bad}|W) \cdot P(W) \]

Example: Decision Networks

\( W \) \hspace{1cm} \( P(W|F=\text{bad}) \)

| W     | P(W|F=bad) |
|-------|------------|
| sun   | 0.34       |
| rain  | 0.66       |

\( A \) \hspace{1cm} \( U(A,W) \)

<table>
<thead>
<tr>
<th>A</th>
<th>W</th>
<th>U(A,W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>leave</td>
<td>sun</td>
<td>100</td>
</tr>
<tr>
<td>leave</td>
<td>rain</td>
<td>0</td>
</tr>
<tr>
<td>take</td>
<td>sun</td>
<td>20</td>
</tr>
<tr>
<td>take</td>
<td>rain</td>
<td>70</td>
</tr>
</tbody>
</table>

Umbrella = leave

\[ \text{EU(leave|bad)} = \sum_w P(w|\text{bad})U(\text{leave}, w) = 0.34 \cdot 100 + 0.66 \cdot 0 = 34 \]

Umbrella = take

\[ \text{EU(take|bad)} = \sum_w P(w|\text{bad})U(\text{take}, w) = 0.34 \cdot 20 + 0.66 \cdot 70 = 53 \]

Optimal decision = take

\[ \text{MEU}(F = \text{bad}) = \max_a \text{EU}(a|\text{bad}) = 53 \]