Institute of Actuaries of Australia

Using Non-Parametric Techniques to Understand Your Data

Prepared by Charles Pollack

Presented to the Institute of Actuaries of Australia
XIV General Insurance Seminar 2003
9-12 November 2003

The Institute of Actuaries of Australia
Level 7 Challis House 4 Martin Place
Sydney NSW Australia 2000
Telephone: +61 2 9233 3466 Facsimile: +61 2 9233 3446
Email: insact@actuaries.asn.au Website: www.actuaries.asn.au

Introduction

On many occasions, actuaries and analysts alike have a need to understand the nuances of the data they are using. The amount of time consumed doing this interrogation is best kept to a minimum. However the approach taken can cause the time spent to differ significantly.

This paper will compare the speed of three different approaches and focus on aspects of the fastest – non parametric techniques. It does not focus greatly on the technical aspects of each method beyond the headline advantages and disadvantages.

The data analysis discussed here is purely one of exploration. That is, understanding how variables are correlated, characteristics of specific sub sections of data and sources of data errors and missing data. Absolute perfection in prediction, or finding techniques that produce the best possible lift is not the focus of this paper.

Approaches to Analysis

There are three general approaches to data exploration where the speed of analysis and time to completion can be compared to crawling, walking and running.

Crawl – One and Two way tables
Walk – interactive analysis (eg SAS/Insight)
Run – Non parametric data mining (eg CART, PRIM)

Obviously, everyone would like to able 'run' straight away. However it is usually necessary to gain experience 'crawling' and 'walking' before the analyst is ready to 'run'. That way the analyst is best placed to know in a particular situation which of the three speeds will yield the best result.

Let's recap the crawl: One, Two and Multi-Way tables.

This is a very widely used method for simply looking at the lay of the data. This could be just analysing frequency of values in each field available for analysis or tables summarising a particular measure, grouped by each value (or range of values) in various fields. To understand the relationship between multiple variables, tables that group the data by combinations of variables can be prepared.

There are two major limitations with preparing tables to understand the relationships between variables. Firstly, the deeper into variable relationships you go (eg 3 or more variables at once), the more data that is required to understand that relationship.

Secondly, and the reason for likening this approach to crawling, is that it is slow. Even if you have a bunch of programs set up to prepare the tables quickly, there is a significant amount of time required to go through each table and understand the relationships that can occur across multiple tables. You would generally start with one way tables, select variables of interest and proceed to two way tables, then three way and so on.

People often use Excel Pivot Tables for 'fast crawling'. However there are still limitations to what can be done with that tool.

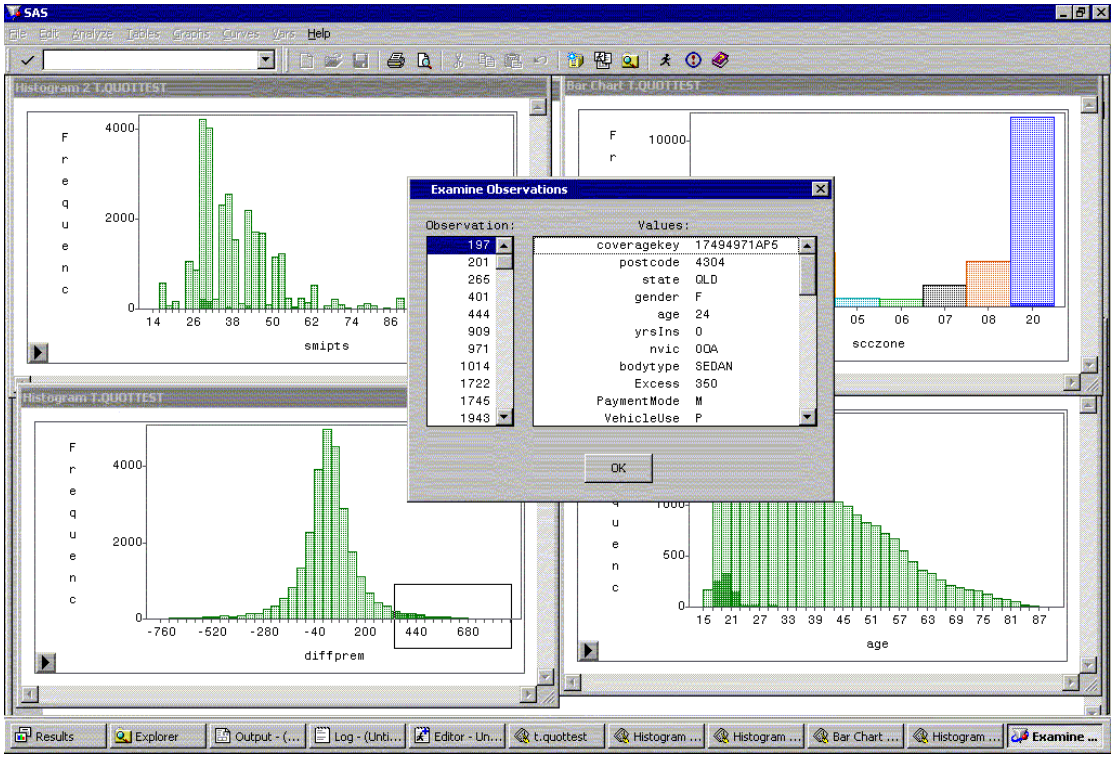
Now for walking: Interactive Analysis with SAS/Insight

SAS/Insight has been around for a long time and has been one of the few modules that has hardly changed between Version 6.12 and Version 9 in SAS. Despite its age, it is still one of the best interactive tools available. You can open a dataset – depending on the power of your machine and number of columns in the dataset, it can manage over 100,000 records – and do a number of interactive charts.

The beauty of the interactive charts is that you can click on data in one chart and see what is highlighted in other charts. This is basically very quick multi-way table analysis. There is no need for

tables to be prepared, new charts can be added with the click of a button and you can click-click-click quickly to see how things change across variables and thereby understand relationships in the data.

Plus, data can be hidden or excluded from analysis quickly and colour-coding can be used. You can also quickly double click to see individual observations behind the charts. There are many other features with which you are probably already familiar.



This all adds up to a pretty speedy way to understand your data. There are limitations however. The ability of your machine to handle large files can sometimes make the process slow (sampling the data can help this though). High order categorical variables (eg postcode, userid, vehicle) can be difficult to analyse and generally need some sort of grouping to be done for the analysis to be tractable – this is no different to one-way tables. And, as with one-way tabulation, investigation of very small cells of data or a response variable with a very low response rate can be quite difficult.

Time to run: Non-parametric techniques – CART

There are lots of non-parametric techniques around – Neural Nets, MART, PRIM, MARS, CART, Random Forests to name a few – and in this discussion we will focus on CART. The key feature of CART (and PRIM) that makes it particularly useful for this analysis is the ease of understanding the resulting model (tree). Neural Nets, MART and Random Forests all suffer from being difficult to interpret (and explain). They might do a very good job of classification, however for exploring data it is necessary to understand the data and its relationships.

CART (Classification And Regression Trees) is not new. The original CART Monograph (Breiman, Friedman, Olshen and Stone) was written in 1984 and commercial software has been available in GUI implementation for about a decade with newer versions well refined and able to be integrated with other data sources (eg SAS datasets). The traditional application of CART is predicting binary response variables such as customer renewal and direct mail responses. There are literally thousands of papers written on these uses. Multi-nomial response variables are also catered for.

The application we are going to investigate here essentially involves identifying a piece of your data (see examples below) and classifying that group as ‘Yes’, with the remainder being the ‘No’ group. CART will then use the variables that you specify from the dataset to predict the ‘Yes’ group. In other words, CART determines which characteristics of the predictor set are related to the Yes group but not the No group. The aim is to create groups (nodes) of pure yes or no. Obviously pure nodes are seldom

found, however this approach leads to nodes that are predominantly yes or no and can hence be classified as one or the other.

The result is a non-linear 'tree' which makes no assumptions about the underlying data structures.

A further discussion of the theory behind CART trees, including how they train, rules for splitting and approaches to model validation are not covered here. An interested reader is directed to the references at the end of this article.

Limitations

The main limitation of CART is that it is a 'greedy' tool. At each split, the data is split into two parts with further splitting done on each of the parts. As a result, each time a split is performed there is less data available than at the previous split. This is a bit like the problem that is faced when looking at multi-way tables. Techniques such as PRIM (Patient Rule Induction Method) do not suffer from this. CART is demonstrated here however due to its accessibility for this particular application.

Features

Important features of CART (in addition to its ease of use and interpretation) are that it:

- can perform well on small datasets;
- is generally fast to model;
- can perform well when the response group of interest is small as a proportion of the overall dataset;
- provides a number of other additional analyses (such as surrogate and competitor variables);
- can deal with missing values in predictor variables;
- is not impacted by outliers;
- can deal with high order categorical variables (such as postcode and userid) - although care must be taken with these;
- works well for exploration using default settings (eg Gini splitting, priors equal, etc) and generally gives a useful tree first go; and,
- starts with no pre-conceived knowledge of the data structure that could colour its view. In other words, constraints such as linearity are not imposed.

Case Studies

1. Dodgy Data Example
2. Who are these people getting big decreases?
3. Mix comparison

1. Dodgy Data Example

This case study covers a real life situation where we were trying to replicate premiums that were being calculated by a mainframe system. We were achieving 95% accuracy after we pulled out the manually overridden premiums. The issue remaining was that we didn't know what was wrong with the last 5% and suspected we had done something wrong in our program. A number of the miscalculated records were manually checked but nothing was obvious.

Over 40 variables were available on the dataset. Some were used for the premium calculation and others were underwriting or information variables. A number of the categorical variables had very high numbers of unique values. These included postcode, vehicle id, client number, policy number and userid.

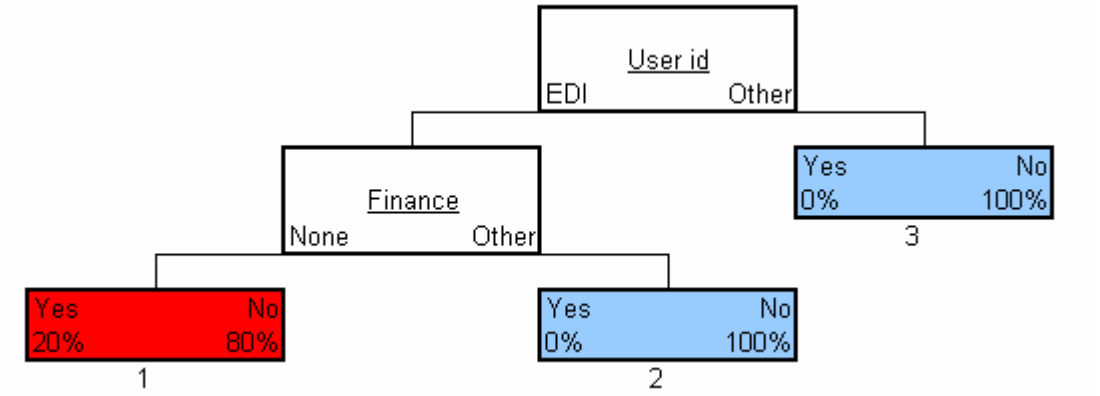
Using the crawl approach – one way tables – is extremely time consuming with high-dimension categorical variables. A decision is usually made to group the values, but in some cases no particular grouping is obvious. Tables are then prepared for each variable calculating the proportion of records for each value of the variable that meet the criteria being investigated. This could lead to pages of tables for a single variable.

In this case, we would calculate the proportion of records where the premium has been miscalculated for each level of the categorical variable being looked at. We could then order the categorical variable

by descending proportion to identify groups of interest. This however is prone to problems where some levels only have one record in the dataset. In this case the % will be either 100% or 0% putting them either at the top or the bottom of the list. So sorting out the true problems from the insignificant problems is very time consuming. Removing the levels with few observations in the file could remove features from the data. Finally, this has to be repeated for each variable. And then two-way tables might be considered adding further elapsed time. A very slow way to move indeed!

Similarly, walking – interactive analysis with the visualisation tool – has problems in this instance with the low response rate. The proportion of errors is so low on the various charts, that visually comparing (say) 2.5% error rate to 7.5% error rate is quite difficult. This approach has a lot of problems visualising high-dimension categorical variables as well.

Pointing CART at the dataset really is like running in this instance – classifying the error records as the 'Yes' group. Give it all the variables in the dataset except the policy number (as every record had its own policy number, this could have been a perfect predictor). Less than 10 minutes later, without tweaking any CART options, we have the answer.



Node 3 was a pure No group (ie. all the calculations were correct).
Node 2 was a pure No group.
All the Yes risks ended up in Node 1, however so did some Nos.

With this information, we then dug around on the mainframe to discover that the problem was with the Finance Code on certain EDI transactions being set to 'N' when the transaction was loaded into the mainframe, when in fact there was Finance on the vehicle. The mainframe always accepted the premium sent by the EDI system (which was calculating the premium with the correct finance code) and so when we recalculated premiums with Finance Code = N – the value stored in the mainframe – we were getting different premiums to the premium stored in the mainframe. (We then arranged for the dodgy 'N' codes in the mainframe to be fixed). All solved in around 30 minutes.

Whilst in this instance the crawling approach, (calculating the proportion of errors for each level of the categorical variable and ranking by descending proportion) would have at least found the primary splitter (Userid=EDI), the process of doing this for each variable would have been lengthy and there could have been a lot of dry gullies run up on other variables before we reached Userid.

2. Who Are These People Getting Big Decreases?

In this example, we were interested in a group of customers getting a large decrease on proposed premium rates. This wasn't just a simple matter of changing values in rate tables. Due to the complexity of the rating structure and the differences between it and the previous structure, it wasn't easy to tell who was getting the decreases. Indeed, the cumulative impact of changes to different rating factors can be quite significant for some customers.

Unlike the previous example, the problem was constrained to just the rating factors as we were comparing a premium basis. This reduced the number of variables of interest to less than 15 meaning that walking or crawling may have been an option. Postcode was still a variable of interest however and it of course has many levels.

Using the crawling approach, a number of key factors were identified, however these were already known and did not add up to explain the very large decreases being witnessed on the records of interest.

With the walking approach, there was no particular structure to the big decreases obvious either. It just reaffirmed the results from the crawling approach.

When we classified the large decreases as 'Yes' and pointed CART at the price change file, within 5 minutes we had a simple tree which summarised the situation and could be used to explain things to management. Indeed, the structure was somewhat more complicated than one, two or even three way tables could have explained and the result was of course much faster to obtain. Certainly as a tool to quickly confirm that the source of extreme price changes is in line with expectations (and not some stuff up), CART is very useful.

3. Mix Comparison

Whilst the examples above have appeared trivial, they generally take considerable time to resolve by crawling or walking. This example, on the other hand, is a bit more complex. In this instance we were interested in the change in mix of new business before and after a price change.

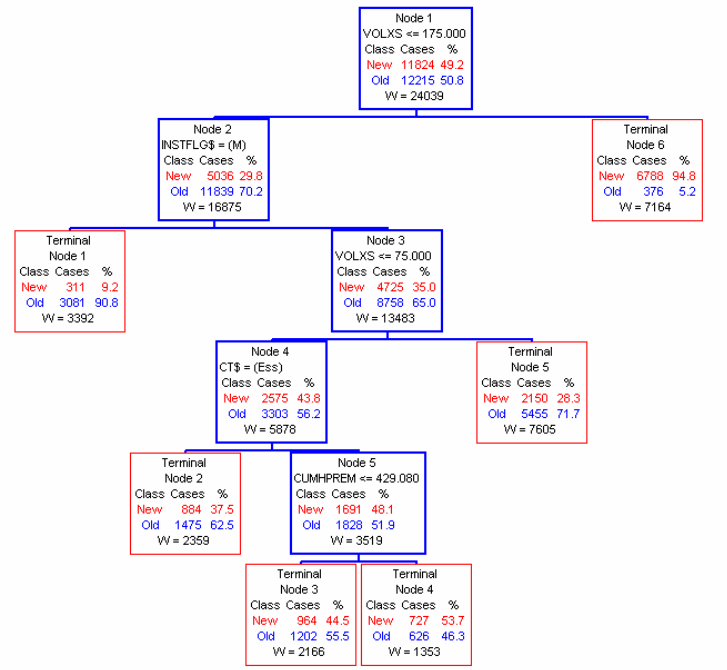
Several months of new business from before the price change and several months of new business from after the price change was extracted from the mainframe system. The month of the price change was left out due to some slight differences, by distribution area, in the actual date the new premiums went live. The old rates group was classified as 'No' and the new rates group was classified 'Yes'.

The crawling approach focused mainly on the major factors and used the usual crude rating buckets for grouping the data. To the extent that the rating structure changes were quite detailed, these high level summarise, whilst identifying some mix changes, did not identify mix changes that were expected as a result of certain factor combinations. Even when preparing two way tables, these changes were not very evident.

Walking through interactive analysis also was difficult as the mix changes were relatively subtle. When looking for movements in proportions around 50% (ie if the mix didn't change 50% would be new and 50% old for each level) small differences can actually mean big change to a portfolio. This is sometimes difficult to pick up by eye.

Running with CART quickly produced a tree summarising the factors that differed between the Yes and No group. For example, on a motor comparison exercise, the Yes group had a lot more young drivers (consistent with a reduction in premiums for that group) as well as a change in the mix of vehicle ages (again, consistent with the price changes put in place). While this is a fairly trivial example which would have been found using the other two approaches, the speed with which the *important* variables could be found (as CART automatically detected them) makes this approach much more direct.

A tree from another similar case, this time involving the mix of Home Insurance written, is shown below.



In the Home Insurance example, the default excess that the call centre consultants used to quote a premium was changed and this was reflected in the tree.

Whilst this seems to be a trivial proof of price or underwriting changes that we already knew, it is actually a very important part of the actuarial control cycle. In fact, when a rate change goes in, it is always important to see if the structural changes that occur in the portfolio are consistent with the assumptions or expectations that were held when the rates were set. Deviation from this expectation can signal a potential problem.

Even when rates haven't changed, it can be useful to compare mix over time. The presence of a particular variable as a predictor of mix difference when that variable has cross subsidies built in (eg No Claim Bonus) can indicate where the portfolio might have a problem. I.e. the cross subsidy might be being eroded requiring remedial pricing action.

The use of CART in this instance really speeds up the process of comparing mix over time and, importantly, identifying the factors of interest. Other techniques, whilst being able to yield the same result, are generally much slower and at times do not yield the same result if the user is not persistent. In fact, because CART has no preconceptions about which variables are important and which aren't, it can sometimes identify things which are completely unexpected. This is a key advantage of its non-parametric nature.

Conclusion

There are obviously many techniques that can be used to understand the data that you have for modelling. For many practitioners however, the time spent to gain this understanding is crucial. For consultants it can mean the difference between making a profit or a loss on a job. It is really no different for internal practitioners in a company. The fastest techniques that support this analysis are obviously going to be favoured.

Whilst there may be times when it is appropriate to crawl (use one and two way tables) or even walk (interactive data analysis with a data viewer) through a data analysis task, on most occasions running (using a tool such as CART) will be the approach used.

Having a tool at your disposal that makes no assumptions about the data you are analysing is very beneficial. Of course, being able to understand the outputs of that tool is critical and to that end CART is a snack.

Whilst there are many things that can be done to tune a CART tree, the key benefit of CART for this sort of data exploration is how much information can be gleaned from the data without tuning and fiddling with options.

Finally, the speed at which the models can be run, including the fact that for most data types no further preparation of the data is required, makes CART such a useful tool which sets it aside from other, slower to use, techniques. In other words, an experienced user can run through a data analysis task every time, instead of crawling.

A few examples of the application of CART to this analysis have been given here. Essentially the limit is your imagination. Just find the right way to classify the data of interest and start running.

References

Breiman, L. , Friedman, J. , Olshen, R. and Stone, C. (1984) Classification And Regression Trees. Kluwer Academic Publishers.

Hastie,T. , Tibshirani, R., Friedman, J. (2001) The Elements of Statistical Learning. Springer Verlag