Regression Tree Issues

Interpretation

- Regression trees have **no interpretation** like linear regression
- It is **computationally infeasible** to consider every possible data partition, especially with large samples.
- Generally, they tend to be used when **predictive accuracy** is the goal, rather than interpretation, and the **cross-validation error** is minimized.
- But, **linear regression** methods also often **outperform** regression **trees** in terms of predictive accuracy
- So, why bother with regression trees?

Regression Tree Methods

- Plain regression trees are easy to fit and provide **quick** predictive results; but **linear regression** provides more sophistication, modeling **flexibility** and **interpretability**.
- If the modeling **goal** is not interpretability, but **predictive accuracy**, regression trees should be tried among the pool of methods and the results evaluated to select the **most accurate** predictive model
- There are **complex** variations of the regression tree methods that **sometimes outperform** linear regression in **test accuracy**
- Some of these are:
  - ✓ Bootstrap Aggregation (**Bagging**)
  - ✓ **Random Forests**
  - ✓ **Boosting**
- We will discuss these later in the context of **classification** trees, but the principles and methods are similar for regression trees.

R Tips

`tree(){tree}` → Function in the `{tree}` package to fit regression and classification trees

`regtree.fit=tree(y~x1+x2+etc.,data=dataName)` → Fits a regression tree when y is a continuous value variable

`plot(regtree.fit)` → Plots the `regtree.fit` tree object

`text(regtree.fit,pretty=0)` → To add labels and text to the tree

`cv.regtree=cv.regtree(regtree.fit)` → Computes the CV for various pruned trees

`plot(cv.regtree$y,cv.regtree$dev,type='b')` → Plot y vs. deviance

`prune.regtree.fit=prune.tree(regtree.fit,best=5)` → Prunes the `regtree.fit` object to 5 terminal nodes

KOGOD SCHOOL
of
BUSINESS