A Comparison of PRIM and CART for Exploratory Analysis

With Application to the Insurance Industry

Charlie Pollack B Ec. F.I.A.A.
The objective of this presentation is to discuss the relative merits of CART and PRIM for exploratory analysis.

CART is a well known tree-based modelling tool which builds ‘trees’ by recursively splitting that data. Each split is performed on a proportion (on average, half) of the data from the previous split. Hence, as the tree builds, smaller and smaller nodes of data are split. Each split is on a single factor, selected in such a fashion as to optimise a selected splitting rule.

PRIM is not really a tree based modelling tool, but the resulting structure can be presented in a sort of tree fashion. Indeed, at each step in the model, a rule is developed which is by nature a complex rule involving a number of variables. Only a small proportion of the data is ‘peeled away’ from the main body of data by each rule. The remainder is then used to develop the next rule.

In this instance, the example of Insurance Quote data is used to illustrate the ways in which these two tools can be used to investigate the existence of patterns or features in the data.
Insurance Premium Quoting

- Customer Information (age, sex, location)
- Vehicle Information (age of vehicle, value of vehicle, several different classifications of vehicle)
- Other policy options and features
- Premium amount charged.
- Some information from new rating structure (eg classification of vehicle, geographic risk, calibration flag).
- Some information from old rating structure (eg classification of vehicle, geographic risk).

The Insurance quote data used for this analysis had a variety of fields (over 25 different fields) in addition to the response variable.

The response variable was a binary indicator of the success or failure to turn the quote in to a sale.
Data

• 4 months of quote data
• First 3 months used for modelling/testing (randomly split 2:1)
• Last month used for Holdout analysis (left aside until final comparison of predictive success).
• Overall success rate 39.7%

1/4 of the data (the most recent month in the time series) was held aside for later comparison.

This left the first 3 months to be used for model/test files. The data was randomly split 2:1 model/test so that the model and test files each had similar proportions of quotes from the first three months.

One-way tables were performed on the data to get some quick ideas of the important variables and to look at the overall sales success rate. Additionally, this allowed the opportunity to check out the valid categories or range of each variable and to ascertain (confirm) the distribution of the quotes within those categories.
CART - Run 1

- Largely default settings, except MINCHILD (set to 100)
- PRIORS EQUAL
- Model/Test - 1 tree
- All variables included except premium

CART was run first in model/test mode.

(Often we would run in exploratory mode and not bother pruning the tree using test data. Such a tree would be grown to its limits and would not give the user any idea of the validity of some of the splits. However, on this occasion as, among other things, the objective was to compare predictive power, the pruning approach was used.)

All the default settings in CART were used (including GINI splitting), with the exception of MINCHILD. This was set to 100 to stop spurious small splits being created.

(The approach we normally use is that we want a child node which, if the average response was achieved, would yield a stable response rate. With a node with fewer than 100 quotes, we would expect to see random variation in the response variable becoming significant and hence making the node an unreliable one.)

Importantly, PRIORS were set to EQUAL for this run - no penalty was put on either value of the response variable.

Every variable was included in the tree building process EXCEPT PREMIUM. Premium was excluded as this variable has, in similar exercises in the past, been found to mask the importance and relationships of other variables. This is due to the way in which values of those variables (such as age of driver) can lead to certain premium outcomes (young drivers = higher premiums).

Whilst the overall level of premium does impact the chance of sales success (the higher the premium, the more likely the customer is to shop around and the more elastic, on a % basis, they will be). However for a given customer (eg age = 19, vehicle value = $100,000, …) the impact of those factors on premium should be similar with other companies and hence the premium outcome is less important. So removing premium from the models means that focus can be on the relative sales success by each predictor.

1 tree was built from the model/test data and investigated.
The first tree that came out had Windscreen Excess Removal option as the primary splitter.

In both child nodes, the next splitter was No Claim Discount. This variable needs some explanation: No Claim Discount is intended to be a discount given to customers who have had good claims experience. Unfortunately, the implementation in the Australian market is flawed in that the steps between each level are too big.

So the difference between the best level and the worst level, in only 6 steps, is a factor of 2.5 (ie the worst level pays 2.5x the premium of the best level).

To add to this, many insurers make it difficult to lose the NCD (whether it be through ‘protection’ or only having certain claims qualify to force a step back) meaning that, over time, a large proportion of the market has drifted towards rating 1 (over 90% in some states).

The result is that customers on the worst level are unfairly treated and the response of most companies now is to give them a more favourable level. Hence, for a customer quoted by this company on the worst level, another company might have given them a more favourable level. The huge leverage of the discount structure then means that, all other things, including the base premium being equal, the competitor will charge a MUCH lower premium. So, for the small number of customers quoted on a low NCD, the response rates are VERY low.

NCD is also a ‘given’ in the pricing structure. As the objective of this exercise was to investigate the sales success of the pricing structure itself, then removing the impact of NCD is desirable. To do this, it was decided to remove the small amount (about 7%) of the data that had a low NCD.

The low NCD business was then considered ‘Node 0’ and treated as the alternate primary splitter and grafted on later.
CART - Run 2

- Remove NCD > 4 from data and rerun.
- ‘Right side’ of tree failed to build further despite size. Highest response rate!
- This group was of particular interest due to its size (37% of data).
- Surely there must be some even higher sub groups within this node?!

With the low NCD portion of the data removed, the tree was rebuilt as per the first run.

The resulting ‘optimal’ tree, as indicated from the test data relative cost, was a surprisingly small tree with only 5 terminal nodes. One node, the right child from the primary splitter, contained 37% of the quotes and had a VERY high response rate. This group was therefore of interest. However CART opted to prune this side of the tree very early on in the pruning sequence.

The left side of the tree was also quite simple. Age was found to be the next splitter on the left side of the primary splitter. The age bands CART selected, even though age was treated as categorical, worked out to be under 29 and over 71 (ie all bands in between went right).

Again, the left side of that split was pruned early in the pruning sequence. It covered 16% of the quotes and had a VERY low response rate.

CART continued to split and find some interesting relationships.
This illustration of the resulting CART tree, with the excluded NCD data shown grafted on with a dotted line, shows how the 6 (including node 0) nodes have very different response rates and are quite sizable.

Like age, scc_c_freq is a ranked categorical variable. There was no ranking constraint placed on it (i.e. it was treated as true categorical instead of real). CART selected adjacent categories (9 and 10) yet did not select the two higher categories (11 and 12) to be in the same group. This was also an interesting split to be further investigated.
CART - Run 3

- Set PRIORS = 1.25 for N class.
- Right Side ONLY builds this time.
- Finds some interesting groups of higher and lower response (but still high relative to the left side of the tree) within that right side.

The large node with 37% of the data and a high response rate was of sufficient interest that we fiddled with the PRIORS to try to get CART to penalise itself for finding ‘No’ cases in the data. A first go was to use 1.25 for the N class.

Coincidentally, (some would say through good application!) in this build ONLY the right side of the tree was built. The left side that was seen before didn’t build at all.

Some interesting groups were also revealed in the right side of the tree - groups which we were seeking - some with very high response rates.
Now have an idea of some high and low response groups to compare to expectations.

It was decided to manually graft this model to the other model to give a combined CART tree (as though CART had found it the first time).
CART - Final

- Join results from Run 2, Run 3 and data excluded after Run 1 to get 'grafted' tree.

The grafted bit from the earlier runs is shown here as shaded.
PRIM focuses on a particular outcome of the target variable at a time. I.e. it looks for high levels of the response variable and the response variable is tweaked such that a 1 is the response of interest. I.e. If we are interested in the N group, N would be response 1 and Y reponse 0.

The first PRIM run was done with response Y as the target.

PRIM was used ‘out of the box’ largely with Beta0 (the minimum support size) set to 7.5%. Alpha (the peeling speed parameter) was set to 0.1 (reasonably slow). Bottom-up pasting was not used to remove unnecessary components of the rules developed.

As PRIM focuses on peeling off relatively small sized boxes with a high level of the target variable, it is very effective at finding hot-spots. The smaller the Beta0 value selected, the smaller the box and more likely that the response of the box found will be higher (further away from the group mean).

That said, there is a trade off with small boxes. Small boxes are usually (but not always) defined by very complex rules requiring a number of variables to be filtered. The smaller the box, the more complex, and harder to grasp, the rule will be. Hence Beta0 was selected with a view to creating boxes that were small, yet would not be overly complicated.

Using Beta0 to control rule complexity is a very crude approach. It is often better, and PRIM allows this, for the user to adjust the rule in the interactive build phase.
The PRIM model here has been presented as a sort of tree however it should be stressed that it is not really a tree based model. It can be clearly seen how small amounts of data are peeled away at each split. Each split is itself a complicated rule (see next slide).

Note the large ‘remainder’ box. This came about when PRIM could not find any more useful box splits.

Indeed, the general response rate of the nodes created reduces towards the grand mean with the remainder, being a little below the grand mean.

An interesting aspect of PRIM boxes is to understand the extent to which each box is similar or dissimilar to another. Variable importance, as reported by CART and some other tools, is not really a direct output of this model, however over or under representation of levels of each variable can be analysed.

Additionally, it is possible to investigate the impact that a variable has on the model by randomising the data in that variable and seeing the impact on the response rates of the boxes created by the rules.
Example PRIM Rule

IF wo = 'Y'
    AND gvehclass NE 'Passenger'
    AND ageband IN ('29-35' '36-45' '46-55' '56-70')
    AND vehgrp IN ('C' 'E' 'L' 'N' 'P' 'Q' 'S' 'X')
    AND afc_c_freq < 'FC11'
    AND fnce NOT IN ('Secured Bank Loan' 'Secured Credit Union Loan')
    AND ageveh > '0000'
THEN box = 1;

This is an example of the components of a single PRIM rule.

Data not meeting this rule’s criteria are left over in the remaining data for developing the next rule.

Hence each rule is conditional on the data also not meeting prior rules. In other words, if NOT box1rule AND NOT box2rule AND vari = ... AND varj IN (...) AND ... THEN box = 3.
PRIM - Target = N

- Switch to N as desired response.
- Otherwise, same settings as for before.
- Finds very LOW response ‘hot spots’.
- Much lower than best found by CART.
- Again, large ‘remainder’ box.

As only the Y cases were being investigated in the first PRIM model, a second was run with the target being the N cases to see if nodes of particularly LOW response could be found.
Again, the rules are presented in a tree form, with the response rate of the boxes gradually converging towards the grand mean.
PRIM - Target Switch

- Alternate between Y and N as target.
- Look for bumps and troughs consecutively.
- Results (rules) somewhat different to manually grafting the two trees.
- Not quite as good on the N group as a pure N target model. (This could be a function of the choice of consecutive switching. There could be a formula for selecting whether a particular split should have target Y or target N to get the best improvement, this was not investigated.)

It was decided to investigate whether one PRIM model could be built covering both the Y and N cases. In this instance, the strategy of alternating between building a rule to find a Y box and a rule to find a N box was selected.

This strategy assumes that consecutive switching is optimal however this is seldom the case. Indeed, a type of ‘improvement’ measure could be envisaged whereby both an optimal N rule and a Y rule are built and the one selected which best improves the model. An automatic rule however would not always be able to deal with the trade off between box size and response rate (ie. Is an N box of, say 5% of the data with a response rate double the grand mean better than a Y box with only 2.5% of the data yet a response rate slightly better than half the grand mean?).
The resulting rules can again be presented in a tree structure again (for education purposes mainly, certainly not reflective any tree structure in the way the models were built).

Here, as can be expected, the response rates converge to the grand mean, albeit from both sides leaving a smaller amount of data in the remainder box.
Comparing on Holdout Data

Each of the main models selected (1 Final CART model and the three PRIM models) is shown here with the boxes, their relative size (in green) and the response rates (blue bars) compared to the grand mean (red line). These charts are prepared on holdout data.

Simplistically, it can be said the PRIM Switching model gives the best prediction however the size of the boxes should be considered. The next chart shows this.
Comparing on Holdout Data

This chart shows the bars from the previous charts overlaid, and the width of the boxes adjusted to reflect the amount of support in each box. This shows the way that the PRIM Y and PRIM Switch models do a very good job of finding a box with 5% of the data and a very high response rate. The Switch model also finds a better group of N responders than the best CART model.

Note however the big ‘flat’ area in the middle for all the PRIM models. This is the remainder box.
Comparing on Holdout Data

Presented as a gains chart, CART model seems to do better over the whole range of the data, particularly in the mid range, however PRIM performs well early and late.
Summary

• Both have advantages.
  – CART very quick to give first view.
  – PRIM can find hot spots more effectively, whilst still being reasonably easy to interpret.
• Both have some disadvantages
  – CART greedy (might not find some groups involving small segments of data).
  – PRIM more time consuming, sometimes rules too complex to interpret easily.
Summary

• Which you choose will depend on your need.

• Suggested approach:
  – Start with CART (to at least get a first cut view)
  – Follow up with PRIM to find hot spots
  – Use PRIM to refine views of groups that CART found.