

# Chapter 8: Tree-Based Methods

→ nonparametric supervised methods.

We will introduce *tree-based* methods for regression and classification.

These involve segmenting the predictor space into a number of simple regions.

to make a prediction for an observation, we use mean or mode of training observations in the region to which it belongs.

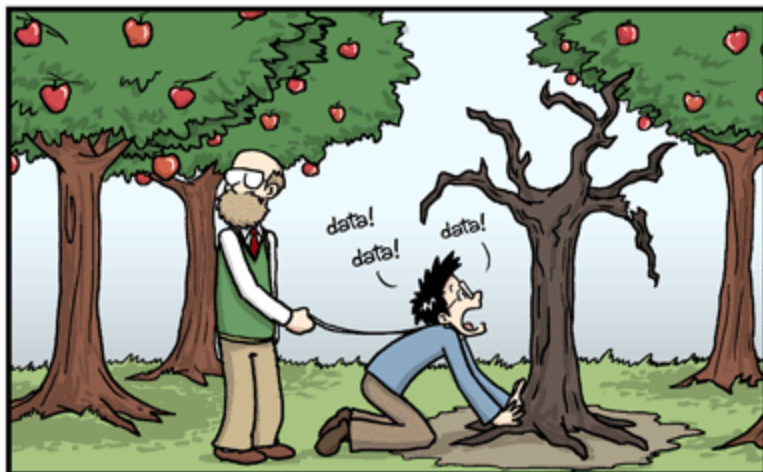
→ quantitative  $y$       → categorical  $y$ .

The set of splitting rules can be summarized in a tree ⇒ “decision trees”.

- simple and useful for interpreting.
- not competitive w/ other supervised approach (e.g. lasso) for prediction.

Combining a large number of trees can often result in dramatic improvements in prediction accuracy at the expense of interpretation.

→ boosting!  
bagging!  
random forests

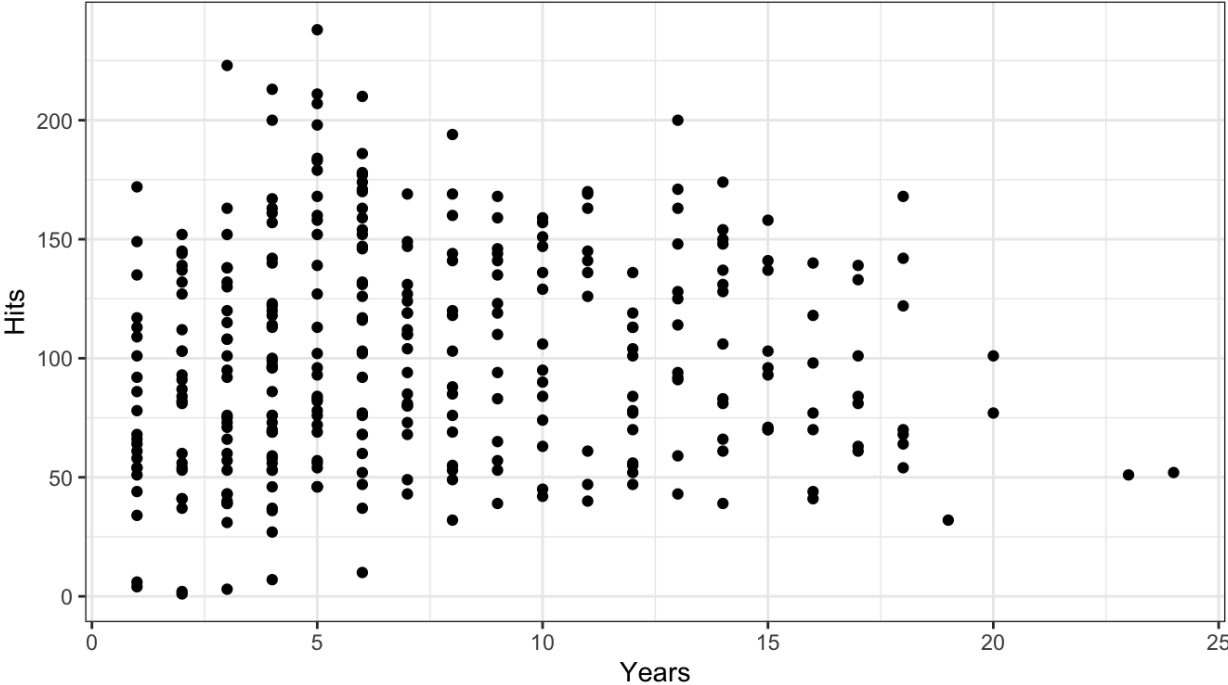


Credit: <http://phdcomics.com/comics.php?f=852>

Decision trees can be applied to both regression and classification problems. We will start with regression.

# 1 Regression Trees

**Example:** We want to predict baseball salaries using the `Hitters` data set based on `Years` (the number of years that a player has been in the major leagues) and `Hits` (the number of hits he made the previous year).



The predicted salary for players is given by the mean response value for the players in that box. Overall, the tree segments the players into 3 regions of predictor space.

We now discuss the process of building a regression tree. There are two steps:

1.

2.

How do we construct the regions  $R_1, \dots, R_J$ ?

The goal is to find boxes  $R_1, \dots, R_J$  that minimize the RSS.

The approach is *top-down* because

The approach is *greedy* because

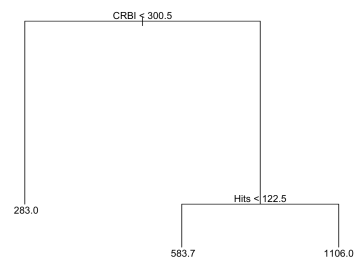
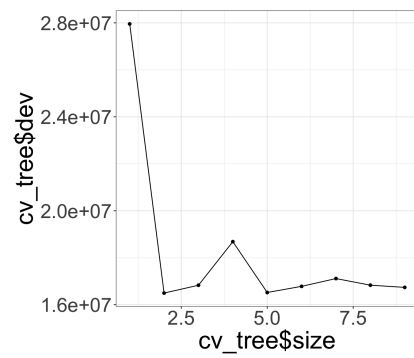
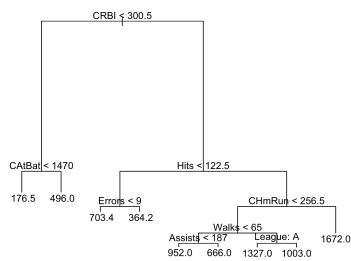
In order to perform recursive binary splitting,

The process described above may produce good predictions on the training set, but is likely to overfit the data.

A smaller tree, with less splits might lead to lower variance and better interpretation at the cost of a little bias.

A strategy is to grow a very large tree  $T_0$  and then *prune* it back to obtain a *subtree*.

Algorithm for building a regression tree:



## 2 Classification Trees

A *classification tree* is very similar to a regression tree, except that it is used to predict a categorical response.

For a classification tree, we predict that each observation belongs to the *most commonly occurring class* of training observation in the region to which it belongs.

The task of growing a classification tree is quite similar to the task of growing a regression tree.

It turns out that classification error is not sensitive enough.

When building a classification tree, either the Gini index or the entropy are typically used to evaluate the quality of a particular split.

## **3 Trees vs. Linear Models**

Regression and classification trees have a very different feel from the more classical approaches for regression and classification.

Which method is better?

### **3.1 Advantages and Disadvantages of Trees**

# 4 Bagging

Decision trees suffer from *high variance*.

*Bootstrap aggregation* or *bagging* is a general-purpose procedure for reducing the variance of a statistical learning method, particularly useful for trees.

So a natural way to reduce the variance is to take many training sets from the population, build a separate prediction model using each training set, and average the resulting predictions.

Of course, this is not practical because we generally do not have access to multiple training sets.



While bagging can improve predictions for many regression methods, it's particularly useful for decision trees.

These trees are grown deep and not pruned.

How can bagging be extended to a classification problem?

## 4.1 Out-of-Bag Error

There is a very straightforward way to estimate the test error of a bagged model, without the need to perform cross-validation.

## 4.2 Interpretation

# 5 Random Forests

*Random forests* provide an improvement over bagged trees by a small tweak that decorrelates the trees.

As with bagged trees, we build a number of decision trees on bootstrapped training samples.

In other words, in building a random forest, at each split in the tree, the algorithm is not allowed to consider a majority of the predictors.

The main difference between bagging and random forests is the choice of predictor subset size  $m$ .

# 6 Boosting

*Boosting* is another approach for improving the prediction results from a decision tree.

While bagging involves creating multiple copies of the original training data set using the bootstrap and fitting a separate decision tree on each copy,

Boosting does not involve bootstrap sampling, instead each tree is fit on a modified version of the original data set.

Boosting has three tuning parameters:

1.

2.

3.