Hidden Values
Expectation Maximization
Mixture Models

Question from last time:
Why does bootstrap select ~63% of data in the end?
\[ p(i \text{ selected at least once}) = 1 - (1 - \frac{1}{N})^N \]
\[ = 1 - \frac{1}{e} \approx 0.63 \]

Example of Decision Tree

<table>
<thead>
<tr>
<th>Gender</th>
<th>Age</th>
<th>Family Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>33</td>
<td>5</td>
</tr>
<tr>
<td>Female</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

(see also on Wikipedia)

Random Forests

Each decision node is "Random Decision Stump"
- picks random features (subset of features)
- uses infogain
- you can probe with "test-out" samples

\[ x \in \{ X_i \} \approx 63\% \text{ of original data} \]
Hidden Values
- learning when some values are unobserved, missing, hidden, latent

<table>
<thead>
<tr>
<th>Gender Age</th>
<th>Family Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male 33</td>
<td>5</td>
</tr>
<tr>
<td>Female 10</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
</tr>
</tbody>
</table>

Semi-Supervised Learning
- Idea: getting labels is expensive, getting unlabeled data is cheap
- can we train on \( \{X_u, y_u\} \) and \( \{X, y\} \)

\[
X_u = \begin{bmatrix}
X_u \ \\
\vdots
\end{bmatrix}
\]
\[
X = \begin{bmatrix}
X_u \ \\
\vdots
\end{bmatrix}
\]
\[
y_u = \begin{bmatrix}
y_u \ \\
\vdots
\end{bmatrix}
\]
\[
y = \begin{bmatrix}
y \ \\
\vdots
\end{bmatrix}
\]

Information inequality:
more data can't hurt

"Missing at Random" (MAR)
- The fact that it is missing does not depend on the missing value.

- E.g., Digit Classification

\[
\begin{array}{c}
2 \\
3
\end{array} \rightarrow 2. \quad \text{- missing random pixels: MAR}
\]
\[
\begin{array}{c}
2 \\
3
\end{array} \rightarrow 3. \quad \text{- hide the labels of all the } 2\text{'s examples (not MAR)}
\]

- If not MAR, you need to model WTH data is missing.

Approach #1
1. Imputation: replace ?'s with most likely value
   - e.g., guess the age of the missing
2. Fit model with "imputed values"
3. Can impute missing data with new, fitted model to try and be better
   - "hard - EM"
Probabilistic Approach

Notation: \( X: \) observable variables \( H: \) hidden variables

\[
P(X) = \sum_H p(X, H)
\]
(integral if \( H \) is a continuous random variable)

\[E_{\phi} S.L.\]

\[
p(Y_1, X_1, X_2) = \prod_{i=1}^{N} p(y_i, x_i) \prod_{h} p(x_1, x_2)
\]

Problem: We assume \(-\log p(X, H)\) is "nice" (closed-form convex)

Maximize \( p(X) \):

\[
\log(p(x)) = \log \left( \sum_H p(X, H) \right)
\]

if probability \( p(x) \) is exponential, we have familiar log-sum-exp

\[
\text{Minimize} \quad -\log(p(x)) = -\log \left( \sum_H p(X, H) \right)
\]

\[
\text{Convex because the summation inside log}
\]

\[
\log(1 + \exp(w^T x)) = \log(\exp(w^T x) + \exp(w^T x)) \quad \text{"convex"}
\]

Expectation Maximization

Local Optimizer when \( \log(p(X, H)) \) is "nice"

Problem: \[
\max_\theta p(X | \theta)
\]

\[
E_{\theta} = \frac{\sum_h p(h \mid \theta) \phi(h)}{\sum_h p(h \mid \theta)}
\]

\[
\text{E-step}: \quad Q(\theta \mid \theta^t) = E_{H \mid X, \theta} \left[ \log p(X, H \mid \theta) \right]
\]

\[
\text{M-step}: \quad \theta^t = \arg \max_{\theta} Q(\theta \mid \theta^t) = \sum_h p(h \mid \theta^t) \log p(X, h \mid \theta^t)
\]

Theorem

\[
\log(p(X \mid \theta^{t+1})) - \log(p(X \mid \theta^t)) \geq Q(\theta^{t+1} \mid \theta^t) - Q(\theta \mid \theta^t)
\]
"Convex Combination"

\[
\alpha_i x_i + \alpha_2 x_2 + \cdots + \alpha_n x_n \quad \alpha_1 > 0 \quad \sum_i \alpha_i = 1
\]

\[f(\sum \frac{\alpha_i}{\sum \alpha_i} x_i) \leq \sum \alpha_i f(x_i)\]

\[-\log p(x | \theta) = -\log \frac{\sum \alpha_i p(x, h_i | \theta)}{p(H | X, \theta)}\]

\[= -\log \left(\sum \frac{\alpha_i}{\alpha_{n_i}} p(x, h_i | \theta)\right)\]

\[= -\frac{\sum \alpha_i}{\alpha_{n_i}} \log p(x, h_i | \theta) + \frac{\sum \alpha_i}{\alpha_{n_i}} \log \alpha_{n_i}\]

\[= -\alpha_n (x | \theta^*) + \text{const} - H(\alpha) \quad \text{regular category}
\]

\[p(x | \theta) = \frac{p(x, h_i | \theta)}{p(H | x, \theta)}\]

\[-\log p(x | \theta) = -\log \left(\log p(x, h | \theta) + \log p(H | X, \theta)\right)\]

\[\sum \alpha_i \left(-\log p(x | \theta^*) = \sum \alpha_i \left(-\log p(x, h | \theta^*) + \log p(H | X, \theta^*)\right)\right)\]

\[= \sum \alpha_i \left[\log p(x, h | \theta^*) + \alpha_i \log p(H | X, \theta^*)\right]\]

\[= \alpha_n (x | \theta^*) + \text{const}
\]

"It does touch the point"
Mixture Models

- Recall fitting Gaussians.

- Want to fit probabilistic model
- but have labels
- but it would be easy if we did!
- motivates mixture models

Let \( Z_i \in \{ 1, 2, \ldots, K \} \)

\[ p_{\theta}(x_i | \theta) = p(x_i | Z_i = k, \theta) \]

"assignment"

\( Z_i \) is a "latent" variable never observed, but would be nice to know

\[ p(x_i | \theta) = \sum_{k=1}^{K} p(x_i, Z_i = k | \theta) \]

\[ \sum_{k=1}^{K} p(x_i, Z_i = k, \theta) p(Z_i = k | \theta) \]

\( K \)-means: hard - EM

with Gaussian

with \( \Sigma_k = I \)

- Usually, mixture models are fit with EM

- If one class is more complicated, can use mixture of a Gaussian
  to get rid of unimodal problem