Learning Bayesian Networks

CSE 473
Last Time

- Basic notions
- Bayesian networks
- **Statistical learning**
  - Parameter learning (MAP, ML, Bayesian L)
  - Naïve Bayes
    - Issues
  - Structure Learning
  - Expectation Maximization (EM)
- Dynamic Bayesian networks (DBNs)
- Markov decision processes (MDPs)
Simple Spam Model

Naïve Bayes assumes attributes independent
Given class of parent
Correct?
Naïve Bayes
Incorrect, but Easy!

• \( P(\text{spam} \mid X_1 \ldots X_n) = \prod_i P(\text{spam} \mid X_i) \)

• How compute \( P(\text{spam} \mid X_i) \)?

• How compute \( P(X_i \mid \text{spam}) \)?
\[ P(X_i \mid S) = \frac{\# \text{SPAM} + \# \text{Envelope} + m}{\# + \# + m} \]
• \( P(\text{spam} \mid X_1 \ldots X_n) = \prod_i P(\text{spam} \mid X_i) \)

Is there any potential problem here?

• We are multiplying lots of small numbers
  Danger of underflow!
  \( 0.5^{57} = 7 \times 10^{-18} \)

• Solution? Use logs and add!
  \( p_1 \times p_2 = e^{\log(p_1)+\log(p_2)} \)
  Always keep in log form
$P(S \mid X)$

- Easy to compute from data if $X$ discrete

<table>
<thead>
<tr>
<th>Instance</th>
<th>$X$</th>
<th>Spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>4</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

- $P(S \mid X) = \frac{1}{4}$ \textit{ignoring smoothing...}
P(S | X)

• What if X is real valued?

<table>
<thead>
<tr>
<th>Instance</th>
<th>X</th>
<th>Spam?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.01</td>
<td>&lt;T</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>&lt;T</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>&lt;T</td>
</tr>
<tr>
<td>4</td>
<td>0.03</td>
<td>&gt;T</td>
</tr>
<tr>
<td>5</td>
<td>0.05</td>
<td>&gt;T</td>
</tr>
</tbody>
</table>

• What now?
Anything Else?

P(S|0.04)?

<table>
<thead>
<tr>
<th>#</th>
<th>X</th>
<th>S?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>0.03</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>0.05</td>
<td>T</td>
</tr>
</tbody>
</table>
Smooth with Gaussian then sum

"Kernel Density Estimation"

<table>
<thead>
<tr>
<th>#</th>
<th>X</th>
<th>S?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>0.01</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>0.03</td>
<td>T</td>
</tr>
<tr>
<td>5</td>
<td>0.05</td>
<td>T</td>
</tr>
</tbody>
</table>
Spam?

What's with the shape?

\[
P(S|X=0.023) \quad \frac{P(S|X=0.023)}{P(\neg S|X=0.023)}
\]

© Daniel S. Weld
Analysis
Recap

• Given a BN structure (with discrete or continuous variables), we can learn the parameters of the conditional prop tables.
What if we *don’t* know structure?
Outline

• Statistical learning
  Parameter learning (MAP, ML, Bayesian L)
  Naïve Bayes
    • Issues
  Structure Learning
  Expectation Maximization (EM)
• Dynamic Bayesian networks (DBNs)
• Markov decision processes (MDPs)
• Search thru the space of possible network structures!
  (for now, assume we observe all variables)
• For each structure, learn parameters
• Pick the one that fits observed data best
  Caveat - won’t we end up fully connected?

????
• When scoring, add a penalty
  ∝ model complexity
• Search thru the space
• For each structure, learn parameters
• Pick the one that fits observed data best

• Problem?
  Exponential number of networks!
  And we need to learn parameters for each!
  Exhaustive search out of the question!

• So what now?
• Uniform prior over random networks?

• Network which reflects expert knowledge?
Learning BN Structure

prior network + equivalent sample size

data

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

improved network(s)
• We described how to do MAP (and ML) learning of a Bayes net (including structure)

• How would Bayesian learning (of BNs) differ?
Outline

• Statistical learning
  Parameter learning (MAP, ML, Bayesian L)
  Naïve Bayes
  • Issues
  Structure Learning
  Expectation Maximization (EM)

• Dynamic Bayesian networks (DBNs)
• Markov decision processes (MDPs)
Hidden Variables

- But we can’t observe the disease variable
- Can’t we learn without it?
We *could*-

- But we’d get a fully-connected network

- With 708 parameters (vs. 78)
  Much harder to learn!
Chicken & Egg Problem

• If we knew that a training instance (patient) had the disease...
  It would be easy to learn $P(\text{symptom} \mid \text{disease})$
  But we can’t observe disease, so we don’t.

• If we knew params, e.g. $P(\text{symptom} \mid \text{disease})$
  then it’d be easy to estimate if the patient had the disease.
  But we don’t know these parameters.
Expectation Maximization (EM)  
(high-level version)

- Pretend we **do** know the parameters
  Initialize randomly
- **[E step]** Compute probability of instance having each possible value of the hidden variable
- **[M step]** Treating each instance as fractionally having *both* values compute the new parameter values
- Iterate until convergence!
Candy Problem

• Given a bowl of mixed candies
• Can we learn contents of each bag, and
• What percentage of each is in bowl?
Naïve Candy

- Flavor: cherry / lime
- Wrapper: red / green
- Hole: 0/1
Simpler Problem

- Mixture of two distributions

Know: form of distr, variance, % = 5, 9
- Just need mean of each
$U_{ml} = \text{argmin}_u \sum_i (x_i - u)^2$
• Randomly
• Initialize: $\theta$
  $\theta_{F1}, \theta_{W1}, \theta_{H1},$
  $\theta_{F2}, \theta_{W2}, \theta_{H2}$
• [E Step]
  In observable case, we would
  Estimate $\theta$ from observed counts
  Of candies in bags 1 and 2
  But bag is hidden, so calculate expected amounts instead
  $\theta^1 = \sum_j P(\text{bag}=1 \mid \text{flavor}_j, \text{wrapper}_j, \text{holes}_j) / N$
  Etc. for other parameters
• [M Step]
  Recompute
• Iterate
Generate Data

- Assume (true) model is $\theta = 0.5$, $\theta_{F_1} = \theta_{W_1} = \theta_{H_1} = 0.8$, $\theta_{F_2} = \theta_{W_2} = \theta_{H_2} = 0.3$

<table>
<thead>
<tr>
<th></th>
<th>W=red</th>
<th></th>
<th>W=green</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>H=1</td>
<td>H=0</td>
<td>H=1</td>
</tr>
<tr>
<td>F=cherry</td>
<td>273</td>
<td>93</td>
<td>104</td>
</tr>
<tr>
<td>F=lime</td>
<td>79</td>
<td>100</td>
<td>94</td>
</tr>
</tbody>
</table>