Machine Learning

Lecture 8
Information theory and Decision Trees

Justin Pearson[1]
mailto:it-1dl034@lists.uu.se

[1] http://user.it.uu.se/~justin/Hugo/courses/machinelearning/

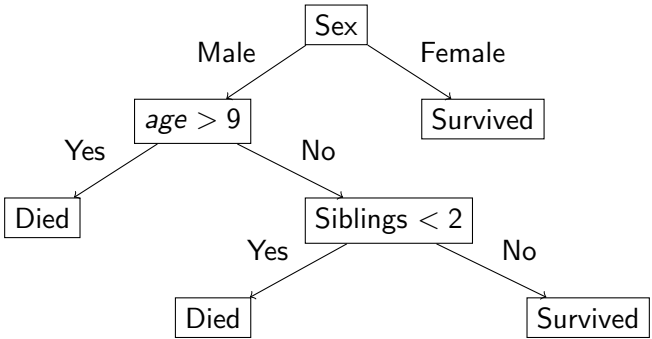
Plan :

- Decision trees.
- Short introduction to Information Theory.
- Complexity results on constructing decision trees.
- Constructing Decision trees using information theory. (ID3 algorithm)
- Pointers to other ways of constructing trees.

The focus will be on decision trees for classification. We will look at regression in a later lecture.

Decision Trees

Suppose we are trying to classify the if a person survived the Titanic. We might learn the following tree :

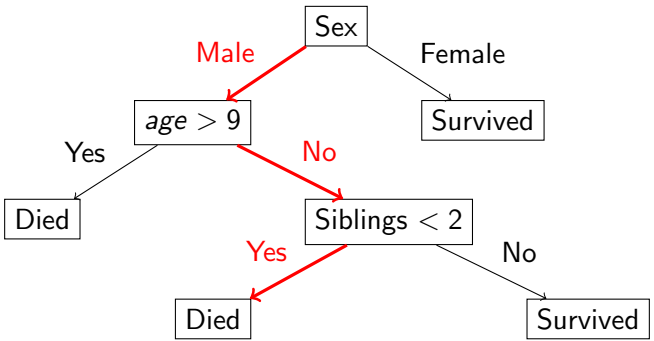


Decision Trees

A decision tree is a rooted directed acylic graph. The leaf nodes represent classes or values.
For the moment all of our classifications will be categorical.
An assignment of variables gives a path in the tree. The resulting classification is the value in the leaf node.

Decision Tree Example

Given a male aged 6 with 0 siblings, then he would have died.



Advantages of Decision Trees

- We can represent non-linear functions
- Small trees are easy to understand and give us an explanation of why a classification was made.

Explainable AI: Why did the machine learning refuse me a bank loan? This is an important ethical, and one day maybe a legal requirement for certain types of machine learning models.

Decision Trees

The big question:

- Given some data how do we learn a decision tree?

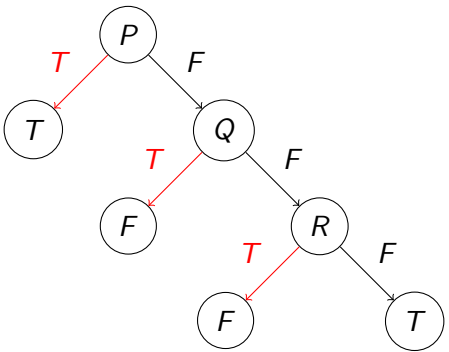
Issues:

- How do we avoid overfitting?
- How do we learn small trees?
- We want the ordering of the nodes to some how represent the importance of the features.

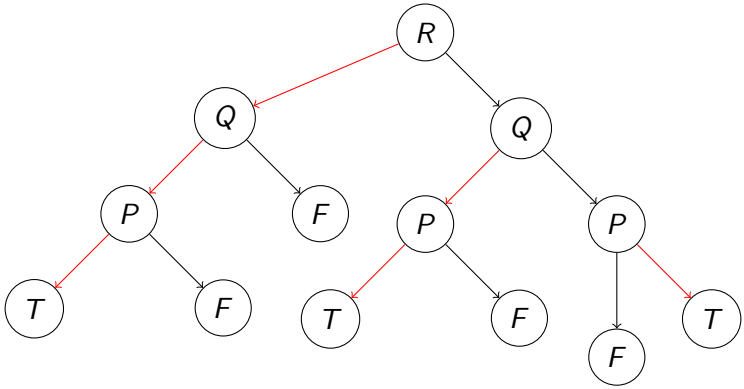
Decision Trees — Boolean Values

Even for the Boolean case the order you consider the features gives different size and shape trees.

Unless I've made a mistake the this tree and the tree on the next slide represent the same function.



Decision Trees — Boolean Values



Red edges represent true branches.

Decision Trees — Boolean Values — Some facts

- For Boolean functions you can get exponential blow up in the size for different orders.
- It is NP-hard to find an ordering that gives the smallest tree.

Implications for machine learning: We have to use heuristics to find small trees. We will look at one approach using information theory.

Revision?

Definitions of logarithms:

$$\log_2 x = y \Leftrightarrow 2^y = x$$

Thus $\log_2 4 = 2$ since $2^2 = 4$, and $\log_2 1 = 0$ since $2^0 = 1$.
Don't forget:

$$\log_b x = \frac{\log x}{\log b}$$

Measuring Information

In computer science we often measure capacity on a logarithmic scale.
For example:

- 2K is twice the size of 1K, even though the number of states (possible combinations of 0s and 1s) grows exponentially: 2^{2048} vs 2^{1024}.

Measuring information has something to do with probability.

Information Theory

- Information was invented by Claude Shannon in the context of communication theory.
- If I tell you that it is cold in winter time, this is a low information message. It is not unexpected.[2]
- If I tell you that it is 25C on January 1st in Uppsala, this is an unexpected message, so it contains a lot of information.

[2] This is a very British way of saying: "It is expected".

Measuring Information

Given some discrete probability distribution over a set of events, denoted $p_1 \ldots, p_n$. Then the information is defined to be

$$H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i) = \sum_{i=1}^{n} p_i \log_2(\frac{1}{p_i})$$

This is related to entropy in statistical mechanics. As computer scientists we take logs to the base 2, and so we measure information in bits.

Properties of H and I

Given an event that occurs with probability p we want a $I(p)$ that measures the information of that event. We want I to have certain properties

- $I(p)$ is monotonically decreasing in p. The higher the probability of the event, the less information.
- $I(p) \geq 0$.
- $I(1) = 0$. An even with probability 1 has no information.
- if p and q are independent events then we want $I(pq) = I(p) + I(q)$.

Given these properties the only mathematically sensible choice is

$$I(p) = \log(\frac{1}{p}) = -log(p)$$

Properties of H and I

$$I(p) = -\log(p)$$

- Remember $\log_2 1 = 0$, so if you have an event if probability 1 then $log(1/1) = 0$. So an expected event has no information.
- As p gets smaller and smaller $log_2(1/p)$ gets bigger. Unexpected events have more information.

Another view of H

- H is the expected value or average value of $I(p)$.
- Let X be a random variable. If X takes the values $X_1, \ldots, X_n$ with probability $p_1, \ldots, p_n$ then the expected value of X is

$$\mathbb{E}(X) = \sum_{i=1}^{n} p_i x_i$$

- Then we can write $H(X)$ as

$$H(X) = \mathbb{E}(I(X))$$

where $I(x_i)$ is the information gained when event x happens that is

$$I(x_i) = \log_2(\frac{1}{p_i})$$

Flipping a fair coin

Given a coin with an equal probability of being heads or tails then we can calculate the entropy.

$$H(\frac{1}{2}, \frac{1}{2}) = -\frac{1}{2}log(\frac{1}{2}) - -\frac{1}{2}log(\frac{1}{2}) = -(log\frac{1}{2})$$

Using properties of logarithms we have that (remember that we are doing logarithms in base 2.

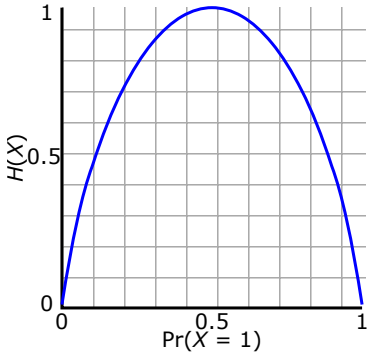
$$-(log\frac{1}{2}) = -(log1 - log2) = -(0 - 1) = 1$$

You get some information (1 bit) from observing the result of the coin toss.

Unfair coin with probability p

$$H(p, 1-p) = -p \log p - (1-p) \log(1-p)$$

If you plot the graph you get



So if you have a biased coin you get less information when you are told the result of the coin flip.

Towards decision trees

Given a sample with attributes we will try to estimate H.
For example, suppose we have 47 people, 23 have a car, and 24 do not have a car. Then the entropy will be

$$-\frac{23}{47}\log_2\frac{23}{47}-\frac{24}{47}\log_2\frac{24}{27}\approx 0.9997$$

Towards Decision Trees — Running Example

Consider the following data set (taken from Wikipedia)

Class	Mut1	Mut2	Mut3	Mut4
C	1	1	1	0
C	1	1	0	1
C	1	0	1	1
C	0	1	1	0
NC	0	0	0	0
NC	0	1	0	0
NC	1	1	0	0

The entropy of the data set is

$$-\underbrace{\frac{4}{7}\log_2\frac{4}{7}}_{\text{C}}-\overbrace{\frac{3}{7}\log_2\frac{3}{7}}^{\text{NC}}\approx 0.985$$

Towards decision trees

Suppose we have a sample where each data item has a number of attributes $A_1, \ldots A_n$. For the moment assume that the attributes are binary.
We can form the sets X_i^T all the samples that have attribute i, and X_i^F all the samples that do not have attribute i.
At the top of the tree we want to split on the attribute that gives us the most information gain.

Conditional Entropy

If you look up the formula for conditional entropy the formula can look rather complicated. Conditional entropy is nothing more the entropy of some conditional event.

$$H(X|Y=a) = -\sum_{x \in X} P(x|Y=a) \log_2 P(x|Y=a)$$

You can use the fact that $P(X|Y) = \frac{P(X,Y)}{P(X)}$ or think about what you are trying to calculate.

Conditional Entropy – Example

Consider the following observations measured over 100 days.

	Cloudy (c)	Not Cloudy ($\overline{c}$)
Raining (r)	24/100	1/100
Not Raining ($\overline{r}$)	25/100	50/100

Suppose that it is raining. What are the two probabilities $P(c|r)$ and $P(\overline{c}|r)$

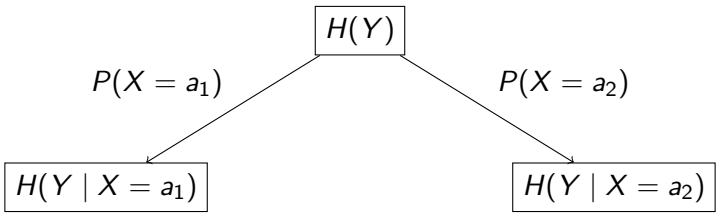
- $P(c|r)$ look at the raining row in the table, there are 25 observed days and 24 cloudy days so we get $24/25$.
- $P(\overline{c}|r)$ is simply $1/25$.

So

$$H(Y|X=r) = -\underbrace{\frac{24}{25}\log_2\frac{24}{25}}_{\text{Cloudy}} - \overbrace{\frac{1}{25}\log_2\frac{1}{25}}^{\text{NotCloudy}} \approx 0.24$$

Information Gain

Assume that the attribute X takes the values a_1 and a_2.



Pick an attribute X that maximises

$$H(Y) - \left(\underbrace{P(X = a_1)H(Y \mid X = a_1) + P(X = a_2)H(Y \mid X = a_2)}_{\text{average information when split}} \right)$$

Information Gain

Assume X takes the values $r_1, \ldots, r_n$. Then the information gain on splitting the sample on the attribute X is

$$H(Y) - \sum_{i=1}^{n} P(X = r_i) H(Y|X = r_i)$$

If you are trying to estimate this from data, then let $|X = r|$ denote the size subset of the sample where the feature X has the value r and so the information gain is:

$$H(Y) - \sum_{i=1}^{n} \frac{|X = r_i|}{|Y|} H(Y|X = r_i)$$

Information Gain

Looking at the value :

$$G = H(Y) - \sum_{i=1}^{n} P(X = r_i) H(Y|X = r_i)$$

If G is small then

$$\sum_{i=1}^{n} P(X = r_i) H(Y|X = r_i)$$

is about the same size as $H(Y)$, then splitting on that attribute is not a good idea.

Information Gain

Looking at the value :

$$G = H(Y) - \sum_{i=1}^{n} P(X = r_i) H(Y|X = r_i)$$

If G is large this means that

$$\sum_{i=1}^{n} P(X = r_i) H(Y|X = r_i)$$

is small relative to $H(Y)$, then splitting on that attribute is a good idea.

Running Example

Let's split on the parameter Mut1, that has two values 0 and 1. So we get two sets.
First Mut1 equals 0

Class	Mut1	Mut2	Mut3	Mu4
C	0	1	1	0
NC	0	0	0	0
NC	0	1	0	0

Second Mut1 equals 1

Class	Mut1	Mut2	Mut3	Mu4
C	1	1	1	0
C	1	1	0	1
C	1	0	1	1
NC	1	1	0	0

Running Example — Relative Entropy

For Mut1 equals 0

Class	Mut1	Mut2	Mut3	Mu4
C	0	1	1	0
NC	0	0	0	0
NC	0	1	0	0

The entropy $H(Y \mid \mathrm{Mut1} = 0)$ is

$$-\underbrace{\frac{1}{3}\log_2(\frac{1}{3})}_{\mathrm{C}} - \overbrace{\frac{2}{3}\log\frac{2}{3}}^{\mathrm{NC}} \approx 0.918$$

Running Example — Relative Entropy

For Mut1 equals 1

Class	Mut1	Mut2	Mut3	Mu4
C	1	1	1	0
C	1	1	0	1
C	1	0	1	1
NC	1	1	0	0

The entropy $H(Y \mid \mathrm{Mut1} = 1)$ is

$$\underbrace{-\frac{3}{4}\log_2(\frac{3}{4})}_{\mathrm{C}} \overbrace{-\frac{1}{4}\log\frac{1}{4}}^{\mathrm{NC}} \approx 0.811$$

Running example — Information gain

To calculate the information gain if you split on the attribute Mut1. You need

- The entropy at the root $H(Y) = 0.985$.
- The conditional entropy of the two splits $H(Y \mid \mathrm{Mut1} = 0) = 0.918$ and $H(Y \mid \mathrm{Mut1} = 1) = 0.811$.

Then the weighted sum gives you

$$H(Y) - \left(\frac{|\mathrm{Mut1} = 0|}{7} H(Y \mid \mathrm{Mut1} = 0) + \frac{|\mathrm{Mut1} = 1|}{7} H(Y \mid \mathrm{Mut1} = 1) \right)$$

which equals

$$H(Y) - (\frac{3}{7}0.918 + \frac{4}{7}0.811) \approx 0.128$$

Running Example

If you do the calculation for all the attributes then it is better to split on Mut3.

Mut1	0.128
Mut2	0.006
Mut3	0.521
Mut4	0.292

Running Example — Split with Mut3

Class	Mut1	Mut2	Mut3	Mut4
C	1	1	0	1
NC	0	0	0	0
NC	0	1	0	0
NC	1	1	0	0

Class	Mut1	Mut2	Mut3	Mut4
C	1	1	1	0
C	1	0	1	1
C	0	1	1	0

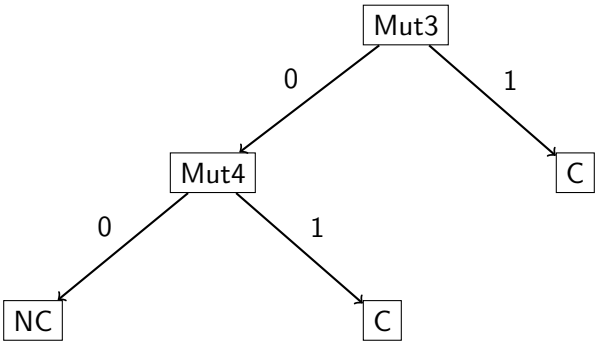
Notice when $\mathrm{Mut3} = 1$ then we know which class we are in.

Running Example – Split on $\mathrm{Mut3} = 0$

Class	Mut1	Mut2	Mut3	Mut4
C	1	1	0	1
NC	0	0	0	0
NC	0	1	0	0
NC	1	1	0	0

If you do the information calculation again then the best thing to split on is $\mathrm{Mut4}$.

Final Decision Tree



Decision trees other directions

- You can use decision trees for regression, we will look at it in a later lecture.
- There are other measures instead of Entropy such as the Gini coefficient and the miss classification ratio. Obviously complicated, but Gini is faster to compute, while entopry can give you better results.

Advantages/Disadvantages of Decision trees

- Easy to understand what the algorithm has learned.
- Can learn very non-linear boundaries.
- Does not require that much preprocessing
- Not many parameters to tune.

Advantages/Disadvantages of Decision trees

- Prone to overfitting
- Small changes in the data can mean that you learn very different trees.
- Computationally more expensive than other methods.

Later we'll look at Ensemble methods, but to avoid overfitting you can try to learn smaller trees and not use all the features in your data set.