Advanced Topics in Statistics

$\sim$ *Decision trees* $\sim$

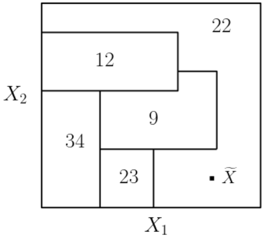
5

UNIVERSITY OF EXETER

Introduction.

Tree-based methods are based on the idea of segmenting the predictor space into a number of simple regions. Then when predicting a new observation, we just take the mean of all the training observations that belong to the same region as this new observation.

- Here we have two predictors and five distinct regions.
- Depending on which region our new $\boldsymbol{X}$ comes from we would make one of five possible predictions for Y. These are the values indicated inside the regions.
- E.g. the observation $\widetilde{X}$ would be assigned the prediction $Y = 22$.



The set of splitting rules can be summarised in a tree (see later), thus we call these learning approaches decision tree methods.

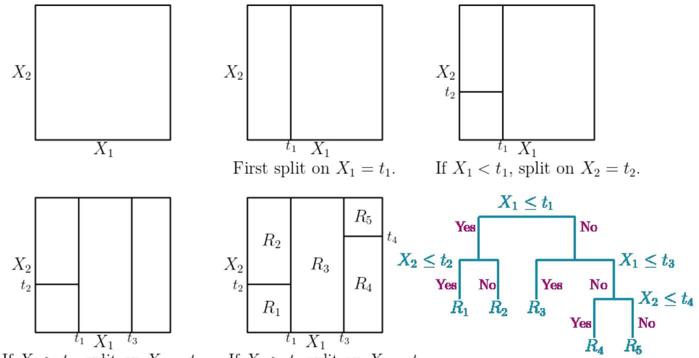
- Tree-based methods have the advantage of easy interpretability, and that the decision rules can be easily explained even to non-experts.
- They can be applied to both regression and classification problems.

Partitioning the predictor space.

One way to make predictions is to divide the predictor space (i.e. all the possible values for $X_1, X_2, \ldots, X_p$) into distinct regions, say $R_1, R_2, \ldots, R_k$.

Then for every X that falls in a particular region (say R_j) we make the same prediction.

Generally we create the partitions by iteratively splitting one of the X variables into two regions.



First split on $X_1 = t_1$.

If $X_1 < t_1$, split on $X_2 = t_2$.

If $X_1 > t_1$, split on $X_1 = t_3$.

If $X_1 > t_3$ split on $X_2 = t_4$.

Regression example - Baseball players' salaries.

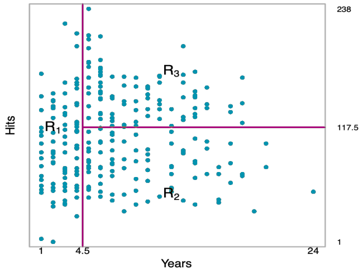
As an example consider the `Hitters` dataset that includes the following information on some baseball players: salary (in thousands of dollars), number of years that he had played in major leagues and the number of hits that he made in the previous year.

We would like to be able to predict the baseball player's (log transformed) `Salary` using the `Years` and the `Hits` variables. Thus we fit a regression tree to the data.

The top split is at `Years=4.5`. For players who have played in major leagues for less than 4.5 years, the predicted (log) salary is the mean of all the training observations for which `Years<4.5`.

Players with `Years>=4.5` are split further depending on the number of hits they made. We make the split at `Hits=117.5`. Thus the regions are:

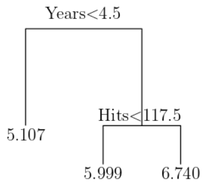
$$R_1 = \{X \mid \texttt{Years<4.5}\}$$
$$R_2 = \{X \mid \texttt{Years>=4.5, Hits<117.5}\}$$
$$R_3 = \{X \mid \texttt{Years>=4.5, Hits>=117.5}\}$$



Baseball players' salaries.

The mean log salary in the first region is 5.107, thus we predict $e^{5.107} = 165.174$ thousands of dollars for these players.

In R_2 the mean log salary is 5.999, while in R_3 it's 6.740, thus we predict $e^{5.999} = 403.026$ and $e^{6.740} = 845.561$ thousands of dollars respectively.



The interpretation of the regression tree is as follows:

`Years` is the most important factor in determining `Salary`.

Less experienced players earn lower salaries than more experienced players. For experienced players the number of hits made in the previous year also affects the salary. Those who made more hits, earn more.

In the tree analogy:

- The regions R_1, R_2 and R_3 are known as terminal nodes or leaves of the tree. (Note that the tree is typically drawn upside down).
- The points along the tree where the predictor space is split are referred to as internal nodes.
- The segments that connect the nodes are the branches of the tree.

How do we construct the regions?

In theory the regions could have any shape, as long as they are not overlapping.

But for simplicity, and for easy interpretability, decision trees create high-dimensional rectangles, or boxes.

For regression trees, the aim is to find boxes $R_1, \ldots, R_J$ that minimise the MSE,

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2,$$

where $\hat{y}_{R_j}$ is the mean response for the training observations within the jth box.

It is however computationally infeasible to consider every possible partition of the feature space into J boxes.

For this reason we use a so-called top-down greedy approach, also known as recursive binary splitting.

- *Top-down*, because we start with all observations being in one region, and then successively splits the predictor space up. Each split goes a step deeper in the tree.
- *Greedy*, because at each step we make the split that's best at that particular step, without thinking ahead, that what might lead to a better tree further down the line.

RECURSIVE BINARY SPLITTING.

When splitting the tree we consider all the predictors $X_1, \ldots, X_p$ and all the possible cutpoints s for each predictor, and choose the pair (X_j, s), where a split would give the greatest decrease in MSE for the training data.

That is, we seek j, s such that if

$$R_1(j,s) = \{\boldsymbol{X} | X_j < s\} \quad \text{and} \quad R_2(j,s) = \{\boldsymbol{X} | X_j \geq s\}$$

then

$$\sum_{i:\boldsymbol{x}_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\boldsymbol{x}_i \in R_2(j,s)} (y_i - \hat{y}_{R_1})^2$$

is minimised. Here $\hat{y}_{R_1}$ and $\hat{y}_{R_2}$ are the mean responses for the training observation in region $R_1(j,s)$ and $R_2(j,s)$ respectively.

Then we repeat the process looking for the next best split, except that we must also consider whether to split the first region or the second region up. Again, the criteria is smallest MSE.

This results in three regions. Next we split one of these further, and so on.

The process continues until a stopping criterion is reached, such as our regions have too few observations to continue e.g. all regions have 5 or fewer points.

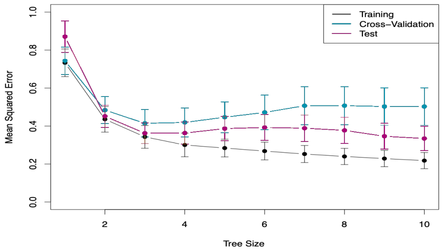
PRUNING THE TREE.

A large tree (i.e. one with many terminal nodes) tend to overfit the training data, leading to poor test set performance, while a smaller tree with fewer splits (that is, fewer regions) might lead to lower variance and better interpretation at the cost of a little bias.

It is usually not a good idea to have a MSE dependent stopping criteria for the tree-building process, since a seemingly worthless split can be followed by one that dramatically decreases the MSE.

Instead we build a very large tree, and improve accuracy by 'pruning' this tree i.e. cutting off some of the terminal nodes.

How do we know how far back to prune the tree? We use a version of cross validation to see which tree has the lowest error rate.



E.g. The `Hitters` dataset has further variables.

Using nine of these and using cross validation to prune the resulting tree, shows that the minimum cross validation error occurs at a tree size of 3 ((i.e. the number of leaf nodes is 3, which is the tree we built earlier).

Classification trees.

A classification tree is very similar to a regression tree except that we try to make a prediction for a categorical rather than continuous Y.

For each region (or node) we predict the most commonly occurring category among the training data within that region.

The tree is grown in exactly the same way as with a regression tree (i.e. recursive binary splitting) except that minimising the MSE no longer makes sense.

A natural alternative is using the classification error rate, which in this case is simply the fraction of the training observations in that region that do not belong to the most common class.

The classification error rate is not very effective for tree growing, there are two other measures that are used instead, the 'Gini index' and the 'cross-entropy'. Intuitively, they both measure node purity, i.e. a small value indicates that a node contains predominantly observations from a single class.

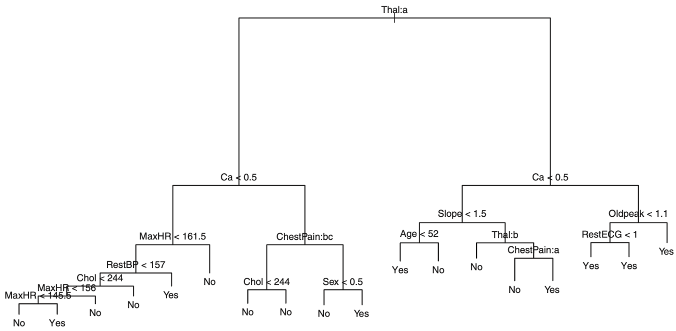
- Gini index and cross-entropy are usually used to evaluate the quality of a particular split in the tree growing process.
- Classification error rate is preferred for pruning, prediction accuracy of the final pruned tree is the goal.

HEART DATASET EXAMPLE.

The `Heart` dataset contain a binary outcome `HD` for 303 patients who presented with chest pain. An outcome value of `Yes` indicates the presence of heart disease based on an angiographic test, while `No` means no heart disease.

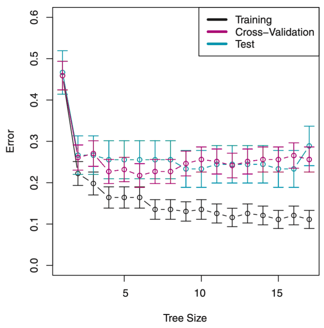
There are 13 predictors including `Age`, `Sex`, `Chol` (a cholesterol measurement), and other heart and lung function measurements.

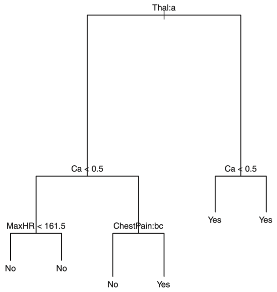
Without pruning we get the following decision tree.



HEART DATASET PRUNING.

Upon pruning the tree, cross-validation results in a tree with six terminal nodes:





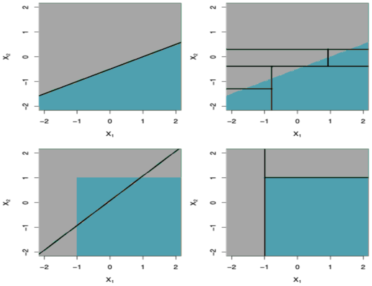
Note that decision trees can also handle qualitative predictor variables (e.g. the `ChestPain` variable above) quite easily. When doing a split on this branch we just assign some of the qualitative values to one branch and the rest to the other branch.

E.g. the `ChestPain:bc` split assigns the second and third values of the `ChestPain` variable to the left branch, where the possible values are typical angina, atypical angina, non-anginal pain, and asymptomatic

Tree vs Linear models.

Which model is better?

- If the relationship between the predictors and response is linear, then classical linear models such as linear regression would outperform regression trees.
- On the other hand, if the relationship between the predictors is non-linear, then decision trees would outperform classical approaches.
- In certain cases, regardless of the test error, prediction using a tree may be preferred for the sake of interpretability and visualisation.



The top row shows an example where the true decision boundary is linear, in which case the linear model outperforms the decision tree.

The bottom row shows an example where the true decision boundary is non-linear. Here the decision tree has a better performance.

Pros and Cons of decision trees.

Decision trees have many advantages:

- Trees are very easy to explain to people (probably even easier than linear regression).
- Some people believe that decision trees more closely mirror human decision-making than do the previously discussed regression and classification approaches.
- Trees can be plotted graphically, and are easily interpreted even by non-expert.
- They work fine on both classification and regression problems.
- Trees can easily handle qualitative predictors without the need to create dummy variables.

However they don't have the same prediction accuracy as some of the more complicated approaches.

The solve the problem of prediction accuracy we can aggregate many decision trees, by using methods like bagging and random forests.