CSC242: Intro to AI

Lecture 22
Administrivia

Posters! Tue Apr 24 and Thu Apr 26

Idea!

Presentation!

2-wide x 8-high landscape pages
Learning Probabilistic Models
Supervised Learning

• Given a training set of $N$ example input-output pairs:
  $$(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$$
  where each $y_j = f(x_j)$

• Discover function $h$ that approximates $f$
Linear Classifiers

- Linear regression for fitting a line to data
- Linear classifiers: use a line to separate the data
- Gradient descent for finding weights
- Hard threshold (perceptron learning rule)
- Logistic (sigmoid) threshold
- Neural Networks: Network of linear classifiers
- Support Vector Machines: State of the art for learning supervised learning of classifiers
Learning Probabilistic Models
$h_1$ 100% cherry
$h_2$ 75% cherry 25% lime
$h_3$ 50% cherry 50% lime
$h_4$ 25% cherry 75% lime
$h_5$ 100% lime
$H \in \{h_1, h_2, h_3, h_4, h_5\}$

Observations: \hspace{1cm} D_1 = \bullet \hspace{1cm} D_2 = \bullet \hspace{1cm} D_3 = \bullet \hspace{1cm} \ldots

Goal: Predict the flavor of the next candy
| Bags | Agent, process, disease, ...
| Candies | Actions, effects, symptoms, results of tests, ...
| $D_1=\bigcirc$ |  
| $D_2=\bigcirc$ | Observations  
| $D_3=\bullet$ |  
| Goal | Predict next candy | Predict agent’s next move 
| | Predict next output of process | Predict disease given symptoms and tests |
\[ H \in \{h_1, h_2, h_3, h_4, h_5\} \]

Observations: \( D_1 = \text{●} \quad D_2 = \text{●} \quad D_3 = \text{●} \quad \ldots \)

Goal: Predict the flavor of the next candy
Strategy 1

• Predict (estimate) the underlying distribution $h_i$

• Use that to predict the next observation
Bayesian Learning

• Compute the probability of each hypothesis distribution

• Use that to compute a weighted estimate of the possible values for the next observation
Bayesian Learning

\[ P(h_i \mid d) = \alpha P(d \mid h_i)P(h_i) \]

Likelihood of the data under the hypothesis

Hypothesis prior
Bayesian Learning

Likelihood of disease given symptoms/tests

\[ P(h_i \mid d) = \alpha P(d \mid h_i)P(h_i) \]

Likelihood that the disease caused the symptoms/tests

Prior probability of the disease
Bayesian Learning

\[ P(h_i \mid d) = \alpha P(d \mid h_i) P(h_i) \]

Likelihood of the data under the hypothesis

Hypothesis prior
$P(H) = \langle 0.1, 0.2, 0.4, 0.2, 0.1 \rangle$
\[ \mathbf{P}(H) = \langle 0.1, 0.2, 0.4, 0.2, 0.1 \rangle \]

\[ P(d \mid h_i) = \prod_j P(d_j \mid h_i) \quad \text{if i.i.d.} \]
Independent Identically Distributed (i.i.d.)

• Probability of a sample is independent of any previous samples

\[ P(D_i|D_{i-1}, D_{i-2}, \ldots) = P(D_i) \]

• Probability distribution doesn’t change among samples

\[ P(D_i) = P(D_{i-1}) = P(D_{i-2}) = \ldots \]
\[ P(H) = \langle 0.1, 0.2, 0.4, 0.2, 0.1 \rangle \]

\[ P(d \mid h_i) = \prod_j P(d_j \mid h_i) \quad \text{if i.i.d.} \]
\[ P(d \mid h_i) \]

- \( h_1 \): 0
- \( h_2 \): 0.25^{10}
- \( h_3 \): 0.5^{10}
- \( h_4 \): 0.75^{10}
- \( h_5 \): 1
\[ P(h_i \mid d) = \alpha P(d \mid h_i)P(h_i) \]
\[ P(H) = \langle 0.1, 0.2, 0.4, 0.2, 0.1 \rangle \]
Bayesian Prediction

\[ P(D_{N+1} \mid d) = \sum_i P(D_{N+1} \mid d, h_i) P(h_i \mid d) \]

\[ = \sum_i P(D_{N+1} \mid h_i) P(h_i \mid d) \]

\[ = \alpha \sum_i P(D_{N+1} \mid h_i) P(d \mid h_i) P(h_i) \]
\[ P(d_{N+1} = \text{lime} \mid d_1, \ldots, d_N) \]
Bayesian Learning

\[ P(X \mid d) = \alpha \sum_i P(X \mid h_i)P(d \mid h_i)P(h_i) \]
Maximum A Posteriori (MAP)

$$h_{MAP} = \arg\max_{h_i} P(h_i \mid d)$$

$$P(X \mid d) \approx P(X \mid h_{MAP})$$
\[ P(d_{N+1} = \text{lime} \mid d_1, \ldots, d_N) \]
Maximum A Posteriori (MAP)

\[
h_{MAP} = \arg\max_{h_i} P(h_i \mid d)
\]

\[
P(X \mid d) \approx P(X \mid h_{MAP})
\]
What About Overfitting?

- Expressive hypothesis space allows many hypotheses that fit the data well
- Solution: Use hypothesis prior to penalize complexity
- Usually more complex hypotheses have a lower prior probability than simple ones
Maximum Likelihood Hypothesis

- Assume uniform hypothesis prior
- No hypothesis preferred to any other \( a \) priori (e.g., all equally complex)

\[
h_{MAP} = \arg\max_{h_i} P(h_i | d)
\]

\[
= \arg\max_{h_i} P(d | h_i) = h_{ML}
\]
Statistical Learning

- Bayesian Learning
- Hypothesis prior
- Likelihood of data given hypothesis
- Weighted average over all hypotheses
- MAP hypothesis: single best hypothesis
- ML hypothesis: uniform hypothesis prior
$\mathbf{P}(H) = \langle 0.1, 0.2, 0.4, 0.2, 0.1 \rangle$
<table>
<thead>
<tr>
<th>Bags</th>
<th>Agent, process, disease, ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candies</td>
<td>Actions, effects, symptoms, results of tests, ...</td>
</tr>
<tr>
<td>( D_1 = )</td>
<td>( \bullet )</td>
</tr>
<tr>
<td>( D_2 = )</td>
<td>( \bullet )</td>
</tr>
<tr>
<td>( D_3 = )</td>
<td>( \bullet )</td>
</tr>
<tr>
<td>Observations</td>
<td>Predict next output of process</td>
</tr>
<tr>
<td>Goal</td>
<td>Predict agent’s next move</td>
</tr>
<tr>
<td></td>
<td>Predict next candy</td>
</tr>
<tr>
<td></td>
<td>Predict disease given symptoms and tests</td>
</tr>
</tbody>
</table>
Bayesian Networks

- A Bayesian network represents a full joint probability distribution between a set of random variables
- Uses conditional independence to reduce the number of probabilities need to specify it and make inference easier
Learning and Bayesian Networks

- The distribution defined by the network is parameterized by the entries in the CPTs associated with the nodes
- A BN defines a space of distributions corresponding to the parameter space
Learning and Bayesian Networks

• If we have a BN that we believe represents the causality (conditional independence) in our problem

• In order to find (estimate) the true distribution

• We learn to find the parameters of the model from the training data
Alarm

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>P(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>t</td>
<td>.95</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>.94</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>.29</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>.001</td>
</tr>
</tbody>
</table>

JohnCalls

<table>
<thead>
<tr>
<th>A</th>
<th>P(J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>.90</td>
</tr>
<tr>
<td>f</td>
<td>.05</td>
</tr>
</tbody>
</table>

MaryCalls

<table>
<thead>
<tr>
<th>A</th>
<th>P(M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>.70</td>
</tr>
<tr>
<td>f</td>
<td>.01</td>
</tr>
</tbody>
</table>

Burglary

P(B) = .001

Earthquake

P(E) = .002
Parameter Learning (in Bayesian Networks)
Flavor

$h_\Theta$

$P(F=\text{cherry})$

$\Theta$

$\text{Flavor}$
\[ P(F=\text{cherry}) \]

\[ h_\Theta \]

\[ P(d \mid h_\Theta) = \prod_j P(d_j \mid h_\Theta) \]

\[ = \Theta^c \cdot (1 - \Theta)^l \]
Maximum Likelihood Hypothesis

\[ \arg\max_{\Theta} P(d \mid h_\Theta) \]
Log Likelihood

\[ P(d \mid h_\Theta) = \prod_j P(d_j \mid h_\Theta) = \Theta^c \cdot (1 - \Theta)^l \]

\[ L(d \mid h_\Theta) = \log P(d \mid h_\Theta) = \sum_j \log P(d_j \mid h_\Theta) = c \log \Theta + l \log(1 - \Theta) \]
Maximum Likelihood Hypothesis

\[ L(d \mid h_\Theta) = c \log \Theta + l \log(1 - \Theta) \]

\[ \operatorname{argmax}_\Theta L(d \mid h_\Theta) = \frac{c}{c + l} = \frac{c}{N} \]
Flavor

\[ P(F = \text{cherry}) \]

\[ \Theta \]

Wrapper

\[
\begin{array}{|c|c|}
\hline
F & P(W = \text{red} | F) \\
\hline
\text{cherry} & \Theta_1 \\
\text{lime} & \Theta_2 \\
\hline
\end{array}
\]
Flavor

\[ P(F = \text{cherry}) \]

\[ \Theta \]

\[ h_{\Theta, \Theta_1, \Theta_2} \]

**Wrapper**

| \( F \)    | \( P(W = \text{red}|F) \) |
|------------|-----------------------------|
| cherry     | \( \Theta_1 \)              |
| lime       | \( \Theta_2 \)              |
\[ P(F = f, W = w \mid h_{\Theta, \Theta_1, \Theta_2}) = \]
\[ P(F = f \mid h_{\Theta, \Theta_1, \Theta_2}) \cdot P(W = w \mid W = f, h_{\Theta, \Theta_1, \Theta_2}) \]
\[ P(F = c, W = g \mid h_{\Theta, \Theta_1, \Theta_2}) = \Theta \cdot (1 - \Theta_1) \]
<table>
<thead>
<tr>
<th>$F$</th>
<th>$W$</th>
<th>( P(F=f, W=w \mid h_{\Theta, \Theta_1, \Theta_2}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>cherry</td>
<td>red</td>
<td>( \Theta \Theta_1 )</td>
</tr>
<tr>
<td>cherry</td>
<td>green</td>
<td>( \Theta (1-\Theta_1) )</td>
</tr>
<tr>
<td>lime</td>
<td>red</td>
<td>( (1-\Theta) \Theta_2 )</td>
</tr>
<tr>
<td>lime</td>
<td>green</td>
<td>( (1-\Theta) (1-\Theta_2) )</td>
</tr>
<tr>
<td>$F$</td>
<td>$W$</td>
<td>$P$</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>--------------------</td>
</tr>
<tr>
<td>cherry</td>
<td>red</td>
<td>$\Theta \Theta_1$</td>
</tr>
<tr>
<td>cherry</td>
<td>green</td>
<td>$\Theta (1-\Theta_1)$</td>
</tr>
<tr>
<td>lime</td>
<td>red</td>
<td>$(1-\Theta) \Theta_2$</td>
</tr>
<tr>
<td>lime</td>
<td>green</td>
<td>$(1-\Theta) (1-\Theta_2)$</td>
</tr>
<tr>
<td>$F$</td>
<td>$W$</td>
<td>$P$</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>cherry</td>
<td>red</td>
<td>$\Theta \Theta_1$</td>
</tr>
<tr>
<td>cherry</td>
<td>green</td>
<td>$\Theta \ (1-\Theta_1)$</td>
</tr>
<tr>
<td>lime</td>
<td>red</td>
<td>$(1-\Theta) \Theta_2$</td>
</tr>
<tr>
<td>lime</td>
<td>green</td>
<td>$(1-\Theta) \ (1-\Theta_2)$</td>
</tr>
</tbody>
</table>
\[ P(\mathbf{d} \mid h_{\Theta, \Theta_1, \Theta_2}) = \]
\[(\Theta \Theta_1)^{r_c} \cdot (\Theta (1 - \Theta_1))^{g_c} \cdot ((1 - \Theta) \Theta_2)^{r_l} \cdot ((1 - \Theta)(1 - \Theta_2))^{g_l} \]
\[= \Theta^c (1 - \Theta)^l \cdot \Theta_1^{r_c} (1 - \Theta_1)^{g_c} \cdot \Theta_2^{r_l} (1 - \Theta_2)^{g_l} \]

\[ L(\mathbf{d} \mid h_{\Theta, \Theta_1, \Theta_2}) = c \log \Theta + l \log (1 - \Theta) + \]
\[ [r_c \log \Theta_1 + g_c \log (1 - \Theta_1)] + \]
\[ [r_l \log \Theta_2 + g_l \log (1 - \Theta_2)] \]
\[\Theta = \frac{c}{c + l} = \frac{c}{N}\]

\[\Theta_1 = \frac{r_c}{r_c + g_c} = \frac{r_c}{c}\]

\[\Theta_2 = \frac{r_l}{r_l + g_l} = \frac{r_l}{l}\]
Flavor

\[ h_{\Theta, \Theta_1, \Theta_2} \]

\[ P(F=\text{cherry}) \]

\[ \Theta \]

\[ \text{Flavor} \]

\[ \text{Wrapper} \]

\[ \begin{array}{|c|c|}
| F    | P(W=\text{red}|F) |
|-------|-------------------|
| cherry| \Theta_1          |
| lime  | \Theta_2          |
\end{array} \]
\[ \Theta = \frac{c}{c + l} = \frac{c}{N} \]

\[ \Theta_1 = \frac{r_c}{r_c + g_c} = \frac{r_c}{c} \]

\[ \Theta_2 = \frac{r_l}{r_l + g_l} = \frac{r_l}{l} \]

\[
\text{argmax } L(d \mid h_{\Theta, \Theta_1, \Theta_2}) = \text{argmax } P(d \mid h_{\Theta, \Theta_1, \Theta_2})
\]
Naive Bayes Models

Class

Attr_1  Attr_2  Attr_3  ...

Naive Bayes Models

\{ \textit{mammal, reptile, fish, ...} \}

\begin{itemize}
  \item Class
  \item Furry
  \item Warm Blooded
  \item Size
\end{itemize}
Naive Bayes Models
Naive Bayes Models

\{ mammal, reptile, fish, \ldots \}

\[
\text{Class} \quad \text{Furry} \quad \text{Warm Blooded} \quad \text{Size} \quad \ldots
\]
Naive Bayes Models

\{ terrorist, tourist \}

Class

Arrival Mode

One-way Ticket

Furtive Manner

...
Naive Bayes Models

Disease

Test\textsubscript{1}  Test\textsubscript{2}  Test\textsubscript{3}  ...
Learning Naive Bayes Models

- Naive Bayes model with $n$ Boolean attributes requires $2^n+1$ parameters.
- Maximum likelihood hypothesis $h_{ML}$ can be found with no search.
- Scales to large problems.
- Robust to noisy or missing data.
Learning with Complete Data

- Can learn the CPTs for a Bayes Net from observations that include values for all variables
- Finding maximum likelihood parameters decomposes into separate problems, one for each parameter
- Parameter values for a variable given its parents are the observed frequencies
• 20.2.3: Maximum likelihood parameter learning: Continuous models
• 20.2.4: Bayesian parameter learning
• 20.2.5: Learning Bayes net structure
• 20.2.6: Density estimation with nonparametric models
\{ \text{terrorist, tourist} \}

\text{Class}

\text{Arrival Mode} \quad \text{One-way Ticket} \quad \text{Furtive Manner} \quad \text{...}
<table>
<thead>
<tr>
<th>Arrival</th>
<th>One-Way</th>
<th>Furtive</th>
<th>...</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>taxi</td>
<td>yes</td>
<td>very</td>
<td>...</td>
<td>terrorist</td>
</tr>
<tr>
<td>car</td>
<td>no</td>
<td>none</td>
<td>...</td>
<td>tourist</td>
</tr>
<tr>
<td>car</td>
<td>yes</td>
<td>very</td>
<td>...</td>
<td>terrorist</td>
</tr>
<tr>
<td>car</td>
<td>yes</td>
<td>some</td>
<td>...</td>
<td>tourist</td>
</tr>
<tr>
<td>walk</td>
<td>yes</td>
<td>none</td>
<td>...</td>
<td>student</td>
</tr>
<tr>
<td>bus</td>
<td>no</td>
<td>some</td>
<td>...</td>
<td>tourist</td>
</tr>
</tbody>
</table>
Disease

Test$_1$  Test$_2$  Test$_3$  ...

<table>
<thead>
<tr>
<th>Test</th>
<th>Test2</th>
<th>Test3</th>
<th>...</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>...</td>
<td>?</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>...</td>
<td>?</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>...</td>
<td>?</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>...</td>
<td>?</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>...</td>
<td>?</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>...</td>
<td>?</td>
</tr>
</tbody>
</table>
78 parameters

708 parameters
Hidden (Latent) Variables

• Can dramatically reduce the number of parameters required to specify a Bayes net

• Reduces amount of data required to learn the parameters

• Values of hidden variables not present in training data (observations)

• “Complicates” the learning problem
EM
Expectation-Maximization

• Repeat

• Expectation: “Pretend” we know the parameters and compute (or estimate) likelihood of data given model

• Maximization: Recompute parameters using expected values as if they were observed values

• Until convergence
<table>
<thead>
<tr>
<th>Flavor</th>
<th>cherry</th>
<th>lime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrapper</td>
<td>red</td>
<td>green</td>
</tr>
<tr>
<td>Hole</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>
\[ P(F, W, H) \]

<table>
<thead>
<tr>
<th>Flavor</th>
<th>Wrapper</th>
<th>Hole</th>
<th>( P(f, w, h) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>cherry</td>
<td>red</td>
<td>t</td>
<td>( p_{c,r,t} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>f</td>
<td>( p_{c,r,f} )</td>
</tr>
<tr>
<td>green</td>
<td>t</td>
<td>( p_{c,g,t} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>f</td>
<td>( p_{c,g,f} )</td>
<td></td>
</tr>
<tr>
<td>lime</td>
<td>red</td>
<td>t</td>
<td>( p_{l,r,t} )</td>
</tr>
<tr>
<td></td>
<td>f</td>
<td>( p_{l,r,f} )</td>
<td></td>
</tr>
<tr>
<td>green</td>
<td>t</td>
<td>( p_{l,g,t} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>f</td>
<td>( p_{l,g,f} )</td>
<td></td>
</tr>
</tbody>
</table>
$P_1(F, W, H)$

$P_2(F, W, H)$

\[ P_1(X_1, X_2, X_3) \] \[ P_2(X_1, X_2, X_3) \]
**\(P(F,W,H)\)**

<table>
<thead>
<tr>
<th>Flavor</th>
<th>Wrapper</th>
<th>Hole</th>
<th>(P(f,w,h))</th>
</tr>
</thead>
<tbody>
<tr>
<td>cherry</td>
<td>red</td>
<td>(t)</td>
<td>(p_{c,r,t})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(f)</td>
<td>(p_{c,r,f})</td>
</tr>
<tr>
<td>green</td>
<td></td>
<td>(t)</td>
<td>(p_{c,g,t})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(f)</td>
<td>(p_{c,g,f})</td>
</tr>
<tr>
<td>lime</td>
<td>red</td>
<td>(t)</td>
<td>(p_{l,r,t})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(f)</td>
<td>(p_{l,r,f})</td>
</tr>
<tr>
<td></td>
<td>green</td>
<td>(t)</td>
<td>(p_{l,g,t})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(f)</td>
<td>(p_{l,g,f})</td>
</tr>
</tbody>
</table>
Bag

- Flavor
- Wrapper
- Hole

Bag = 1

Flavor = cherry | B

θ

θ
\begin{align*}
    Bag & \rightarrow Flavor & P(F=cherry|B) & \Theta_{F1} \\
    Bag & \rightarrow Wrapper & P(W=red|B) & \Theta_{W1} \\
    Bag & \rightarrow Hole & P(H=true|B) & \Theta_{H1}
\end{align*}

\begin{align*}
    Bag & \rightarrow Flavor & P(F=cherry|B) & \Theta_{F2} \\
    Bag & \rightarrow Wrapper & P(W=red|B) & \Theta_{W2} \\
    Bag & \rightarrow Hole & P(H=true|B) & \Theta_{H2}
\end{align*}
\( P(Bag=1) \)

\[ \theta \]

<table>
<thead>
<tr>
<th>Bag</th>
<th>( P(F=\text{cherry} \mid B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \theta_{F_1} )</td>
</tr>
<tr>
<td>2</td>
<td>( \theta_{F_2} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bag</th>
<th>( P(W=\text{red} \mid B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \Theta_{W_1} )</td>
</tr>
<tr>
<td>2</td>
<td>( \Theta_{W_2} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bag</th>
<th>( P(H=\text{true} \mid B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \Theta_{H_1} )</td>
</tr>
<tr>
<td>2</td>
<td>( \Theta_{H_2} )</td>
</tr>
<tr>
<td>Flavor</td>
<td>Wrapper</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>cherry</td>
<td>red</td>
</tr>
<tr>
<td>cherry</td>
<td>red</td>
</tr>
<tr>
<td>lime</td>
<td>green</td>
</tr>
<tr>
<td>cherry</td>
<td>green</td>
</tr>
<tr>
<td>lime</td>
<td>green</td>
</tr>
<tr>
<td>cherry</td>
<td>red</td>
</tr>
<tr>
<td>lime</td>
<td>red</td>
</tr>
<tr>
<td>$N=1000$</td>
<td>$W=\text{red}$</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
</tr>
<tr>
<td></td>
<td>$H=1$</td>
</tr>
<tr>
<td>$F=\text{cherry}$</td>
<td>273</td>
</tr>
<tr>
<td>$F=\text{lime}$</td>
<td>79</td>
</tr>
</tbody>
</table>
EM
Expectation-Maximization

• Repeat

• E: Use the current values of the parameters to compute the expected values of the hidden variables

• M: Recompute the parameters to maximize the log-likelihood of the data given the values of the variables (observed and hidden)
EM
Expectation-Maximization

• Repeat

• E: Use the current values of the parameters to compute the expected values of the hidden variables

• M: Recompute the parameters to maximize the log-likelihood of the data given the values of the variables (observed and hidden)
Summary

• Statistical Learning
• Bayesian Learning
• Maximum A Posteriori (MAP) hypothesis
• Maximum Likelihood (ML) hypothesis
• Learning the parameters of a Bayesian Network
• Complete data: Maximum Likelihood learning
• Hidden variables: EM
For Next Time:
21.0–21.3; 21.5 fyi
Posters!