Classification and Regression Trees

Charlotte Wickham

March 16, 2007

1. Growing trees
2. Pruning trees
3. Supernova

Trees

- Restrict to binary splits
- Computationally infeasible to build every possible tree
- Want a algorithm that builds a “good” tree.

Trees

Two components:

- Growing trees
- Pruning trees

Basic Idea

Growing trees:

- Choose the "best" possible binary split of the data
- Take each side of the split and find the next "best" split for each side.
- Continue until either each terminal node is "pure" or of some minimum size.

Will result in an over-fitted tree. Then need to prune tree:

- Using large tree prune back to find "best" tree of various sizes
- Use cross-validation to choose "best" size.

Growing trees

Need four things:

- A set of possible splits
- A measure of how good a split is
- A stop-splitting rule
- A rule for assigning a terminal node to a class.

Growing trees - Possible splits

Only split on one variable:

- Binary variables
  - one possible binary split
- Categorical variables
  - Splits of the form $x \in S$ where S could be any possible subset of the categorical levels.
  - $2^{L-1} - 1$ possible splits.
- Continuous or ordinal variables
  - Splits of the form $x \leq x_c$.
  - At most N possibilities - where x_c is halfway between to consecutive distinct values.

Need a measure to choose which spilt is best.

Growing trees - Best split

Introduce an impurity function f.
Let the impurity of a node t be,

$$I(t) = \sum_{i=1}^{C} f(p_{it}),$$

where p_{it} is the proportion of those in t that belong to class i in future samples (in practice use the proportions in the learning set possibly times a prior).

- Want $I(t)$ to be maximal when node contains equal amounts of each class.
- Want $I(t)$ to be minimal (i.e. 0) when node contains only one class.

Growing trees - Best split

Given a split, s, that sends a proportion p_R of the data to t_R and p_L to t_L the decrease in impurity from the split is,

$$\Delta I(s,t) = I(t) - p_R I(t_R) - p_L I(t_L)$$

Want to choose a split that maximises this decrease.
Some examples of f:

- Gini index $f(p) = p(1-p)$
- Information index $f(p) = -p\log p$

Growing trees - Stopping rule & assigning classes

Stopping rule

- Want a large tree - since we are going to prune later
- Keep growing until either terminal nodes are very small or are pure.

Assigning classes

- Assign to the class with the largest p_{it}

Pruning

- We now have a large tree which will have likely over-fitted the data. We want to prune back to a smaller tree.
- Basically, define a cost-complexity measure that is the misclassification cost of a tree penalized by its complexity.
  - cost-complexity of T = misclassification rate of $T + \alpha|T|$.
  - $|T|$ is the number of terminal nodes and measures the complexity of tree T.
- Find T that minimises the cost-complexity for various α.

Turns out this defines a sequence of nested subtrees of our original large tree.

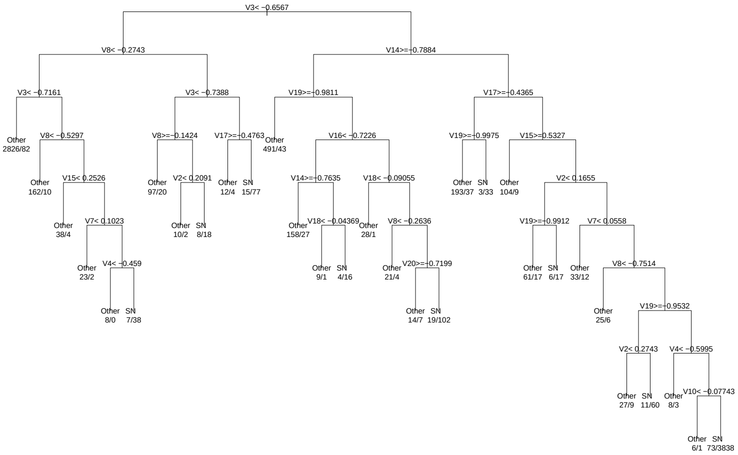
Choosing the level of pruning

- How do we choose the best sized tree from the sequence generated?
- Want to minimise the misclasification rate
- Misclassification rate on the training data will always decrease with increasing tree size
- Need an estimate if the misclassification rate for new data
- Cross validation!

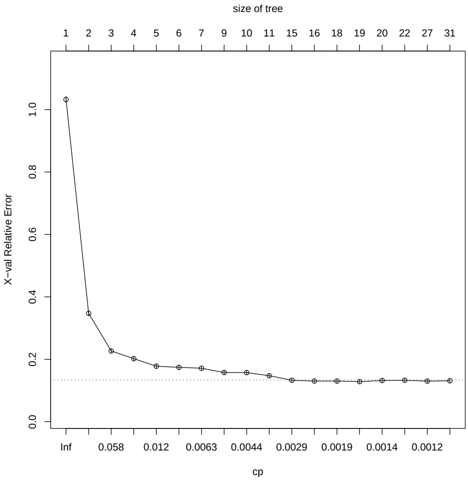
Supernova Data

- 5000 Supernova and 5000 other objects
- Split into two sets 9000 in training 1000 in test
- Build tree and prune based on training set using cross validation
- Test prediction on test set and compare to support vector machines.

Growing full length tree

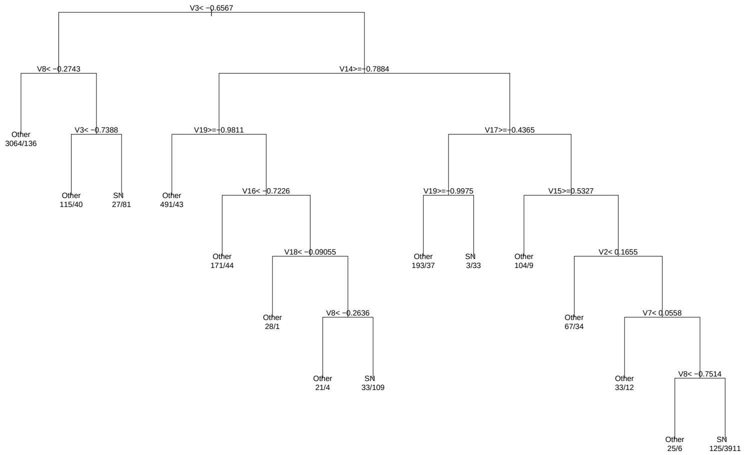


Pruning



- Look for smallest tree within 1sd of tree with minimum error.
- About 14 splits.

Pruned tree


V3< −0.6567
V8< −0.2743
V14>=−0.7884
Other
3064/136
V3< −0.7388
V19>=−0.9811
V17>=−0.4365
Other
115/40
SN
27/81
Other
491/43
V16< −0.7226
V19>=−0.9975
V15>=0.5327
Other
171/44
V18< −0.09055
Other
193/37
SN
3/33
Other
104/9
V2< 0.1655
Other
28/1
V8< −0.2636
Other
67/34
V7< 0.0558
Other
21/4
SN
33/109
Other
33/12
V8< −0.7514
Other
25/6
SN
125/3911

Left branch

Performance on test set

- Classification Tree

		Prediction: Other	Prediction: Supernova
Actual	Other	461	39
	Supernova	45	455

Error = 8.4%

- Best Support Vector Machine

		Prediction: Other	Prediction: Supernova
Actual	Other	484	16
	Supernova	32	468

Error = 4.8%

Other things to look into

- Try different impurity measures
- Using linear combination of variables as splits
- Priors?

L. Breiman, J. Friedman, R. Olshen, and C. Stone.
Classification and Regression Trees.
Wadsworth and Brooks, Monterey, CA, 1984.

Brian D. Ripley and N. L. Hjort.
Pattern Recognition and Neural Networks.
Cambridge University Press, New York, NY, USA, 1995.
ISBN 0521460867.

Terry M Therneau and Beth Atkinson. R port by Brian Ripley ¡ripley@stats.ox.ac.uk¿.
rpart: Recursive Partitioning, 2006.
URL `http://mayoresearch.mayo.edu/mayo/research/biostat/splusfunctions.cfm`.
R package version 3.1-32.