CSC242: Intro to AI

Lecture 23
Learning: The Big Picture So Far

- Function Learning
- Linear Regression
- Linear Classifiers
- Neural Networks
Learning Probabilistic Models
Bags: Agent, process, disease, ...

Candies: Actions, effects, symptoms, results of tests, ...

Observations:
- $D_1 = \bullet$
- $D_2 = \bullet$
- $D_3 = \bullet$

Goal:
- Predict next candy
- Predict agent’s next move
- Predict next output of process
- Predict disease given symptoms and tests
Learning and Bayesian Networks
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 need to learn the parameters of the model from the training data
Flavor

$h_\Theta$

$P(F=cherry)$

$\Theta$
\begin{align*}
P(F=\text{cherry}) \quad \Theta \\
\end{align*}
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
  \[ P(D_i) = P(D_{i-1}) = P(D_{i-2}) = \cdots \]
Flavor

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

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

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

\[ P(d \mid h_{\Theta}) \]
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

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

\[ \arg \max_{\Theta} L(d \mid h_{\Theta}) = \frac{c}{c + l} = \frac{c}{N} \]
Flavor

Wrapper
\[ P(F=\text{cherry}) \]
\[
\begin{array}{c}
\Theta
\end{array}
\]

\begin{figure}
\centering
\begin{tikzpicture}
  \node[draw, ellipse] (flavor) at (0,0) {\textit{Flavor}};
  \node[draw, rectangle] (pfc) at (3,3) {$P(F=\text{cherry})$};
  \node[draw, rectangle] (pwr) at (3,1) {$\Theta$};

  \node[draw, ellipse] (wrapper) at (0,-3) {\textit{Wrapper}};
  \node[draw, table] (table) at (3,-1) {
    \begin{tabular}{|l|c|}
    \hline
    \textit{F} & \textit{P(W=red|F)} \\
    \hline
    \textit{cherry} & \Theta_1 \\
    \textit{lime} & \Theta_2 \\
    \hline
    \end{tabular}
  };

  \path[->] (flavor) edge (pfc);
  \path[->] (pfc) edge (pwr);
  \path[->] (wrapper) edge (table);
\end{tikzpicture}
\end{figure}
\( h_{\Theta, \Theta_1, \Theta_2} \)

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

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

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

\[\Theta_2 = \frac{r_l}{r_l + g_l} = \frac{r_l}{l}\]
\( h_{\Theta, \Theta_1, \Theta_2} \)

\[
\begin{array}{c}
P(F=\text{cherry}) \\
\Theta \\
\end{array}
\]

\begin{tikzpicture}
  
  \node (flavor) at (0,0) {Flavor};
  \node (wrapper) at (0,-3) {Wrapper};
  \node (table) at (2,-2) {
    \begin{tabular}{|c|c|}
      \hline
      \( F \) & \( P(W=\text{red}|F) \) \\
      \hline
      \text{cherry} & \Theta_1 \\
      \text{lime} & \Theta_2 \\
      \hline
    \end{tabular}
  };

  \draw[->] (flavor) -- (wrapper);
  \draw[->] (table) -- (flavor);
  \end{tikzpicture}
\[ \Theta = \frac{c}{c + l} = \frac{c}{N} \]

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

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

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

Class

Attr$_1$  Attr$_2$  Attr$_3$  ...

...
Naive Bayes Models

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

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

Class

Attr₁

Attr₂

Attr₃

...
Naive Bayes Models

\{ \textit{mammal}, \textit{reptile}, \textit{fish}, \ldots \}
Naive Bayes Models

\[
\{ \text{terrorist, tourist} \}
\]

Class

\[
\begin{align*}
\text{Arrival Mode} & \quad \text{One-way Ticket} & \quad \text{Furtive Manner} \\
\end{align*}
\]
Naive Bayes Models

Disease

Test$_1$  Test$_2$  Test$_3$  ...
Learning Naive Bayes Models

• Naive Bayes model with $n$ Boolean attributes requires $2^n + 1$ parameters.

• Maximum likelihood hypothesis can be found with no search.

• Probabilities are observed frequencies.

• 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.
\{ \text{terrorist, tourist} \}

\begin{tabular}{l}
\textbf{Class} \\
\end{tabular}

\begin{tabular}{lll}
\textbf{Arrival} & \textbf{One-way} & \textbf{Furtive} \\
\textbf{Mode} & \textbf{Ticket} & \textbf{Manner} \\
\end{tabular}

\ldots
<table>
<thead>
<tr>
<th>Arrival</th>
<th>One-Way</th>
<th>Furtive</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>taxi</td>
<td>yes</td>
<td>very</td>
<td>terrorist</td>
</tr>
<tr>
<td>car</td>
<td>no</td>
<td>none</td>
<td>tourist</td>
</tr>
<tr>
<td>car</td>
<td>yes</td>
<td>very</td>
<td>terrorist</td>
</tr>
<tr>
<td>car</td>
<td>yes</td>
<td>some</td>
<td>tourist</td>
</tr>
<tr>
<td>walk</td>
<td>yes</td>
<td>none</td>
<td>student</td>
</tr>
<tr>
<td>bus</td>
<td>no</td>
<td>some</td>
<td>tourist</td>
</tr>
</tbody>
</table>
Disease

Test₁

Test₂

Test₃

...
<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
Learning: The Big Picture for 242

- Function Learning
- Linear Regression
- Linear Classifiers
- Neural Networks
- Learning Probabilistic Models (Bayes Nets)
Tue 23 Apr & Thu 25 Apr
Posters!
Get there early...