Constructing Decision Trees

Alice Gao

Lecture 9

Readings: R & N 18.3

Based on work by K. Leyton-Brown, K. Larson, and P. van Beek

Outline

Learning Goals

By the end of the lecture, you should be able to

- Compute the entropy of a probability distribution.
- Compute the expected information gain for selecting a feature.
- Trace the execution of and implement the ID3 algorithm.

Jeeves the valet - training set

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Jeeves the valet - the test set

Day	Outlook	Temp	Humidity	Wind	Tennis?
1	Sunny	Mild	High	Strong	No
2	Rain	Hot	Normal	Strong	No
3	Rain	Cool	High	Strong	No
4	Overcast	Hot	High	Strong	Yes
5	Overcast	Cool	Normal	Weak	Yes
6	Rain	Hot	High	Weak	Yes
7	Overcast	Mild	Normal	Weak	Yes
8	Overcast	Cool	High	Weak	Yes
9	Rain	Cool	High	Weak	Yes
10	Rain	Mild	Normal	Strong	No
11	Overcast	Mild	High	Weak	Yes
12	Sunny	Mild	Normal	Weak	Yes
13	Sunny	Cool	High	Strong	No
14	Sunny	Cool	High	Weak	No

Constructing the "best" decision tree

We want a decision tree to be

- Consistent with all the training examples and
- As small (shallow) as possible.

Unfortunately, it is intractable to find the smallest consistent decision tree.
Thus, we will use heuristics to find a small consistent tree.

How do we learn a decision tree?

It is computationally intractable to find the optimal order of testing features.
The idea of the ID3 algorithm:

- A greedy divide-and-conquer approach.
- Test the most important feature at each step.
- Solve the sub-problems recursively.

The ID3 algorithm

Algorithm 1 ID3 Algorithm (Features, Examples)

1: If all examples are positive, return a leaf node with decision yes.
2: If all examples are negative, return a leaf node with decision no.
3: If no features left, return a leaf node with the majority decision of the examples.
4: If no examples left, return a leaf node with the majority decision of the examples in the parent.
5: else
6: choose the most important feature f
7: **for** each value v of feature f **do**
8: add arc with label v
9: add subtree $ID3(F - f, s \in S | f(s) = v)$
10: **end for**

Base cases of the ID3 algorithm

No features left:
See notes on the course website.

No examples left:
See notes on the course website.

Choosing the most important feature

- Want a feature that allows us to make a decision as soon as possible — reduce uncertainty as much as possible
- Information content of a feature = uncertainty before testing the feature - uncertainty after testing the feature
- Measure uncertainty using the notion of entropy.

Given a distribution $P(c_1), ..., P(c_k)$ over k outcomes $c_1, ..., c_k$, the entropy of the distribution is

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

CQ: Entropy of a distribution

CQ: What is the entropy of the distribution $(0.5, 0.5)$?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

(A) 0.2
(B) 0.4
(C) 0.6
(D) 0.8
(E) 1

CQ: Entropy of a distribution

CQ: What is the entropy of the distribution $(0.01, 0.99)$?

$$I(P(c_1), ..., P(c_k)) = -\sum_{i=1}^{k} P(c_i) \log_2(P(c_i))$$

(A) 0.02
(B) 0.04
(C) 0.06
(D) 0.08
(E) 0.1

Entropy of a distribution over two outcomes

Consider a distribution $p, 1-p$ where $0 \leq p \leq 1$.

- What is the maximum entropy of this distribution?
- What is the minimum entropy of this distribution?

Expected information gain of testing a feature

Before testing a feature, there are p positive examples and n negative examples. The entropy before testing the feature is

$$I(\frac{p}{p+n}, \frac{n}{p+n})$$

Suppose that the feature has k values $v_1, ..., v_k$. After testing the feature, for each value v_i, there are p_i positive examples and n_i negative examples. The expected entropy after testing the feature is

$$\sum_{i=1}^{k} \frac{p_i + n_i}{p+n} * I(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i})$$

The expected information gain is

$$I(\frac{p}{p+n}, \frac{n}{p+n}) - \sum_{i=1}^{k} \frac{p_i + n_i}{p+n} * I(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i})$$

CQ: Entropy and information gain

CQ: What is the entropy of the examples
before we select a feature for the root node of the tree?

(A) 0.54

(B) 0.64

(C) 0.74

(D) 0.84

(E) 0.94

CQ: Entropy and information gain

CQ: What is the expected information gain
if we select **Outlook** as the root node of the tree?

(A) 0.237

(B) 0.247

(C) 0.257

(D) 0.267

(E) 0.277

CQ: Entropy and information gain

CQ: What is the expected information gain
if we select **Humidity** as the root node of the tree?

(A) 0.151

(B) 0.251

(C) 0.351

(D) 0.451

(E) 0.551

Revisiting the Learning Goals

By the end of the lecture, you should be able to

- Compute the entropy of a probability distribution.
- Compute the expected information gain for selecting a feature.
- Describe/trace/implement the ID3 algorithm.