

Pattern Recognition: Week 6

Decision trees

John Quinn

April 23, 2008

Deciding whether to give loans

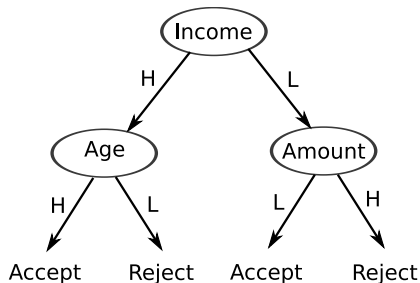
Imagine you are a loan manager at a bank, and have to decide whether to accept applications for loans. You have some data about previous loan applicants and whether they managed to repay or not:

Age	Sex	Income	Loan amount	Repayed?
H	M	H	M	Y
L	F	M	M	Y
M	M	M	L	Y
M	M	M	H	N
L	F	L	M	N
H	F	H	M	Y
M	M	M	L	N
M	F	M	H	Y

How can you use this information to help you make a decision?

Decision trees

Decision trees split up the data one attribute at a time. The leaf nodes in the tree are the output (e.g. whether to accept or reject the loan)



How would you try and learn an effective tree of this sort from the previous data?

Applications of decision trees

Decision tree learning has been applied to many problems.

- Diagnosis of medical problems
- Fraud detection
- Intrusion detection in networks
- Classifying faults in machinery
- Recognising disease in plants
- Forecasting high electricity load

How to choose attributes

We have to find some way of choosing which variable/attribute to split on. A good variable must in some way provide the best information about the target.

There are different ways of working this out. . .

Information gain

We can calculate how much information about the target variable is in each variable. This is related to the entropy of the target variable after it is split up. Entropy in this context can be thought of as a measure of disorganisation or disorder – the lower it is, the more organised the target variable is.

We are trying to decide the best new node of the tree. The information gain for a potential node can be calculated as follows:

$$I = \sum_{v \in \text{values}} p(v) \sum_{o \in \text{outputs}} p(o, v) \log_2 p(o, v)$$

where $p(v)$ is the proportion of records with value v , and $p(o, v)$ is the proportion of records with value v having output o .

Amount	Repayed?	
M	Y	$p(\text{Amt} = L) = \frac{2}{8}$
M	Y	$p(\text{Amt} = M) = \frac{4}{8}$
L	Y	$p(\text{Amt} = H) = \frac{2}{8}$
H	N	$p(\text{Repayed} = Y, \text{Amt} = L) = \frac{1}{2}$
M	N	$p(\text{Repayed} = N, \text{Amt} = L) = \frac{1}{2}$
M	Y	$p(\text{Repayed} = Y, \text{Amt} = M) = \frac{3}{4}$
L	N	$p(\text{Repayed} = N, \text{Amt} = M) = \frac{1}{4}$
H	N	$p(\text{Repayed} = Y, \text{Amt} = H) = 0$
		$p(\text{Repayed} = N, \text{Amt} = H) = 1$

So to calculate the score for the 'Loan amount' column, we have

$$\begin{aligned} I &= \sum_{v \in \text{values}} p(v) \sum_{o \in \text{outputs}} p(o, v) \log_2 p(o, v) \\ &= \frac{2}{8} \left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2} \right) + \frac{4}{8} \left(\frac{3}{4} \log_2 \frac{3}{4} \frac{1}{4} \log_2 \frac{1}{4} \right) + \\ &\quad + \frac{2}{8} (0 \log_2 0 + 1 \log_2 1) \\ &= (0.25 \times -1) + (0.75 \times -0.81) + (0.25 \times 0) \\ &= -0.86 \end{aligned}$$

Worked out in this way, the score is always negative. The closer to zero it is, the better. If the score is zero then splitting on that value perfectly separates the outputs.

Continuous data

- What is the input data is continuous rather than categorical?
- In this case, we can split the data up by determining if it is greater to or less than a particular threshold.
- If there are n items of data, then there are $n - 1$ possible thresholds.
- We can calculate the information gain for each threshold and choose the best one.

Potential problems

Branching variables

What if there are variables which are highly branching? (e.g. an ID number which is different in every example)

Data which is rotated

A decision tree splits up the data space into rectangular (cuboid) sections. This might not be very efficient, for example if the data lies at an angle. Rotating the data in different ways could give different results.

Ensembles

- The decision tree method can be made more reliable by using many different decision trees and averaging the results.
- We can make different decision trees by making copies of the training data in which some rows are repeated and some are deleted (the basis of **bootstrap aggregation** or **bagging**).

Pruning

- We normally don't want to build the tree all the way to the end. After some time the
- Pruning is a way of avoiding **overfitting**
- There may be small random variations in the data – we want to capture the essence of the relationship between the variables and the target without including the irrelevant details.
- The normal technique is to learn the full tree and then cut back the parts which do not add much in terms of accuracy.