Paper reviews

- Should be useful feedback for the authors
- A critique of the paper
- No paper is perfect!
  - if you don’t understand it, state it
- Technically sound vs. convinced
- Give explicit examples, the more the better
  - cite sections, paragraphs, tables, figures, equations, etc.
- Make different sections clear
  - many conference reviews will have a similar format

Asking questions about distributions

- We want to be able to ask questions about these probability distributions
- Given $n$ variables, a query splits the variables into three sets:
  - query variable(s)
  - known/evidence variables
  - unknown/hidden variables
- $P(query \mid evidence)$
  - if we had no hidden variables, we could just multiply all the values in the different CPTs
  - to answer this, we need to sum over the hidden variables!
Two approaches

- **Enumeration**
  - top-down, multiply probabilities and sum out the hidden variables
  \[
p(FO \mid hb, lo) = \alpha p(FO) p(lo \mid FO) \sum_bp(bp) \sum_dp(do \mid FO, bp)p(hb \mid do)
\]
- **Variable elimination**
  - avoids repeated work
  - bottom-up (right to left)
  - two operations: point-wise product of factors and summing out hidden variables
  \[
p(FO \mid hb, lo) = f_1(fo)f_2(lo, fo) \sum bp(bp) \sum do(do, fo, bp)f_3(hb, do)
\]

So is VE any better than Enumeration?

- Yes and No…
  - For singly-connected networks (poly-trees), YES
  - In general, NO
    - The problem is NP-Hard

Bayesian Network Inference

- **But...**inference is still tractable in some cases.
- Special case: trees (each node has one parent)
- VE is LINEAR in this case

So, what about all those graphs with cycles?

Approximate Inference!
Approximate Inference by Stochastic Simulation

- Recall when we wanted to find out the underlying distribution (of say a coin or die) we used sampling to estimate it.

- Basic Idea:
  - Draw N samples from the distribution
  - Compute an approximate probability \( P \)
  - Eventually, for large samples sizes this converges to the true probability \( P \).
Sampling Basics: Sampling from an empty network

Cloudy

Sprinkler

Rain

Wet Grass

C P(S|C)
T .10
F .50

C P(R|C)
T .80
F .20

S R P(W|S,R)
T T .99
T F .90
F T .90
F F .01

Sample: [T, F, T, T]
Calculating probabilities

• If we do this a number of times, then we can approximate answers to queries

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>R</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

What is the probability of rain?

\[
\begin{align*}
\text{Estimated probability} & = \frac{\text{num with rain}}{\text{total samples}} \\
& = \frac{2}{10} = 0.2 \\
\end{align*}
\]

Rejection sampling

• What if we want to know the probability conditioned on some evidence?
  – \( p(\text{rain} \mid \text{wet\_grass}) \)

<table>
<thead>
<tr>
<th>C</th>
<th>S</th>
<th>R</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{Estimated probability} & = \frac{\text{num with rain and wet\_grass}}{\text{num with wet\_grass}} \\
& = \frac{1}{5} = 0.2 \\
\end{align*}
\]
Likelihood weighting

• The problem with rejection sampling is that we may have to generate a lot of samples
  – low probability/rare events
  – large networks
• Likelihood weighting
  – rather than randomly sampling over all of the variables, only randomly pick values for the query variables and hidden variables
  – for those, the evidence variables weight the examples based on the likelihood of obtaining their fixed value

Likelihood weighting: \( p(\text{rain} \mid \text{cloudy}, \text{wet_grass}) \)

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>R</th>
<th>( P(W \mid S, R) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>.99</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>.01</td>
<td></td>
</tr>
</tbody>
</table>

Likelihood weighting: \( p(\text{rain} \mid \text{cloudy}, \text{wet_grass}) \)

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>R</th>
<th>( P(W \mid S, R) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>.99</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>.01</td>
<td></td>
</tr>
</tbody>
</table>

Likelihood weighting: \( p(\text{rain} \mid \text{cloudy}, \text{wet_grass}) \)

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>R</th>
<th>( P(W \mid S, R) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>.99</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>.01</td>
<td></td>
</tr>
</tbody>
</table>

Likelihood weighting: \( p(\text{rain} \mid \text{cloudy}, \text{wet_grass}) \)

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>R</th>
<th>( P(W \mid S, R) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>.99</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>.01</td>
<td></td>
</tr>
</tbody>
</table>

Likelihood weighting: \( p(\text{rain} \mid \text{cloudy}, \text{wet_grass}) \)

<table>
<thead>
<tr>
<th></th>
<th>S</th>
<th>R</th>
<th>( P(W \mid S, R) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>.99</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>.01</td>
<td></td>
</tr>
</tbody>
</table>
Likelihood weighting: $p(\text{rain} | \text{cloudy}, \text{wet\_grass})$

**Sample: [T, F, T, T]**

**Weight:** $0.5 \times 0.9 = 0.45$

**Fixed:** weight = 0.9
Likelihood weighting: \[ p(\text{rain} \mid \text{cloudy}, \text{wet grass}) \]

\[
\begin{array}{cccc}
(C, S, R, W) & \text{weight} & \text{weight with rain, cloudy, wet grass} & \text{weight with cloudy, wet grass} \\
(T, F, T, T) & 0.45 & & \\
(T, F, T, T) & 0.45 & & \\
(T, T, F, T) & 0.005 & & \\
(T, T, T, T) & 0.495 & & \\
(T, T, T, T) & 0.495 & & \\
(T, T, T, T) & 0.45 & & \\
(T, T, T, T) & 0.45 & & \\
(T, T, T, T) & 0.45 & & \\
\end{array}
\]

Problem with likelihood weighting?

- As number of variables increased, weights will be very small
  - similar to rejection sampling, will only be a small number of higher probability ones that will actually effect the outcome
- If evidence variables are late in the ordering (BN), simulations will be not be influenced by evidence and so samples will not look much like reality

Approximate Inference using MCMC

- MCMC = Markov chain Monte Carlo
- Idea: Rather than generate individual samples, transition between "states" of the network
- Possible transition

Choose one variable and sample it given its Markov Blanket

MCMC Sampling

- Start in some valid configuration of the variables
- Repeat the following steps:
  - pick a non-evidence variable
  - randomly sample given its markov blanket
  - count this new state as a sample
- If the process visits 20 states where Rain is true and 60 states where Rain is false,
  - Then the answer to the query is \(<20/80, 60/80> = <0.25, 0.75>\)
MCMC

If you know Sprinkler=T and Wet Grass=T, there are 4 network states

Wander for awhile, average what you see

Document classification

• Naive Bayes classifier works surprisingly well for its simplicity
• We can do better!

(Big Boy models)

Revisiting the Naïve Bayes model

“Generating” a document

• The generative story of a model describes how the model would generate a sample (document)
• It can help understand the independences and how the model works
• As before, we can generate a random sample from the BN

n words in our vocabulary

How does that work for Naïve Bayes?
How would we generate a positive document?
Sampling

\[
P(\text{class}) \quad \ldots \quad P(\text{class})
\]

\[
P(w_1 | \text{class}) \quad P(w_2 | \text{class}) \quad P(w_3 | \text{class}) \quad \ldots \quad P(w_n | \text{class})
\]

document

Sampling

\[
P(\text{class}) \quad \ldots \quad P(\text{class})
\]

\[
P(w_1 | \text{class}) \quad P(w_2 | \text{class}) \quad P(w_3 | \text{class}) \quad \ldots \quad P(w_n | \text{class})
\]

document

randomly sample

the word occurs in the document

Sampling

\[
P(\text{class}) \quad \ldots \quad P(\text{class})
\]

\[
P(w_1 | \text{class}) \quad P(w_2 | \text{class}) \quad P(w_3 | \text{class}) \quad \ldots \quad P(w_n | \text{class})
\]

document

Sampling

\[
P(\text{class}) \quad \ldots \quad P(\text{class})
\]

\[
P(w_1 | \text{class}) \quad P(w_2 | \text{class}) \quad P(w_3 | \text{class}) \quad \ldots \quad P(w_n | \text{class})
\]

document

randomly sample
How are we simplifying the data?
Bag of words representation

- Notice that there is no ordering in the model
  - “I ate a banana” is viewed as the same as “ate I banana a”
- Called the “bag of words” representation

NB model

- A word either occurs or doesn’t occur
  - no frequency information
- Word occurrences are independent, given the class
  - when we sample, the only thing we condition on is the class

Incorporating frequency

- Multinomial model:
  - rather than picking whether or not a word occurs, pick what each word in the document will be
  - now rather than having boolean random variables, our random variables space is the number of words in the document

Sampling

What will the conditional probability tables look like?
Each position in the document has a distribution over all words and each class.

In practice, we use the same distribution for all word positions!
The random pick a word for the second position

**Multinomial model**

- Called a multinomial model because the word frequencies drawn for a document of length $m$, follow a multinomial distribution
  - sampling with replacement from a fixed distribution
- Word occurrences are still independent!
  - doesn’t matter what other words we’ve drawn
- Although technically the position is specified, doesn’t really give us positional information
- Still a naïve Bayes model!

**Boolean NB vs. Multinomial NB**

Boolean NB vs. Multinomial NB

Industry Sector data (71 classes, web pages)


Plate notation

- It can be tedious to write out all of the children in a BN
- When they’re all the same type, we can use “plate” notation
  - A plate represents a set of variables
  - We specify repetition by putting a number in the lower right corner
  - Edges crossing plate boundaries are considered to be multiple edges

Dirichlet Compound Multinomial (DCM)

- To generate a document
  - Pick a class
  - Based on that class, draw a multinomial representing a topic
    - \( p(\text{topic} | \text{class}) \) is represented by a Dirichlet distribution
    - Gives us a distribution over multinomials
  - Based on this multinomial, sample as before
Key problem with NB multinomial: words tend to be “bursty”
- if a word occurs once, it’s likely to occur again
- particularly content words, e.g. Bush
DCM model allows us to model burstiness by picking multinomials for a given document that have a higher probability of occurring

For those that like math 😊

\[
p(x | \alpha) = \frac{\Gamma\left(\sum_{w=1}^{W} \alpha_w \right) \prod_{w=1}^{W} \theta_{w}^{\alpha_w-1}}{\prod_{w=1}^{W} \Gamma(\alpha_w)}
\]

\[
= \frac{\Gamma\left(\sum_{w=1}^{W} \alpha_w + x_w \right) \prod_{w=1}^{W} \theta_{w}^{\alpha_w+x_w-1}}{\prod_{w=1}^{W} \Gamma(\alpha_w + x_w)}
\]

DCM vs. Multinomial

<table>
<thead>
<tr>
<th></th>
<th>Industry</th>
<th>20 Newsgroups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multinomial</td>
<td>0.600</td>
<td>0.853</td>
</tr>
<tr>
<td>DCM</td>
<td>0.806</td>
<td>0.890</td>
</tr>
</tbody>
</table>

http://www.cs.pomona.edu/~dkauchak/papers/kauchak05modeling.pdf

Topic models

- Often a document isn’t just about one idea/topic
- Topic models view documents as a blend of “topics”

topic 1 +
topic 2 +
topic 3
Topic models

How might we model this as a Bayes net?

DCM model

LDA model (Latent Dirichlet Allocation)

LDA

To generate a document
- for each word in the document:
  - pick a topic given the class
  - pick a word given the topic (and the prior)

Key paper:
LDA

- Two binary tasks from industry sector
- Used LDA to extract features
- Then used SVMs (support vector machines) for classification

Document classification

- "Generative models"
  - represent underlying probability distribution
  - can be used for classification, but also other tasks
  - models:
    - Bernouli (boolean) naïve Bayes
    - Multinomial naïve Bayes
    - Dirichlet Compound Multinomial
    - Latent Dirichlet Allocation
- Discriminative models
  - support vector machines
  - markov random fields very good for classification only

Midterm

- Open book
  - still only 75 min, so don’t rely on it too much
- Anything we’ve talked about in class or read about is fair game
- Written questions are a good place to start

Review

- Intro to AI
  - what is AI
  - goals
  - challenges
  - problem areas
Review

- Uninformed search
  - reasoning through search
  - agent paradigm (sensors, actuators, environment, etc.)
  - setting up problems as search
    - state space (starting state, next state function, goal state)
    - actions
    - costs
  - problem characteristics
    - observability
    - determinism
    - known/unknown state space
  - techniques
    - BFS
    - DFS
    - uniform cost search
    - depth limited search
    - iterative deepening

- Informed search
  - heuristic function
    - admissibility
    - combining functions
    - dominance
  - methods
    - greedy best-first search
    - $A^*$

- Adversarial search
  - game playing through search
    - ply
    - depth
    - branching factor
    - state space sizes
    - optimal play
  - game characteristics
    - observability
    - # of players
    - discrete vs. continuous
    - real-time vs. turn-based
    - determinism

- Adversarial search cont
  - minimax algorithm
    - alpha-beta pruning
      - optimality, etc.
    - evaluation functions (heuristics)
      - horizon effect
    - improvements
      - transposition table
      - history/end-game tables
  - dealing with chance/non-determinism
    - expected minimax
  - dealing with partially observable games
Review

- Local search
  - when to use/what types of problems
  - general formulation
  - hill-climbing
    - greedy
    - random restarts
    - randomness
    - simulated annealing
    - local beam search
    - taboo list
  - genetic algorithms

- CSPs
  - problem formulation
    - variables
    - domain
    - constraints
  - why CSPs? applications?
    - constraint graph
  - CSP as search
    - backtracking algorithm
    - forward checking
    - arc consistency
  - heuristics
    - most constrained variable
    - least constrained value
    - ...

Review

- Basic probability
  - why probability (vs. say logic)?
  - vocabulary
    - experiment
    - sample
    - event
    - random variable
    - probability distribution
  - unconditional/prior probability
  - joint distribution
  - conditional probability
  - Bayes rule
  - estimating probabilities

Review

- Bayes nets
  - representation
  - dependencies/independencies
    - d-separation
    - Markov blanket
  - reasoning/querying
    - exact:
      - enumeration
      - variable elimination
    - sampling
      - basic
      - variable elimination
      - MCMC
Review

- Bayesian classification
  - problem formulation, argmax, etc.
  - NB model
  - Other models
    - multinomial, DCM, LDA
  - training, testing, evaluation
  - plate notation