UNIVERSITY OF LEEDS

Class: Machine Learning

Decision Trees

Instructor: Matteo Leonetti

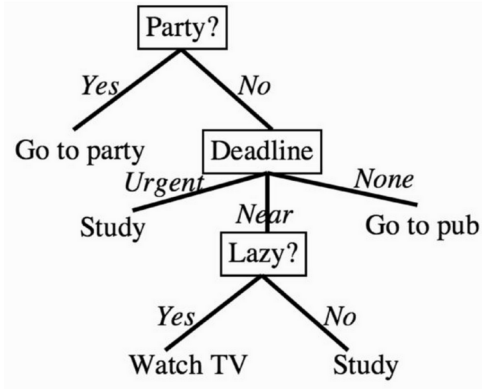
Learning outcomes

UNIVERSITY OF LEEDS

- Define the entropy of a set
- Compute the entropy of a given set
- Define the information gain for a given feature
- Define the Gini Impurity of a set
- Implement the ID3 and CART algorithms

Making Decisions

UNIVERSITY OF LEEDS



Nonmetric data

How to choose the variable for each split?

This is an example from your book, about deciding what to do in the evening. As with any other ML method, we first need to decide what the decision is based on (the *features*). In this case, it's based on whether or not there is a party, whether there is a deadline, and lastly whether you feel lazy or not.

Differently from NNs, and most other ML methods, decision trees can work with *non-metric* data, that is, features that are not numbers.

Building a decision tree amounts to deciding the order in which the splits should be made.

History

UNIVERSITY OF LEEDS

1983 - Ross Quinlan (U. of Sidney)

Learning efficient classification procedures and their application to chess end games.

Entropy and information

UNIVERSITY OF LEEDS

How much information do I receive, with a message X?

X a random variable over possible messages

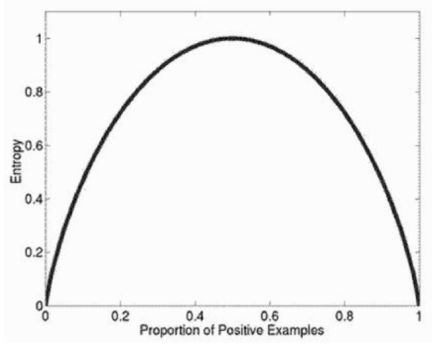
Information

$$I(x) = -\log_2 P(x)$$

Entropy

$$\mathrm{H} = E[I] = \sum_i -p_i \log_2 p_i$$

$$0 \log_2 0 = 0$$



In order to decide which variable to split on next, we are going to use a concept from information theory, that is, the quantity of information, and entropy.

Information is a measure of “surprise”, an unlikely event that happens carries much more information than an event we were pretty sure would happen anyway.

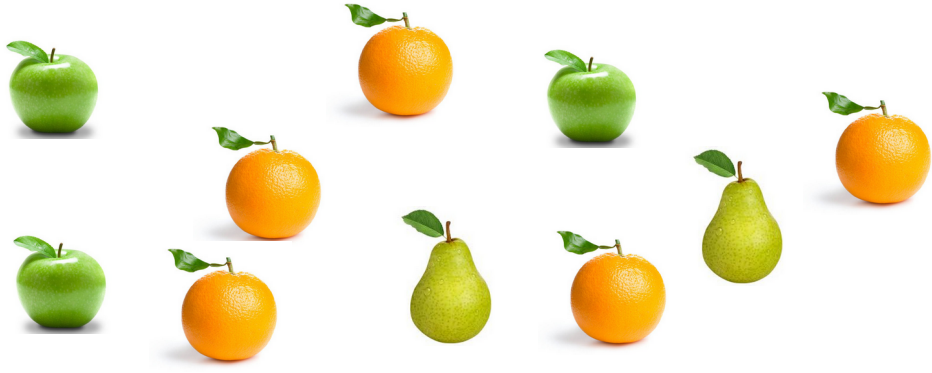
The entropy of a distribution is the average (more precisely, the expected value) of the information, that is, of surprise. For example, if there are only two possible outcomes, I can expect no surprise if one of the two has 0 probability, since I am already sure that the outcome will be the other event, with probability 1. On the other hand, if both events are equally likely, with probability 1/2, I cannot expect one more than the other, which means that my level of surprise is maximum.

With N possible outcomes, the maximum value of the entropy is $\log_2(N)$.

Fruits

UNIVERSITY OF LEEDS

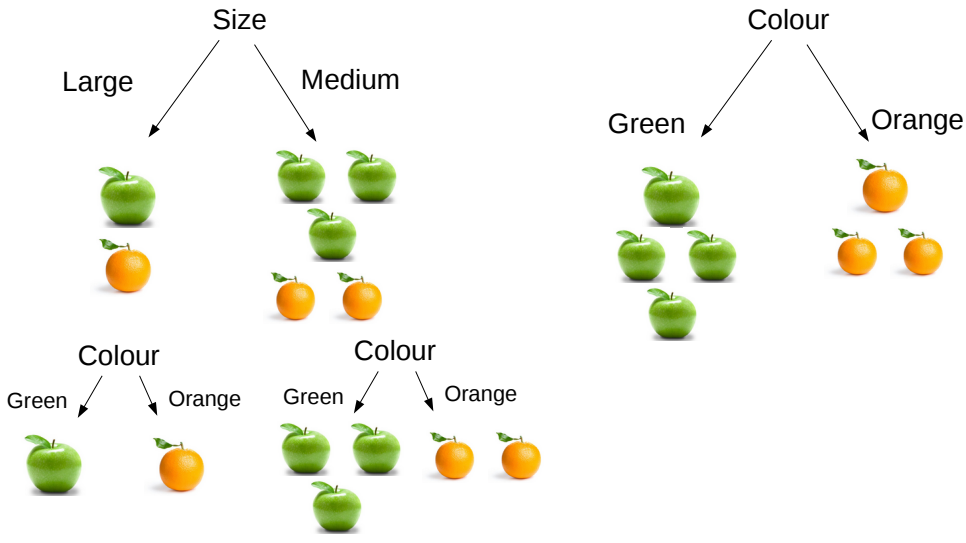
$$\mathrm{H} = E[I] = \sum_{i \in classes} -p_i \log_2 p_i$$



$$\mathrm{H} = \overbrace{-\frac{3}{10}\log_2\frac{3}{10}}^{\text{Apples}} \overbrace{-\frac{5}{10}\log_2\frac{5}{10}}^{\text{Oranges}} \overbrace{-\frac{2}{10}\log_2\frac{2}{10}}^{\text{Pears}} = 1.485$$

Apples and Oranges

UNIVERSITY OF LEEDS



Let's ignore pears for simplicity, and only consider a classifier that is intended to distinguish between apples and oranges.

We consider two features, Size and Colour, each one with two possible values.

Which feature is most appropriate for the first split?

We want the feature that's most "informative", in an information theoretic sense, that is, by looking at the distributions of the two classes before and after the split. If the distribution has decreased in entropy we have reduced the surprise, which means we are more certain. A reduction in entropy corresponds to an increase in information.

Entropy of the set

UNIVERSITY OF LEEDS

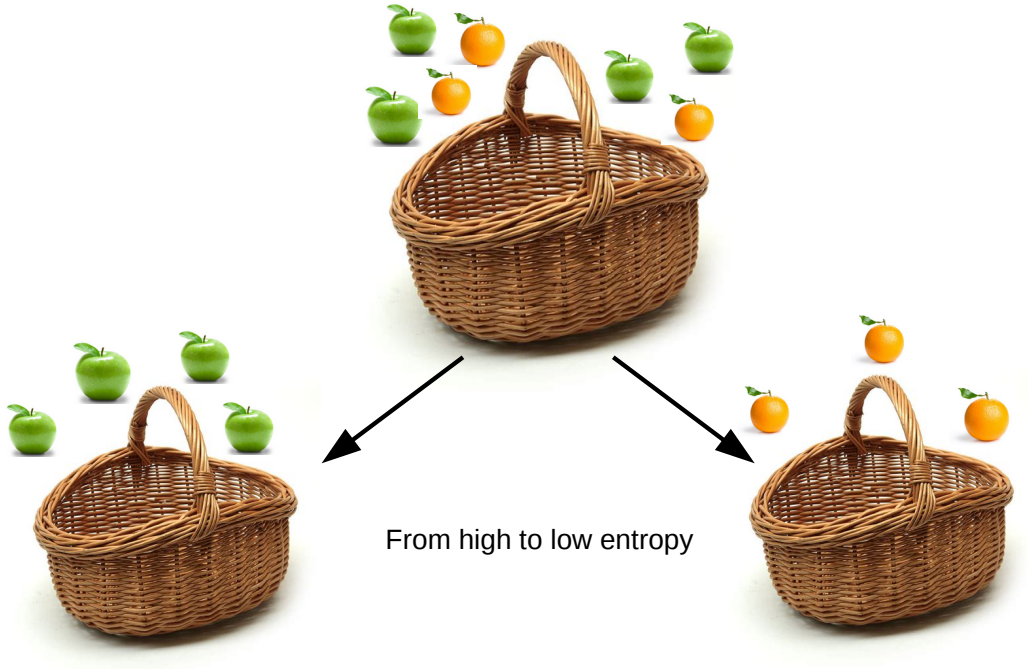
$$H = -p_O \log_2(p_O) - p_A \log_2(p_A) = -\frac{3}{7}\log_2(\frac{3}{7}) - \frac{4}{7}\log_2(\frac{4}{7}) = 0.985$$

First of all, let's compute the entropy of the initial set containing both oranges and apples.

The initial entropy tells me how much I would be able to predict what I would get, if I picked an element of this set. Since Apples and Oranges have almost the same probability, the entropy is closed to maximum.

Entropy of the set

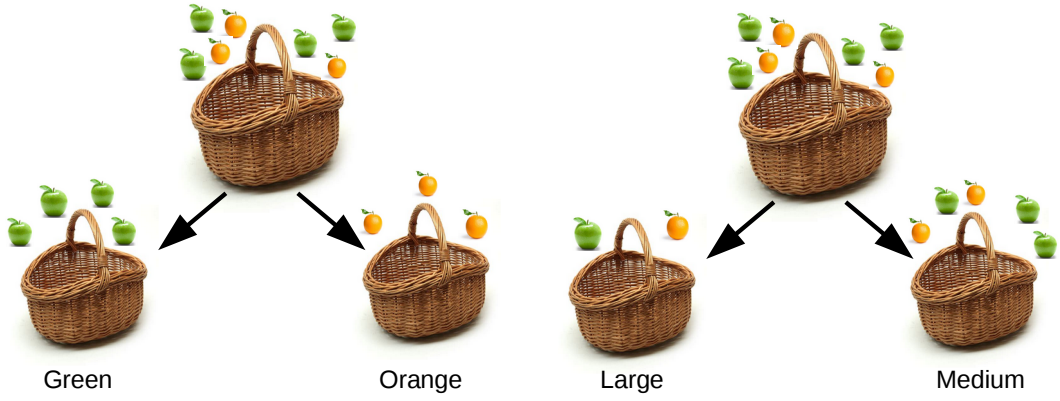
UNIVERSITY OF LEEDS



The split creates two sets whose internal coherence is higher than the original one, that is, with less entropy.

Entropy of the set

UNIVERSITY OF LEEDS



$$H_{colour} = \overbrace{\frac{4}{7}}^{\text{fraction in Green}} \overbrace{(0)}^{\text{entropy of Green}} + \overbrace{\frac{3}{7}}^{\text{fraction in Orange}} \overbrace{(0)}^{\text{entropy of Orange}} = 0$$

$$H_{size} = \overbrace{\frac{2}{7}}^{\text{fraction in Large}} \overbrace{\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right)}^{\text{entropy of Large}} + \overbrace{\frac{5}{7}}^{\text{fraction in Medium}} \overbrace{\left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right)}^{\text{entropy of Medium}} = 0.98$$

So we consider both possible splits, one based on the feature “colour” and the other on the feature “size”.

We compute the entropy of the sets resulting from each split.

One of the two splits results in perfect classification, so has 0 entropy. Splitting with respect to size, instead, results in two sets that have almost the same entropy as before.

Zero entropy means certainty! So the split with respect to colour is the most informative.

Entropy of the set

UNIVERSITY OF LEEDS

$H = 0.985$

$H_{colour} = 0$

$G(\text{Colour}) = H - H_{colour} = 0.985$

$H_{size} = 0.98$

$G(\text{Size}) = H - H_{size} = 0.005$

A decrease in entropy corresponds to an increase in information.

What we want to maximise here is the difference in entropy, which for the reason I just explained is also called *information gain*.

The information gain for the split on colour is much higher than the information gain for the split on size.

Information gain

UNIVERSITY OF LEEDS

Set of elements

elements in S with feature F = f

$$G(S,F) = \mathrm{H}(S) - \sum_{f \in values(F)} \frac{|S_f|}{|S|} \mathrm{H}(S_f)$$

Feature

compare with:

$$\mathrm{H}_{size} = \overbrace{\frac{2}{7}}^{\text{fraction in Large}} \overbrace{\left(-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right)}^{\text{entropy of Large}} + \overbrace{\frac{5}{7}}^{\text{fraction in Medium}} \overbrace{\left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right)}^{\text{entropy of Medium}} = 0.98$$

In general the information gain is expressed as shown above.

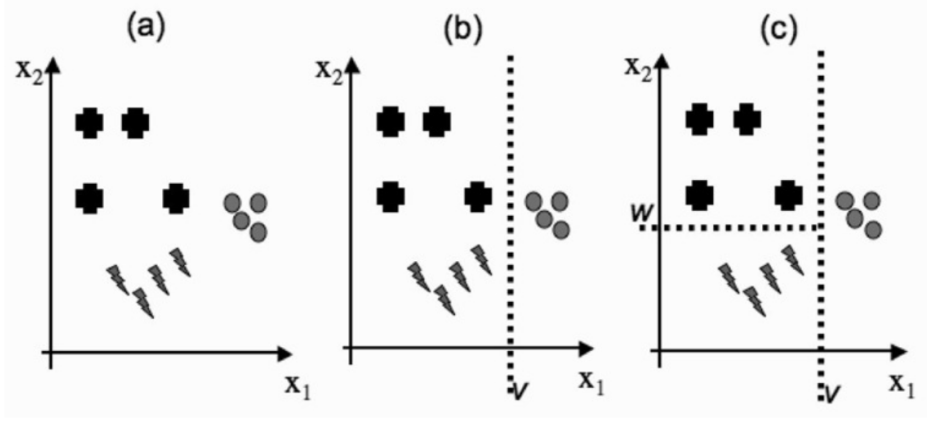
The ID3 algorithm

UNIVERSITY OF LEEDS

- If all examples have the same label:
  - return a leaf with that label
- Else if there are no features left to test:
  - return a leaf with the most common label
- Else:
  - choose the feature $\hat{F}$ that maximises the information gain of S to be the next node using **Equation (12.2)**
  - add a branch from the node for each possible value f in $\hat{F}$
  - for each branch:
    * calculate S_f by removing $\hat{F}$ from the set of features
    * recursively call the algorithm with S_f, to compute the gain relative to the current set of examples

Visualizing splits

UNIVERSITY OF LEEDS



Characteristics

UNIVERSITY OF LEEDS

Greedy with respect to G → potential local minimum

Deals with noisy data (by assigning the label to most common class)

Always uses all the features → prone to overfitting

↓

Pruning

Continuous variables → C4.5

Missing attributes

ID3 is a *greedy* algorithm (like gradient descent) and as such is prone to local minima.

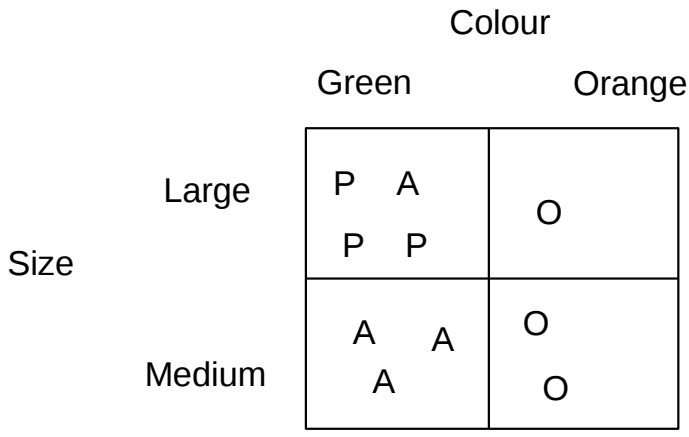
It is has also advantages though: we mentioned before that it can classify non-metric data, and is also quite robust to noise in the data.

On the other hand, by default it uses all the features, which is prone to overfitting.

A more advanced version which includes pruning, and works with continuous variables and missing attributes is called C4.5, and it is the algorithm that you will most commonly find in real life (but the key concepts have been introduced in ID3).

A Different Criterion: Gini Impurity

UNIVERSITY OF LEEDS



$$G(S) = \overbrace{\frac{4}{10}}^{apples}\overbrace{\left(\frac{3}{10}+\frac{3}{10}\right)}^{not\ apples} + \overbrace{\frac{3}{10}}^{pears}\overbrace{\left(\frac{4}{10}+\frac{3}{10}\right)}^{not\ pears} + \overbrace{\frac{3}{10}}^{oranges}\overbrace{\left(\frac{4}{10}+\frac{3}{10}\right)}^{not\ oranges} = \sum_{i}^{C} p_i(1-p_i)$$

Let's now re-introduce pears, so that we have more than 2 classes.

We are now considering a criterion for splitting different from entropy. We asked the question: if I assigned the class according to its frequency in the data, how likely am I to be wrong?

This is called the *Gini Impurity*.

Gini split:

# of classes

$$G(S)=\sum_{i}^{C} p_i(1-p_i)=\sum_{i}^{C}(p_i-p_i^2)=\sum_{i}^{C} p_i-\sum_{i}^{C} p_i^2=1-\sum_{i}^{C} p_i^2$$

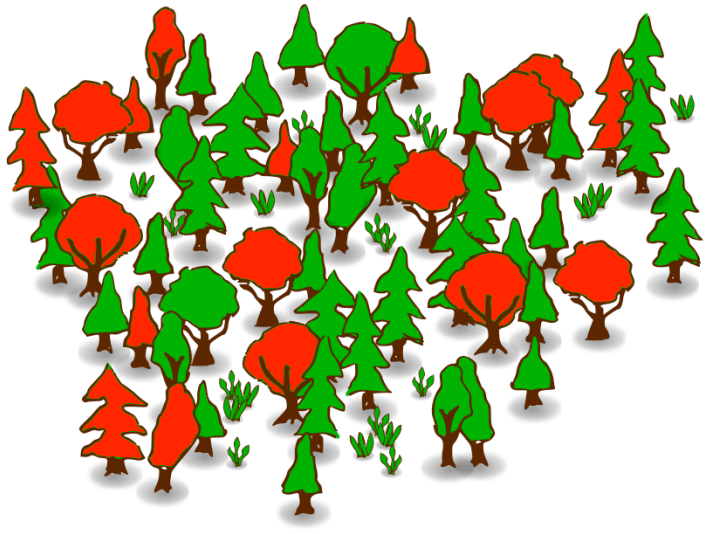
$$G(S,F)=G(S)-\sum_{f\in values(F)}\frac{|S_f|}{|S|}G(S_f)$$

If we use the same definition of the information gain but with the gini impurity instead of entropy we obtain the splitting principle of another algorithm: CART.

CART splits on the variable with the highest gini gain, as opposed to information gain.

Random forests

UNIVERSITY OF LEEDS



Trees are often used in collections called (quite appropriately) *forests*.

Each tree is obtained by introducing randomness in some aspects, the most common are:

- bagging, a random subset of the data is used
- a random subset of the features is used.

Each tree “votes” for a classification, and the classification with most votes is returned by the random forest.

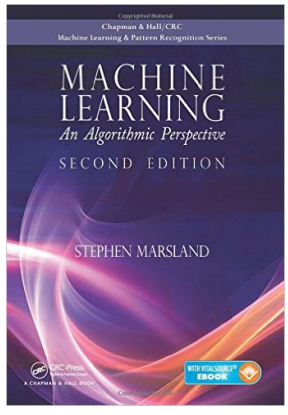
Conclusion

Learning outcomes

UNIVERSITY OF LEEDS

- Define the entropy of a set
- Compute the entropy of a given set
- Define the information gain for a given feature
- Define the Gini Impurity of a set
- Implement the ID3 and CART algorithms

UNIVERSITY OF LEEDS



Chapter 12