CIS 471/571 (Fall 2019): Introduction to Artificial Intelligence

Lecture 16: Bayes Nets - Sampling

Thanh H. Nguyen

Source: http://ai.berkeley.edu/home.html
Bayes' Net Representation

- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over $X$, one for each combination of parents’ values
    \[ P(X | a_1 \ldots a_n) \]
- 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:
    \[ P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i | \text{parents}(X_i)) \]
Bayes' Nets

- Representation
- Conditional Independences
  - Probabilistic Inference
    - Enumeration (exact, exponential complexity)
    - Variable elimination (exact, worst-case exponential complexity, often better)
      - Inference is NP-complete
      - Sampling (approximate)
  - Learning Bayes’ Nets from Data
Example: Alarm Network

\[ P(B \mid +j, +m) \]

<table>
<thead>
<tr>
<th>B</th>
<th>P(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+b</td>
<td>0.001</td>
</tr>
<tr>
<td>-b</td>
<td>0.999</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>P(E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+e</td>
<td>0.002</td>
</tr>
<tr>
<td>-e</td>
<td>0.998</td>
</tr>
</tbody>
</table>

\[ P(A \mid B, E) \]

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>A</th>
<th>P(A \mid B, E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+b</td>
<td>+e</td>
<td>+a</td>
<td>0.95</td>
</tr>
<tr>
<td>+b</td>
<td>+e</td>
<td>-a</td>
<td>0.05</td>
</tr>
<tr>
<td>+b</td>
<td>-e</td>
<td>+a</td>
<td>0.94</td>
</tr>
<tr>
<td>+b</td>
<td>-e</td>
<td>-a</td>
<td>0.06</td>
</tr>
<tr>
<td>-b</td>
<td>+e</td>
<td>+a</td>
<td>0.29</td>
</tr>
<tr>
<td>-b</td>
<td>+e</td>
<td>-a</td>
<td>0.71</td>
</tr>
<tr>
<td>-b</td>
<td>-e</td>
<td>+a</td>
<td>0.001</td>
</tr>
<tr>
<td>-b</td>
<td>-e</td>
<td>-a</td>
<td>0.999</td>
</tr>
</tbody>
</table>
Another Variable Elimination Example

Query: $P(X_3|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3)$

Start by inserting evidence, which gives the following initial factors:

$$p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)$$

Eliminate $X_1$, this introduces the factor $f_1(Z, y_1) = \sum_{x_1} p(x_1|Z)p(y_1|x_1)$, and we are left with:

$$p(Z)f_1(Z, y_1)p(X_2|Z)p(X_3|Z)p(y_2|X_2)p(y_3|X_3)$$

Eliminate $X_2$, this introduces the factor $f_2(Z, y_2) = \sum_{x_2} p(x_2|Z)p(y_2|x_2)$, and we are left with:

$$p(Z)f_1(Z, y_1)f_2(Z, y_2)p(X_3|Z)p(y_3|X_3)$$

Eliminate $Z$, this introduces the factor $f_3(y_1, y_2, X_3) = \sum_z p(z)f_1(z, y_1)f_2(z, y_2)p(X_3|z)$, and we are left:

$$p(y_3|X_3), f_3(y_1, y_2, X_3)$$

No hidden variables left. Join the remaining factors to get:

$$f_4(y_1, y_2, y_3, X_3) = P(y_3|X_3)f_3(y_1, y_2, X_3).$$

Normalizing over $X_3$ gives $P(X_3|y_1, y_2, y_3)$.

Computational complexity critically depends on the largest factor being generated in this process. Size of factor = number of entries in table. In example above (assuming binary) all factors generated are of size 2 --- as they all only have one variable ($Z$, $Z$, and $X_3$ respectively).
For the query $P(X_n \mid y_1, \ldots, y_n)$ work through the following two different orderings as done in previous slide: $Z, X_1, \ldots, X_{n-1}$ and $X_1, \ldots, X_{n-1}, Z$. What is the size of the maximum factor generated for each of the orderings?

Answer: $2^{n+1}$ versus $2^2$ (assuming binary)

In general: the ordering can greatly affect efficiency.
The computational and space complexity of variable elimination is determined by the largest factor.

The elimination ordering can greatly affect the size of the largest factor.
- E.g., previous slide’s example $2^n$ vs. 2

Does there always exist an ordering that only results in small factors?
- No!
Worst Case Complexity?

- **CSP:**
  \[(x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_3 \lor \neg x_4) \land (x_2 \lor \neg x_2 \lor x_4) \land (\neg x_3 \lor \neg x_4 \lor \neg x_5) \land (x_2 \lor x_5 \lor x_7) \land (x_4 \lor x_5 \lor x_6) \land (\neg x_5 \lor x_6 \lor \neg x_7) \land (\neg x_5 \lor \neg x_6 \lor x_7)\]

\[P(X_i = 0) = P(X_i = 1) = 0.5\]

\[Y_1 = X_1 \lor X_2 \lor \neg X_3\]

\[Y_8 = \neg X_5 \lor X_6 \lor X_7\]

\[Y_{1,2} = Y_1 \land Y_2\]

\[Y_{7,8} = Y_7 \land Y_8\]

\[Y_{1,2,3,4} = Y_{1,2} \land Y_{3,4}\]

\[Y_{5,6,7,8} = Y_{5,6} \land Y_{7,8}\]

\[Z = Y_{1,2,3,4} \land Y_{5,6,7,8}\]

- If we can answer $P(z)$ equal to zero or not, we answered whether the 3-SAT problem has a solution.

- Hence inference in Bayes’ nets is NP-hard. No known efficient probabilistic inference in general.
Polytrees

- A polytree is a directed graph with no undirected cycles

- For poly-trees you can always find an ordering that is efficient
  - Try it!!

- Cut-set conditioning for Bayes’ net inference
  - Choose set of variables such that if removed only a polytree remains
  - Exercise: Think about how the specifics would work out!
Bayes' Nets

- Representation
- Conditional Independences
  - Probabilistic Inference
    - Enumeration (exact, exponential complexity)
    - Variable elimination (exact, worst-case exponential complexity, often better)
    - Inference is NP-complete
      - Sampling (approximate)
  - Learning Bayes’ Nets from Data
Approximate Inference: Sampling
Sampling

- Sampling is a lot like repeated simulation
  - 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

- Sampling from given distribution
  - Step 1: Get sample $u$ from uniform distribution over $[0, 1)$
    - E.g. random() in python
  - Step 2: Convert this sample $u$ into an outcome for the given distribution
    - Each target outcome is associated with a sub-interval of $[0,1)$
    - Sub-interval size is equal to probability of the outcome.

- Example

<table>
<thead>
<tr>
<th>C</th>
<th>P(C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>0.6</td>
</tr>
<tr>
<td>green</td>
<td>0.1</td>
</tr>
<tr>
<td>blue</td>
<td>0.3</td>
</tr>
</tbody>
</table>

- If random() returns $u = 0.83$, then our sample is $C = \text{blue}$
- E.g, after sampling 8 times:
Sampling in Bayes' Nets

- Prior Sampling
- Rejection Sampling
- Likelihood Weighting
- Gibbs Sampling
Prior Sampling
Prior Sampling

\[ P(C) \]

\[ +c \quad 0.5 \\
-\quad c \quad 0.5 \]

\[ P(S|C) \]

\begin{array}{c|c|c}
+c & +s & 0.1 \\
-\quad s & 0.9 \\
-c & +s & 0.5 \\
-\quad s & 0.5 \\
\end{array}

\[ P(R|C) \]

\begin{array}{c|c|c}
+c & +r & 0.8 \\
-\quad r & 0.2 \\
-c & +r & 0.2 \\
-\quad r & 0.8 \\
\end{array}

\[ P(W|S, R) \]

\begin{array}{c|c|c|c}
+s & +r & +w & 0.99 \\
-\quad r & +w & 0.90 \\
-\quad r & -w & 0.10 \\
-s & +r & +w & 0.90 \\
-\quad r & +w & 0.01 \\
-\quad r & -w & 0.99 \\
\end{array}

Samples:

-\quad +c, -s, +r, +w
-\quad -c, +s, -r, +w
...

Prior Sampling

- For $i = 1, 2, \ldots, n$
  - Sample $x_i$ from $P(X_i \mid \text{Parents}(X_i))$
- Return $(x_1, x_2, \ldots, x_n)$
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 \( N_{PS}(x_1 \ldots x_n) \)

- Then \( \lim_{N \to \infty} \hat{P}(x_1, \ldots, x_n) = \lim_{N \to \infty} N_{PS}(x_1, \ldots, x_n) / N \)

\[ = S_{PS}(x_1, \ldots, x_n) \]

\[ = P(x_1 \ldots x_n) \]

- 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
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 $P(C \mid +s)$
  - 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)
Rejection Sampling

- Input: evidence instantiation
- For $i = 1, 2, \ldots, n$
  - Sample $x_i$ from $P(X_i \mid \text{Parents}(X_i))$
  - If $x_i$ not consistent with evidence
    - Reject: return – no sample is generated in this cycle
- Return $(x_1, x_2, \ldots, x_n)$
Likelihood Weighting
Likelihood Weighting

- Problem with rejection sampling:
  - If evidence is unlikely, rejects lots of samples
  - Evidence not exploited as you sample
  - Consider $P( \text{Shape} | \text{blue} )$

- Idea: fix evidence variables and sample the rest
  - Problem: sample distribution not consistent!
  - Solution: weight by probability of evidence given parents

<table>
<thead>
<tr>
<th>Shape</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>pyramid, green</td>
<td>pyramid, red</td>
</tr>
<tr>
<td>sphere, blue</td>
<td>cube, red</td>
</tr>
<tr>
<td>sphere, green</td>
<td>sphere, blue</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shape</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>pyramid, blue</td>
<td>pyramid, blue</td>
</tr>
<tr>
<td>sphere, blue</td>
<td>cube, blue</td>
</tr>
<tr>
<td>sphere, blue</td>
<td>sphere, blue</td>
</tr>
</tbody>
</table>
### Likelihood Weighting

#### Principals

- **Likelihood Weighting**
- **Probability of Condition** ($P(C)$)
  - $+c: 0.5$
  - $-c: 0.5$

#### Conditional Probabilities

- **$P(S|C)$**
  - $+c: +s: 0.1$, $-s: 0.9$
  - $-c: +s: 0.5$, $-s: 0.5$

- **$P(R|C)$**
  - $+c: +r: 0.8$, $-r: 0.2$
  - $-c: +r: 0.2$, $-r: 0.8$

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

#### Samples

- $+c$, $+s$, $+r$, $+w$
- $...$

#### Calculation

- $w = 1.0 \times 0.1 \times 0.99$
Likelihood Weighting

- Input: evidence instantiation
- \( w = 1.0 \)
- for \( i = 1, 2, ..., n \)
  - if \( X_i \) is an evidence variable
    - \( X_i = \) observation \( x_i \) for \( X_i \)
    - Set \( w = w \times P(x_i \mid \text{Parents}(X_i)) \)
  - else
    - Sample \( x_i \) from \( P(X_i \mid \text{Parents}(X_i)) \)
- return \((x_1, x_2, ..., x_n), w\)
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 (leads to Gibbs sampling)
Gibbs Sampling
Gibbs Sampling

- **Procedure**: keep track of a full instantiation \( x_1, x_2, \ldots, x_n \). Start with an arbitrary instantiation consistent with the evidence. Sample one variable at a time, conditioned on all the rest, but keep evidence fixed. Keep repeating this for a long time.

- **Property**: in the limit of repeating this infinitely many times the resulting samples come from the correct distribution (i.e. conditioned on evidence).

- **Rationale**: both upstream and downstream variables condition on evidence.

- In contrast: likelihood weighting only conditions on upstream evidence, and hence weights obtained in likelihood weighting can sometimes be very small. Sum of weights over all samples is indicative of how many “effective” samples were obtained, so we want high weight.
Gibbs Sampling Example: $P(S | +r)$

- **Step 1: Fix evidence**
  - $R = +r$

- **Step 2: Initialize other variables**
  - Randomly

- **Steps 3: Repeat**
  - Choose a non-evidence variable $X$
  - Resample $X$ from $P(X | \text{all other variables})$

Sample from $P(S | +c, -w, +r)$
Sample from $P(C | +s, -w, +r)$
Sample from $P(W | +s, +c, +r)$
Efficient Resampling of One Variable

- Sample from $P(S \mid +c, +r, -w)$

$$P(S\mid +c, +r, -w) = \frac{P(S, +c, +r, -w)}{P(+c, +r, -w)} = \frac{P(S, +c, +r, -w)}{\sum_s P(s, +c, +r, -w)} = \frac{P(+c)P(S\mid +c)P(+r\mid +c)P(-w\mid S, +r)}{\sum_s P(+c)P(s\mid +c)P(+r\mid +c)P(-w\mid s, +r)} = \frac{P(S\mid +r + c)\sum_s P(s\mid +c)P(-w\mid s, +r)}{P(+c)P(+r\mid +c)\sum_s P(s\mid +c)P(-w\mid s, +r)} = \frac{P(S\mid +c)P(-w\mid S, +r)}{\sum_s P(s\mid +c)P(-w\mid s, +r)}$$

- Many things cancel out – only CPTs with $S$ remain!

- More generally: only CPTs that have resampled variable need to be considered, and joined together
Bayes' Net Sampling Summary

- Prior Sampling $P(Q)$
- Likelihood Weighting $P(Q | e)$
- Rejection Sampling $P(Q | e)$
- Gibbs Sampling $P(Q | e)$
Gibbs sampling produces sample from the query distribution $P( Q \mid e )$ in limit of re-sampling infinitely often.

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.