Review: Bayes networks
Bayes network inference

- A general scenario:
  - Query variables: $X$
  - Evidence (observed) variables and their values: $E = e$
  - Unobserved variables: $Y$

- Inference problem: answer questions about the query variables given the evidence variables
  - This can be done using the posterior distribution $P(X \mid E = e)$

$$P(X \mid E = e)$$

- The posterior can be derived from the full joint $P(X, E, Y)$

- Since Bayesian networks can afford exponential savings in representing joint distributions, can they afford similar savings for inference?
Inference example

- Variables: *Cloudy, Sprinkler, Rain, WetGrass*
Inference example

• Given that the grass is wet, what is the probability that it has rained?

\[ P(r | w) \]

\[ P(r, w) = \sum_{C=c, S=s} P(c, s, r, w) \]
Another example

• What determines whether you will pass the exam?
  – **A**: Do you attend class?
  – **S**: Do you study?
  – **Pr**: Are you prepared for the exam?
  – **F**: Is the grading fair?
  – **Pa**: Do you get a passing grade on the exam?

Source: UMBC CMSC 671, Tamara Berg
Another example

\[ P(A=T) = 0.8 \]
\[ P(S=T) = 0.6 \]
\[ P(F=T) = 0.9 \]

| A | S | P(Pr|A,S) |
|---|---|-----------|
| T | T | 0.9       |
| T | F | 0.5       |
| F | T | 0.7       |
| F | F | 0.1       |

| Pr | A | F | P(Pa|A,Pr,F) |
|----|---|---|-------------|
| T  | T | T | 0.9         |
| T  | T | F | 0.6         |
| T  | F | T | 0.2         |
| T  | F | F | 0.1         |
| F  | T | T | 0.4         |
| F  | T | F | 0.2         |
| F  | F | T | 0.1         |
| F  | F | F | 0.2         |
Another example

**Query:** What is the probability that a student attended class, given that they passed the exam?

\[
P(a \mid pa) \propto P(a, pa) \\
= \sum_{S=s, F=f, Pr=pr} P(a, s, f, pr, pa) \\
= \sum_{S=s, F=f, Pr=pr} P(a)P(s)P(f)P(pr \mid a, s)P(pa \mid a, pr, f)
\]

Source: UMBC CMSC 671, Tamara Berg
Efficient inference

• Query: $P(b \mid j, m)$

\[
P(b \mid j, m)
\]

• Can we compute this sum efficiently?
Efficient inference

\[ P(b \mid j, m) \propto P(b) \sum_{E=e} P(e) \sum_{A=a} P(a \mid b, e)P(j \mid a)P(m \mid a) \]
Efficient inference

• Key idea: compute the results of sub-expressions in a bottom-up way and cache them for later use
  – Form of dynamic programming
  – Polynomial time and space complexity for polytrees: networks at most one undirected path between any two nodes
Pictorial structure model

Fischler and Elschlager(73), Felzenszwalb and Huttenlocher(00)

$$\Pr(P_{\text{tor}}, P_{\text{arm}}, \ldots | \text{Im}) \propto \prod_{i,j} \Pr(P_i | P_j) \prod \Pr(\text{Im}(P_i))$$

part geometry

part appearance
Bayesian network inference

• In full generality, NP-hard
  – More precisely, #P-hard: equivalent to counting satisfying assignments

• We can reduce satisfiability to Bayesian network inference
  – Decision problem: is P(Y = true) > 0?

\[ Y = (U_1 \lor U_2 \lor U_3) \land (\neg U_1 \lor \neg U_2 \lor U_3) \land (U_2 \lor \neg U_3 \lor U_4) \]
Bayesian network inference

• In full generality, NP-hard
  – More precisely, #P-hard: equivalent to counting satisfying assignments
• We can reduce satisfiability to Bayesian network inference
  – Decision problem: is \( P(Y = \text{true}) > 0 \)?

\[
Y = (U_1 \lor U_2 \lor U_3) \land (\neg U_1 \lor \neg U_2 \lor U_3) \land (U_2 \lor \neg U_3 \lor U_4)
\]

G. Cooper, 1990
Bayesian network inference

- How to get $P(Y = \text{true})$?
  - Need to add up the probabilities of atomic events in which $Y = \text{true}$
  - These probabilities are $1/2^n$ for atomic events corresponding to satisfying assignments (and correct intermediate values of C’s and D’s), 0 otherwise
  - Therefore, $P(Y = \text{true}) = s/2^n$, where $s$ is the number of satisfying assignments

\[
P(U_1, U_2, U_3, U_4, C_1, C_2, C_3, D_1, D_2, Y) =
P(U_1)P(U_2)P(U_3)P(U_4)
P(C_1 | U_1, U_2, U_3)P(C_2 | U_1, U_2, U_3)P(C_3 | U_2, U_3, U_4)
P(D_1 | C_1)P(D_2 | D_1, C_2)P(Y | D_2, C_3)
\]
Bayesian network inference: Big picture

- Exact inference is intractable
  - There exist techniques to speed up computations, but worst-case complexity is still exponential except in some classes of networks (polytrees)
- Approximate inference (not covered)
  - Sampling, variational methods, message passing / belief propagation...
Parameter learning

- **Inference problem**: given values of evidence variables $E = e$, answer questions about query variables $X$ using the posterior $P(X | E = e)$

- **Learning problem**: estimate the parameters of the probabilistic model $P(X | E)$ given a *training sample* $\{(x_1, e_1), \ldots, (x_n, e_n)\}$
Parameter learning

• Suppose we know the network structure (but not the parameters), and have a training set of complete observations
Parameter learning

• Suppose we know the network structure (but not the parameters), and have a training set of complete observations
  – $P(X \mid \text{Parents}(X))$ is given by the observed frequencies of the different values of $X$ for each combination of parent values
Parameter learning

- Incomplete observations

- **Expectation maximization (EM) algorithm** for dealing with missing data

<table>
<thead>
<tr>
<th>Sample</th>
<th>C</th>
<th>S</th>
<th>R</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>?</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>?</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>?</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>?</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>6</td>
<td>?</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Parameter learning

• What if the network structure is unknown?
  – *Structure learning* algorithms exist, but they are pretty complicated…

<table>
<thead>
<tr>
<th>Sample</th>
<th>C</th>
<th>S</th>
<th>R</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>6</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Summary: Bayesian networks

- Structure
- Parameters
- Inference
- Learning
Modeling the presidential election
Modeling the presidential election

- National factors
- Global polling bias
- State factors
- Poll quality/bias

- State outcome
- National poll
- State poll

How to do inference?
Modeling the presidential election

- Estimate probabilities of winning each state and marginalize out state outcomes using sampling:

[fivethirtyeighth.com](http://fivethirtyeighth.com)

[election.princeton.edu](http://election.princeton.edu)
Another look at inference

- Do we need to do sampling to find $P(E)$?

$S_i$: number of electoral votes gained in each state

$S_1$, $S_2$, ..., $S_n$
Another look at inference

$S_1, S_2, \ldots, S_n$: number of electoral votes gained in each state

$E_1, E_2, \ldots$: Electoral votes gained in states 1 through $i$ through $i$

• Obtaining $P(E_{i+1})$ from $P(E_i)$:

$$ P(E_{i+1} = e) = \sum_s P(E_i = e - s)P(S_{i+1} = s) $$
RoboVote offers two types of polls, which are tailored to different scenarios; it is up to users to indicate to RoboVote which scenario best fits the problem at hand.

**Objective Opinions**

In this scenario, some alternatives are objectively better than others, and the opinion of a participant reflects an attempt to estimate the correct order. RoboVote’s proposed outcome is guaranteed to be as close as possible — based on the available information — to the best outcome. Examples include deciding which product prototype to develop, or which company to invest in, based on a metric such as projected revenue or market share. [Try the demo.](#)

**Subjective Preferences**

In this scenario participants’ preferences reflect their subjective taste; RoboVote proposes an outcome that mathematically makes participants as happy as possible overall. Common examples include deciding which restaurant or movie to go to as a group, which destination to choose for a family vacation, or whom to elect as class president. [Try the demo.](#)