Classification and Regression Trees

Charlotte Wickham

March 16, 2007

1. Details from last week
   - rpart
   - Priors and Costs
   - Regression Trees
2. Bagging

Splits on two variables classified a lot of the “Other” objects. They were:

- perinc (V3) - % flux increase in aperture from REF to NEW
- neighbordist (V8) - distance to the nearest object in REF

rpart

- Does not use balanced cross-validation sets.
- rpart's cp parameter

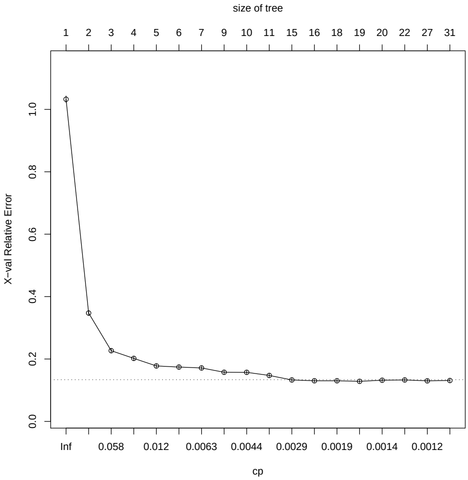
$$R_{cp}(T) = R(T) + cp \times |T| \times R(T_0)$$

  where T_0 is the tree with no splits. (Recall $R(T) =$ misclassification rate of tree T.)
- in CART

$$R_\alpha = R(T) + \alpha \times |T|.$$

  So the tree than minimizes R_{cp} is the same as that which minimizes $R_{\alpha = cp \times R(T_0)}$.

Pruning


size of tree
1
2
3
4
5
6
7
9
10
11
15
16
18
19
20
22
27
31
X−val Relative Error
0.0
0.2
0.4
0.6
0.8
1.0
Inf
0.058
0.012
0.0063
0.0044
0.0029
0.0019
0.0014
0.0012
cp

Priors and Costs

May have prior probabilities for each class $\pi_i, \quad i = 1, \ldots, C$.
May also have cost for misclassifying an i as a j, $L(i, j)$.
Where do these enter the growth and pruning of trees?

- In splitting
- In pruning
- In class assignment
    - Now terminal nodes are assigned to class i where i minimizes

$$\sum_j C(i|j)p(j|t).$$

Priors

Have $\pi_i, \quad i = 1, \ldots, J \quad \sum_{i=1}^{J} \pi_i = 1$. Let $N_j(t)$ be the number of cases of class j falling into node t. Then,

$$p(j, t) = \pi(j) N_j(t) / N_j$$

is the estimate for the probability of a case will be type j and fall into node t.

$$p(t) = \sum_j p(j, t)$$

is the estimate for the probability that any case will fall into node t.

$$p(j|t) = p(j, t) / p(t)$$

is the estimate for the probability of a case will be type j given it falls into node t.

Priors in splitting

Let the impurity of a node t be,

$$I(t) = \sum_{i \neq j} p(i|t)p(j|t), \text{ (Gini index).}$$

And we are choosing split, s, to minimise,

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

where $p_R = p(t_R)/p(t)$ and p_L similarly.
So, the priors affect splitting by simply adjusting the estimated proportions ($p(i|t)$, p_R, etc) needed to calculate the impurity.
Note the classification cost doesn't come in here.

Including misclassification costs in splitting

Two methods. First,

- Generalized Gini impurity

$$I(t) = \sum_{j,i} C(i|j)p(i|t)p(j|t),$$

  - Only depends on symmetrized cost matrix
  - Might not be concave in $\{p(j|t)\}$ so decrease in impurity could be negative.

Including misclassification costs in splitting

Second,

- Altered priors
  - Priors and costs are somewhat interchangeable
  - Let $Q(i|j)$ be the proportion of class i classified as class j by tree T. Then the estimate of misclassification is

$$R(T) = \sum_{i,j} C(i|j)Q(i|j)\pi(j)$$

  - Can find $C'(i|j)$ and $\pi'(j)$ such that R(T) doesn't change.
  - In the case that $C(i|j) = C(j) \quad i \neq j$, let $C'(i|j)$ be unit costs and define

$$\pi'(j) = C(j)\pi(j) / \sum_j C(j)\pi(j)$$

  - Use these altered priors as per usual.

Pruning

Cost-complexity pruning

- Choose tree T that minimizes,

$$R_\alpha(T) = R(T) + \alpha|\widetilde{T}|,$$

  $|\widetilde{T}| =$ the number of terminal nodes of T.
- Priors and costs come into definition of R(T), the misclassification cost of T (or the risk).

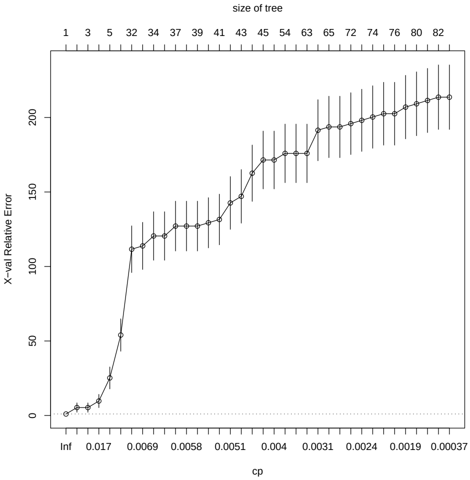
$$R(T) = \sum_{t \in \widetilde{T}} R(t)$$

$$\begin{aligned} R(t) &= r(t)p(t) \\ &= \left(\min_i \sum_j C(i|j)p(j|t) \right) p(t) \end{aligned}$$

Costs and priors in supernova data

- Raquel says more than 10,000 negatives for every positive.
- so try prior $P(\text{sn}) = 1/10000$ $P(\text{other}) = (10000 - 1)/10000$.
- no cost
- Didn't work so well

Priors in supernova data

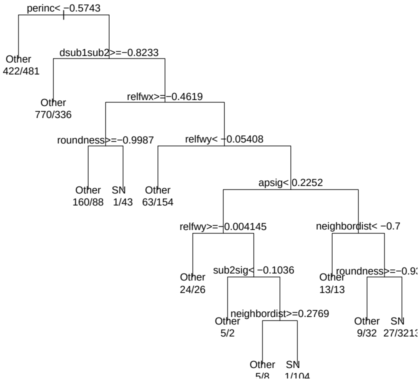


- Error gets bigger as tree grows?

Costs and priors in supernova data

- Try something a little more moderate.
- Still no good if $P(\text{sn}) = 1/100$ $P(\text{other}) = 99/100$.
- So try prior $P(\text{sn}) = 1/10$ $P(\text{other}) = 9/10$.

Priors in supernova data



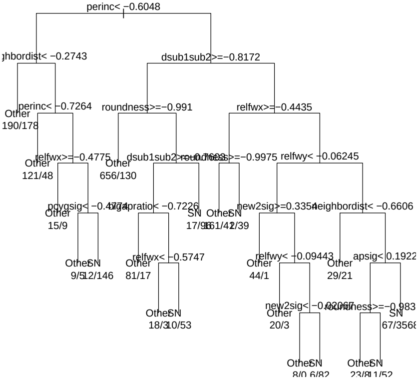
		Prediction	
		Other	SN
Actual	Other	494	6
	SN	142	358

Total error = 14.8%
+ve's = 28.4%
-ve's = 1.2 %

(Compare with
Total error = 8.4%
+ve's = 9%
-ve's = 7.8 %
when no priors.)

Priors in supernova data

Also try, $P(\text{sn}) = 3/10$ $P(\text{other}) = 7/10$.

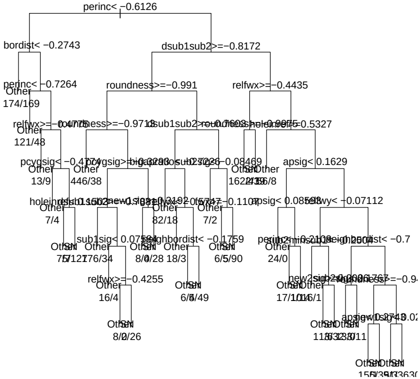


		Prediction	
		Other	SN
Actual	Other	470	30
	SN	59	441

Total error $= 8.9\%$
$+$ve's $= 11.6\%$
-ve's $= 6\%$

Costs in supernova data

Try costs instead: $C(SN|Other) = 2, C(Other|SN) = 1$



		Prediction: Other	Prediction: SN
Actual	Other	476	24
Actual	SN	55	445

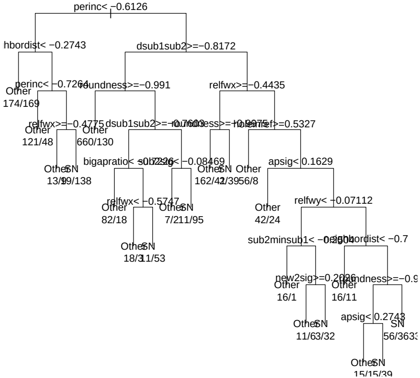
Total error $= 7.9\%$

$+$ve's $= 11\%$

-ve's $= 4.8\%$

Costs in supernova data

Also try: $C(SN|Other) = 3, C(Other|SN) = 1$



		Prediction	
		Other	SN
Actual	Other	475	25
	SN	56	444

Very similar to previous.

Regression Trees

- Simpler than classification trees since the same measure can be used for splitting and pruning.
- Now learning sample consists of $(y_1, \boldsymbol{x_1}), \ldots, (y_n, \boldsymbol{x_n})$. For new $\boldsymbol{x'}$ we want to predict y'.
- Define our error measure (equivalent to impurity and misclassification measures in classification) as

$$R(t) = \frac{1}{N} \sum_{x_n \in t} (y_n - \bar{y}(t))^2$$

  where $\bar{y}(t)$ is the average of all cases in node t.
- Like the sum of squares within a node.

Regression cont...

- Then similarly to the classification case, choose split s which maximises

$$\Delta R(s,t) = R(t) - R(t_L) - R(t_R)$$

- Again, grow a large tree and then prune back.
- Cost complexity measure,

$$R_\alpha(T) = R(T) + \alpha|\widetilde{T}| \quad \text{where } R(T) = \sum_{t \in \widetilde{T}} R(t)$$

- Find subset of trees that minimise R_α for various α. Choose good α by cross validation.
- Terminal nodes are assigned the value $\bar{y}(t)$ the average value of the cases in the node.

Least deviation regression

Previous slides describe least squares regression trees. Alternatively, define

$$R(t) = \frac{1}{N} \sum_{x_n \in t} |(y_n - \nu(t))|$$

where $\nu(t)$ is any median of the cases in node t. Then terminal nodes are assigned the value $\nu(t)$. Can be less sensitive to outliers.

Diamond Data

Measurements of 53935 diamonds and their selling price.

Variable	Description
price	Selling price
carat	weight of the diamond
cut	Fair, Good, Very Good, Premium, Ideal.
color	graded on a letter scale from D to Z. Only D-J in this dataset.
clarity	From good to bad:IF, VVS1, VVS2, VS1, VS2, SI1, SI2, I1.
totaldepth	
table	
width	Physical dimensions
height	
depth	

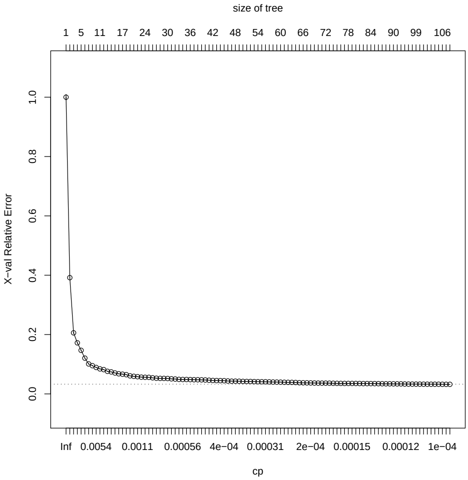
Best Linear Model

```
Call:
lm(formula = log(price) ~ carat + cut + color + clarity + totaldepth +
    table + width + height + depth, data = dia.train)

Residual standard error: 0.1353 on 48502 degrees of freedom
  (16 observations deleted due to missingness)
Multiple R-Squared: 0.9822,     Adjusted R-squared: 0.9822
F-statistic: 1.165e+05 on 23 and 48502 DF,  p-value: < 2.2e-16
```

Look at MSE on test set for `price`. MSE (on original scale) = 309308.8

Regression Tree for diamonds data



- Decreasing very quickly at first and then very slowly
- Try a couple of prunes and check out their performance.
- Prune to size 99 MSE = 218485.4
- Prune to size 50 MSE = 282743.0
- Both do better than regression.

Bagging

Idea:

- Have a sequence of learning sets $\{\mathcal{L}_k\}$ that each contain N independent observations from the same underlying distribution.
- On each learning set we can build a predictor $\phi(x, \mathcal{L}_k)$.
- We want to combine these into one predictor.
- Take averages for continuous ouptut or voting for categorical output.
- But we don't generally have a set of learning samples.
- Make one using bootstrap replicates
- bootstrap aggregating = bagging

Bagging for trees

1. A bootstrap sample, $\mathcal{L}_B$, from $\mathcal{L}$ is selected.
2. A tree is grown on $\mathcal{L}_B$ (and $\mathcal{L}$ is used to choose a pruned subtree).
3. This is repeated K times to give a sequence of predictors, $\phi_1(x), \ldots, \phi_K(x)$.
4. The bagged predictor of, y_n, is $\text{avg}_k \phi_k(x_n)$ for regression trees or is the class having the plurality in $\phi_1(x), \ldots, \phi_K(x)$ for classification trees.

Note: Bagging isn't restricted to trees.

- Bagging works well for unstable predictors
    - Trees
    - Neural Networks
    - Subset selection in linear regression
- Can do worse than the base predictor for stable predictors
    - Nearest Neighbour methods

Bagging Supernova data

		Prediction	
		Other	SN
Actual	Other	477	37
	SN	23	463

Total error = 6%
+ve's = 11.6%
-ve's = 4.6%

Bagging Diamonds Data

MSE $= 689353$

- A lot worse!

Things to find out

- Variable Importance
- Visualising bagged predictors

regression - stratifying cross validation

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.