week 6 now hw 2 due Friday

week 7 M
w career fair no class

week 8 M
w exam 1 4:00-5:30

Spring Break

search
ch 3
ch 4
ch 5
neural network

Previous class

Constructing decision trees

"learning" classification multi-valued concepts

problem data \rightarrow concise representation

look for the minimal tree that
accurately represents this table

search: for this tree

optimization problem

(goal is not explicitly defined)
goal: is to classify objects
"efficiently"
reduce uncertainty
13 examples 3 false 7 true 6
uncertainty: \( \frac{6}{13}, \frac{7}{13} \)

One question: which one would reduce the uncertainty the most?
heuristic:

<table>
<thead>
<tr>
<th>A is true</th>
<th>A is false</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{8}{13} )</td>
<td>( \frac{5}{13} )</td>
</tr>
<tr>
<td>( \frac{2}{8} )</td>
<td>( \frac{0}{5} )</td>
</tr>
</tbody>
</table>

\( B = \frac{5}{13} \) \( C = \frac{1}{13} \) \( D = \frac{5}{13} \) \( E = \frac{6}{13} \)
Entropy: measure of uncertainty
how many bits are needed
to describe the possible outcomes

one value: 0 bits
2 values: 1 bit
4 values: 2 bits

Random variable \( V \)
with values \( V_k \)
each of which
have a probability

entropy:
\[
H(V) = \sum_k p(V_k) \log_2 \frac{1}{p(V_k)}
\]

Example: fair coin flip

\[
H(\text{fair coin}) = - \left[ \frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2} \right]
\]

\[
= - \left[ \frac{1}{2} \log_2 0.5 + \frac{1}{2} \log_2 0.5 \right]
\]

\[
= - \left[ -1 + -1 \right]
\]

\[
= 1
\]
\[ H(\text{loaded}) = - \left[ (0.99 \log_2 0.99) + (0.01 \log_2 0.01) \right] \]
\[ = 0.08 \]

\[ B(q) = \text{entropy of a Boolean variable that is true with probability } q \]

\[ B(q) = - (q \log_2 q + (1-q) \log_2 (1-q)) \]

\[ p: \# \text{ of positive examples} \quad 6 \quad \text{total: 13} \]

\[ n: \# \text{ of negative examples} \quad 7 \quad \text{all: } p+n \]

\[ \text{Entropy before picking any positions} \quad B\left(\frac{p}{p+n}\right) \quad B\left(\frac{6}{13}\right) \]

\[ \text{After picking position } A \]

\[ \text{Remainder (A)} \]

\[ A \text{ has } d \text{ answers for each: how much uncertainty do all I have?} \]

\[ = \sum_{k=1}^{d} \frac{P_k + n_k}{p+n} \quad B\left(\frac{P_k}{P_k + n_k}\right) \]

\[ P_k: \# \text{ of positive examples when } A \text{'s value is } v_k \]

\[ n_k: \# \text{ of negative examples when } A \text{'s value is } v_k \]
"expectation"

lottery  0.1  1000
0.2  10
0.7  0

expected value of this lottery is

\[
0.1 \times 1000 + 0.2 \times 10 + 0.7 \times 0 = 102
\]

using the entropy formula:

gain a reduction in uncertainty when we ask a question

\[
gain(A) = B \left( \frac{p}{p+n} \right) - \text{remainder}(A)
\]

the question (attribute) that has the highest gain will win a spot at the root of the tree.
\[
\text{gain}(A) = B\left(\frac{6}{13}\right) - \left[\frac{2}{13} B\left(\frac{6}{8}\right) + \frac{5}{13} B\left(\frac{0}{5}\right)\right]
\]

= 0.50 for A

\text{B, C, D, E will have lower values for gain.}
Summary

Entropy: measure of uncertainty

Random variable \( V \) with values \( V_k \) each with probability \( P(V_k) \)

\[
H(V) = \sum_k P(V_k) \log_2 \frac{1}{P(V_k)}
\]

\[
= -\sum_k P(V_k) \log_2 P(V_k)
\]

\[
B(q) = - (q \log_2 q + (1-q) \log_2 (1-q))
\]

\[
H(\text{"goal"}) = B\left(\frac{p}{p+n}\right)
\]

Remainder (A): \[
\sum_{k=1} d \frac{P_{k+n_{k-1}}}{p+n} B\left(\frac{P_k}{P_{k+n}}\right)
\]

\[
\text{gain (A)} = H(\text{goal}) - \text{remainder (A)}
\]