Institute of Actuaries of Australia

Beauty Contest for a Line-Up of Models

Prepared by Edward Plowman & John Yick

Presented to the Institute of Actuaries of Australia
XVth General Insurance Seminar 16-19 October 2005

This paper has been prepared for the Institute of Actuaries of Australia's (Institute) XVth General Insurance Seminar 2005. The Institute Council wishes it to be understood that opinions put forward herein are not necessarily those of the Institute and the Council is not responsible for those opinions.

:

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: actuaries@actuaries.asn.au Website: www.actuaries.asn.au

Abstract

Increasing computing power is opening up a wide range of predictive modelling techniques to actuaries, from the fields of traditional statistics and data mining. These techniques are often quite different and difficult to compare directly.

This paper looks at some of the main classes of model available, and compares their performance on a number of real general insurance modelling problems. Performance is measured both in terms of predictiveness, using objective measures – and more subjectively on other qualities that might collectively be described as “elegance”. The models investigated include table-based analysis, generalised linear models, generalised additive models, decision trees, and neural nets.

Keywords: General Insurance, generalised linear / additive models, neural nets, decision trees

1 Introduction

This paper investigates the performance of a number of different predictive modelling techniques on some ‘real world’ insurance problems.

The candidate models are taken from the fields of traditional statistics and from data mining applications, along with some ‘hybrid’ models drawing from both. All the models are, in essence, regression or classification models – i.e. they are aiming to predict a particular outcome (or ‘response’) according to the values of a number of other independent variables. In data mining terms, this is often termed ‘supervised learning’.

The paper uses objective measures to enable a fair comparison of the predictive power of the models. It also offers some commentary on the other relative merits of the models, with reference to criteria such as ease of use, and transparency.

While this paper is not intended as a technical introduction to the modelling techniques used, we have included a description of how the models work in broad terms. We have assumed the reader is reasonably familiar with the concepts and practice – if not the theory – of generalised linear models (GLMs).

2 Meet the Models

2.1 Candidate models

The models tested came from two different ‘schools’. The first of these is traditional statistical and tabular analysis. Generalised Linear Modelling (GLM) is perhaps the most familiar of these analyses to most actuaries.

Our candidate models include more basic forms of analysis, based on summary tabulation:

- Basic one-way analysis (the ‘traditional’ insurance pricing model)
- Two-way analysis

We have then looked at multivariate regression techniques:

- Linear Models
- Generalised Linear Models (in various formulations)

and a further generalisation of these techniques:

- Generalised Additive Models

The second school is that of data mining. Within this, there are two major classes of model that we have investigated – decision trees and neural nets. The decision tree model we have used is CART. With neural nets, we have limited our investigation to a feed-forward network with one hidden layer.

The sections below describe the specific methodologies in broad terms. These are deliberately quite mechanistic, to avoid introducing too much bias in the results from human intervention. As a result, the models fitted are likely to be sub-optimal within their class, but should be more comparable across classes. One of the judging criteria, however, is the degree of improvement that might be achieved through human model selection.

2.2 One-way analysis

We have tabulated the target measure on the training dataset, split on a one-way basis by each available rating factor. Where a rating factor was a continuous variate, we have divided into appropriate bands.

We have then calculated a one-way chi-squared statistic for each rating factor to identify the most significant. We have then selected the most significant variables as the 'pool' of rating factors, although where a rating factor was directly or closely correlated to another, we have only selected one of the factors.

The relativities for each rating factor then became the multipliers for our model. The model intercept was calculated so that the total predicted measure was equal to the actual total for the training data.

2.3 Two-way analysis

From the pool of rating factors selected in (2.2), we have then calculated Cramer's V correlation statistic for each two-way combination of variables.

We have then paired up the two variables with the highest Cramer's V statistic, and calculated the 2-way tabulation of the target measure to give the relativities for those two variables.

We have then paired up the two variables, excluding those already selected, with the highest Cramer's V statistic, continuing to do this until the Cramer's V statistics became too small. The remaining variables were retained in the model using the one-way relativities already calculated.

Again, the model intercept was calculated so that the total predicted measure was equal to the actual total for the training data.

2.4 Linear and Generalised Linear Models

We have fitted GLMs to the training data, using the same selection of variables as in the one-way analysis. We have not attempted to fit any interaction terms or sought to refine the models in any way.

The 'standard' choice of link function and error distribution has been selected in each case, and we have also tested out alternative formulations.

2.5 Generalised Additive Model

A generalised additive model is an extension of GLM.

Despite the name, link functions are still allowed, so the model, expressed in terms of the dependent variable Y, may not be additive in the dependent terms X_i.

The difference is that a GAM relaxes the linearity constraint of a GLM. So, instead of fitting the familiar GLM model structure (ignoring the error term):

$$\text{Link}(Y) = m_0 + m_1X_1 + m_2X_2 + m_3X_3 + \ldots + m_nX_n, \text{ where } m_0 \ldots m_n \text{ are constants}$$

we can fit instead:

$$\text{Link}(Y) = m_0 + f_1(X_1) + f_2(X_2) + f_3(X_3) + \ldots + f_n(X_n)\text{, where } f_1 \ldots f_n \text{ are “smoother” functions}$$

There is not a completely free choice in the f functions; in the implementation we have used, both cubic smoothing splines and LOESS (locally weighted regression splines) were available. Nonetheless, this is a significant increase in flexibility of the form of the function over the linear constraint of a GLM.

(It is true that you can fit a cubic spline with pre-defined knots in a GLM framework, but that is a somewhat different notion from the cubic smoothing splines implemented within GAM.)

For categorical variates (which in the ‘equation’ form shown above reduce to indicator variables $X_i = 0$ or 1), there is effectively no difference between GLM and GAM.

2.6 CART

CART (‘Classification And Regression Tree’) is a decision tree algorithm. Decision trees partition data into ever-decreasing subsets by selecting a ‘splitting rule’, using only the data in the subset to identify which of the available factors gives most discrimination in terms of a particular response.

There are a number of different algorithms, and these differ primarily in:

- the measure of discrimination used to determine splitting rules
- the criteria used to determine when to stop the splitting
- pruning methodologies (used to reduce the complexity of the final tree model, in an attempt to prevent overfitting).

There are some more structural differences between methods, too. CART, for example, will only produce binary splits – i.e. at most two branches at each split. Other methods (e.g. CHAID) produce multi-way splits with one branch for each category of the splitting variable.

We have not sought to compare different decision tree methods in this paper, since there is already plenty of literature available on this comparison. There is no clear best tree algorithm, but we believe that CART acts as a good representative for its class.

On the specific point of binary versus multi-way splits, it is worth pointing out that any multi-way split tree can be exactly reproduced by a binary split tree, but the opposite is not true. On this basis then, a binary split tree such as CART should give a better result.

2.7 (Artificial) Neural Nets

We have used a very simple neural net, a feedforward network with one hidden layer containing 5 nodes.

This can be represented in diagram form as follows:

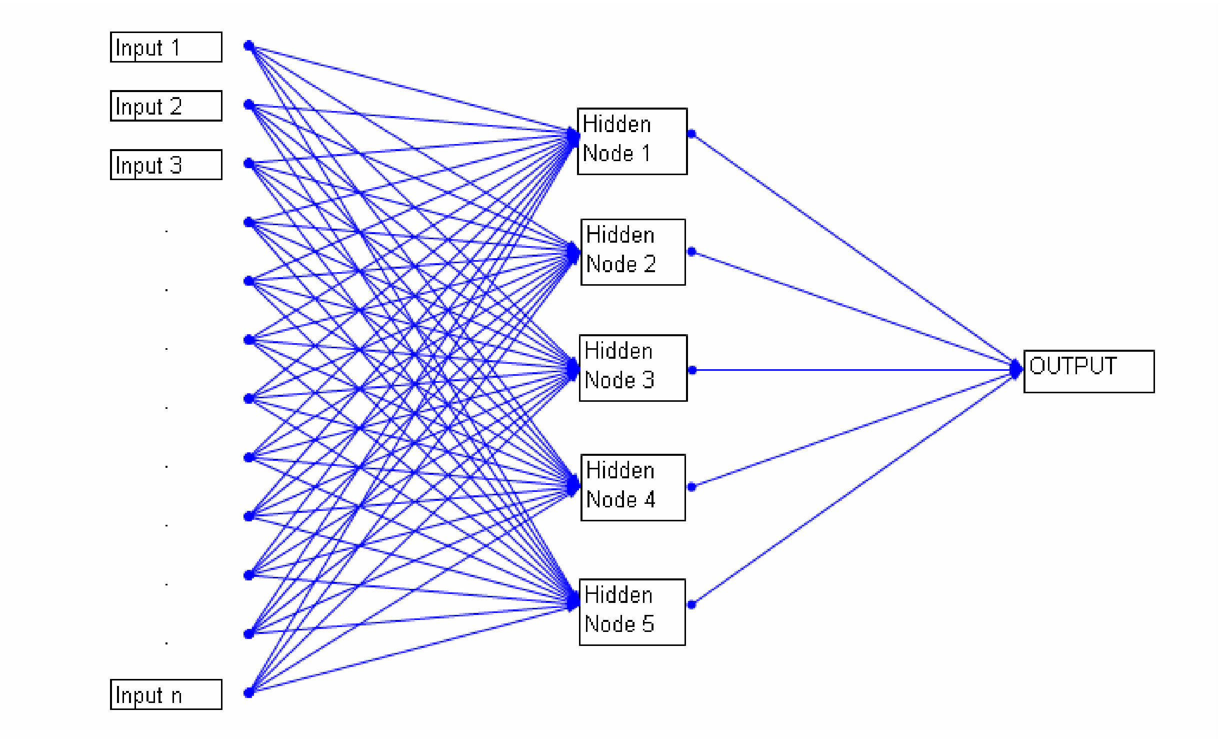

Figure 1: Illustration of the neural net structure fitted

The inputs are the values of the dependent variables (or indicators in the case of categorical variables). Each line linking an input to a hidden node has an associated 'weight'. These are estimated by the fitting algorithm. So, feeding into each hidden node is a weighted sum of the inputs (plus an intercept term). The hidden node applies an ‘activation function’ to this value – in the method we have used, this activation function is:

$$f(x) = \frac{1}{1+e^{-x}}$$

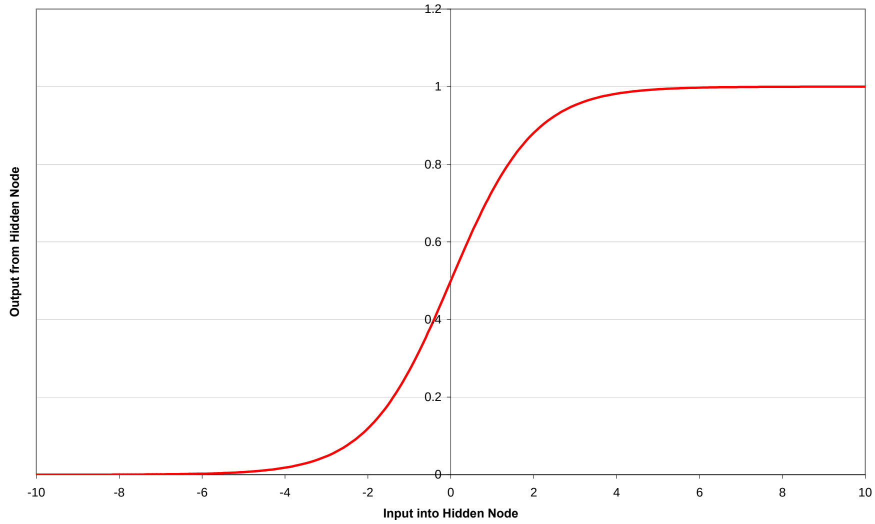

Figure 2: Activation function for the hidden nodes

(Note that this transforms the range (-∞, +∞) into (0, 1).)

The output node is then calculated based on the weighted sum of the outputs from the hidden nodes, again applying an activation function – although in the particular model that we have used, the activation function for the output is just the identity function.

3 The Judging Criteria

3.1 The Ballgown Round

Not all models are as elegant as others. We have assessed – somewhat subjectively – the different models on a number of criteria:

- Ease of setting up the model
- Form and interpretation of output
- Supporting information available with output
- Portability of output
- Ability to deal with trends and other adjustments
- Run-times and scalability (i.e. can the model be applied to large datasets)
- Opportunity to refine the model through manual intervention

3.2 The Swimsuit Round

This round measures how well the model represents the underlying structure of the process being modelled. To do this, we have employed various measures of predictiveness on models fitted to real general insurance data.

Modelling Tests

We have used three real modelling problems encountered in the general insurance field to assess the performance of the different models. These are:

- A retention elasticity analysis for a private motor insurance portfolio
- A 'return to work' probability analysis for a Workers' Compensation portfolio
- An accident claim frequency analysis for a private motor insurance portfolio
- A claim severity analysis for a CTP portfolio

The first two of these are binomial classification problems – i.e. the response variable is lapse / no lapse in the first case, and returned to work / didn't in the second. The third is a 'count' regression problem, and the fourth a regression problem with a continuous output.

The retention, return to work, and CTP severity datasets each contain around 100,000 records, so they are fairly modest in size by insurance modelling standards. The accident claim frequency model has around 400,000 records.

Assessment of Predictive Power

The predictive power of each model has been primarily assessed by dividing each of the source datasets into a training dataset – to fit the model – and testing datasets – to validate the level of accuracy. This is fairly standard practice for most data mining approaches, but not for statistically based models which tend to rely on statistical diagnostics to assess the validity of the models.

Two testing datasets have been created from the source dataset. The source dataset is first split chronologically. So, for example, where we have data from 2001 to 2004, we would reserve the 2004 data for testing. This time-based split is intended to represent the use of a predictive model in practice – i.e. generally we are modelling past experience to predict future experience. We have termed this the 'predicting' dataset.

We have also then split the remaining dataset (2001 to 2003 in our example) by random sampling, with 25% of the data reserved for testing and the remaining 75% used for training. This is the more typical 'data mining' approach for cross-validation of models.

The model is fitted to the training dataset. Predictions are then calculated, from the model results, and attached onto both of the testing datasets. From this, we can look at the results in a number of different ways.

Gains Chart

A gains chart provides a visual summary of how much accuracy 'gain' is achieved by using the model, relative to just a random prediction. It is based on the ranking of the model predictions, so is non-parametric – which means that all types of model can be compared in this way.

A gains chart is generated by first ranking the observations in the dataset according to the prediction given by the model. The x-axis of the gains chart is this ranking expressed as a percentile.

The y-axis is the cumulative value of the observed response variable as a percentage of the total for the whole dataset. If the model were no better than random, then we would expect that the first 5% (say) of ranked cases would have a cumulative value of 5% of the total response. Thus the ‘baseline’ for the gains chart is a 45 degree straight line from 0% to 100%.

If the model does have explanatory power, then the first 5% of ranked cases should have a cumulative value significantly higher than 5% of the total response. Viewed on a graph, this manifests as a convex curve above the baseline model.

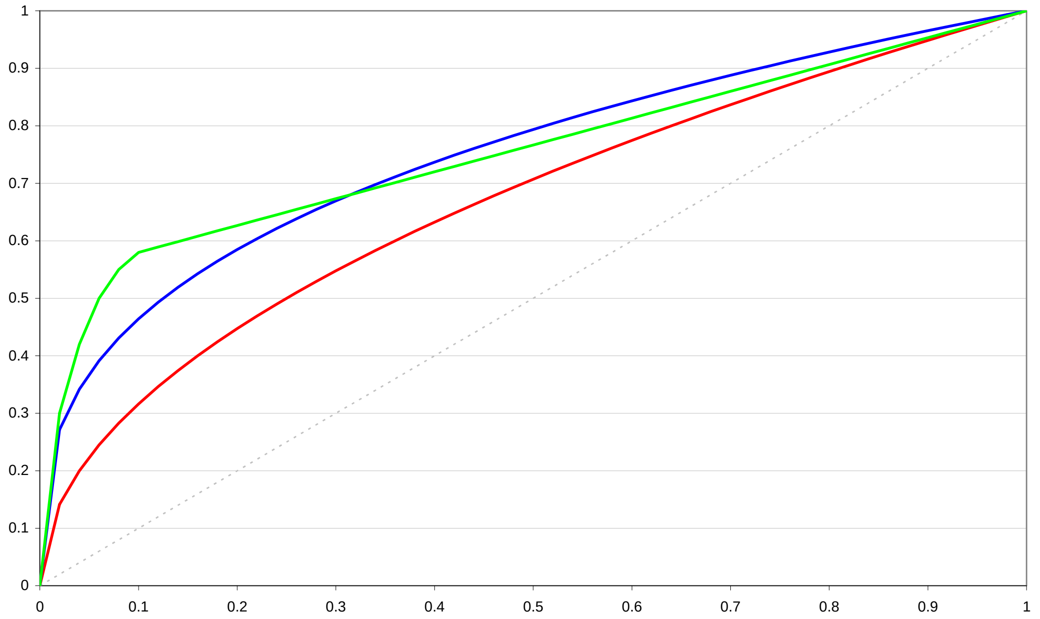

Figure 3: Illustrative gains charts for three different models

Area Under Curve

In the example gains chart shown above, it is clear to see that the blue and the green models are better than the red. The green model is more effective at identifying the most extreme cases than the blue model, but it is less obvious which of the two models is better overall. We can use the area under the graph as an overall measure of the predictiveness of the model, though – in this case, the green model is the better one. A value of 0.5 on this measure indicates a model no better than random.

Misclassification Matrix / Kappa (κ) Statistic

For a classification model, we can look at the misclassification matrix, which – in the binary case – is a 2x2 matrix counting the number of cases that the model classifies as either true or false against whether they are actually observed to be true or false:

	Model = True	Model = False	Total
Observed=True	*a*	*b*	*a+b*
Observed=False	*c*	*d*	*c+d*
Total	*a+c*	*b+d*	*N=a+b+c+d*

The output of the classification models is actually a probability rather than a classification, so first we need to select a 'threshold' probability such that if the modelled probability on a case is less than the threshold, we classify the case as being true, otherwise we classify as being false. The threshold is selected so that the model predicts the correct total for the response (e.g. number of lapses) across all the observations in the training dataset.

The κ-statistic is derived from the misclassification matrix. We would expect some cases to be classified correctly even by a completely random model; *(a+c)(a+b)/N* actually true cases would be correctly classified as true, and *(c+d)(b+d)/N* actually false cases would be correctly classified as false. Call this expected number of correctly classified cases $E_{correct}$. The actual number of correctly classified cases is a+d=$O_{correct}$. The κ-statistic is then calculated as:

$$\kappa = \frac{O_{correct} - E_{correct}}{N - E_{correct}}$$

Sum of Squared Residuals

For a regression model, we need a different goodness of fit measure; we have used the sum of squared residuals:

$$\sum_i (P_i - O_i)^2$$

(where P_i is the model prediction for record i and O_i is the observed actual data for record i).

4 The Ballgown Round

4.1 One-way analysis

Ease of setting up model

It is very straightforward to generate the one-way relativities, although it may involve manually requesting each individual table from the data. No special software is required, although there are packages available that will produce all tables very easily.

It is necessary to 'rebase' the model so that the total predicted response equals the actual total for the modelled dataset.

Form and interpretation of output

The output is a series of one-way tables of factor relativities. This is very straightforward and can be presented graphically. However, there is the potential for 'double-counting' the same risk effect with this approach.

Supporting information available with output

None generated automatically, although simple statistical tests are possible (e.g. chi-squared test for significance of the factor).

Portability of output

It is very straightforward to implement a one-way analysis.

Ability to deal with trends and other adjustments

Possible to include trend adjustments, although trends may be impacted by portfolio mix change effects when looked at on a one-way basis. The method is simple enough that adjustments are easy to apply.

Run-times and scalability

Run-times are quick and will increase in linear terms as the dataset becomes larger.

Data volume requirements

Volume requirements are small, except for factors with many different categories.

Opportunity to refine model

The one-way analysis is rather prescriptive, which limits the scope to improve the model. We could consider rebanding variables, fitting a (univariate) regression line to variables rather than banding, introducing new variables, and removing variables. If adding variables, there is a need to be aware of the risk of double-counting.

Interactions can be investigated, but this would then become a multi-way analysis.

4.2 Multi-Way Analysis

Ease of setting up model

Building an individual table is straightforward. There may be a large number of possible combinations of tables to deal with, so a data cubing or OLAP package is useful to streamline the process. Complexity will increase substantially as the number of dimensions increases – a 3-way table is easy enough to think about, but a 5-way table could be unwieldy.

Generally most factors would be dealt with on a 1-way or 2-way basis, with only selected factors perhaps investigated at higher dimensions.

It is necessary to ‘rebase’ the model so that the total predicted response equals the actual total for the modelled dataset, and this would be a more complex calculation than for a one-way analysis.

Form and interpretation of output

The output is a series of n-way tables of factor relativities. This can be presented graphically, although it is difficult to depict >2-way tables in an easily interpretable way.

2-way tables are generally easy to interpret, but these advantages become rapidly less apparent as the dimensionality increases.

Supporting information available with output

None generated automatically, although simple statistical tests are possible on the tabulated data (e.g. chi-squared test for significance of the factor combination).

Portability of output

Good, although some work may be needed to turn tables into code.

Ability to deal with trends and other adjustments

It is possible to include trend adjustments, although trends may be impacted by portfolio mix change effects. The method is simple enough that adjustments are easy to apply.

Run-times and scalability

If looking at all possible combinations of factors, run-times may become large – particularly where there are a large number of factors to choose from.

Data volume requirements

Volume requirements can be fairly high, particularly to get sufficiently reliable results for unusual combinations of factors (e.g. young drivers in high sum insured vehicles), and increase rapidly as dimensionality increases.

Opportunity to refine model

There is significant scope to look at different combinations of variables for interacting and correlating effects.

4.3 Generalised Linear Model

Ease of setting up model

Given the right software, setting up a GLM is fairly straightforward. Testing that all the statistical assumptions in the model are valid is more difficult, although arguably the impacts of this on the predictiveness of the model itself may be small.

Form and interpretation of output

A set of parameter estimates and supporting statistics, representing the ‘pure’ effects of the factors. It is generally possible to produce a graphical representation of the modelled effects on a factor-by-factor basis, although with some link functions this will not be a direct representation.

It takes experience to be comfortable with interpreting the ‘pure’ effect outputs of GLM, and even then it is possible to get results that are difficult to explain. On the other hand, we can be confident that there is no double-counting of effects when talking to GLM results.

Supporting information available with output

A good range of supporting statistics are available to assess the fit of the model and the uncertainty in the results; some key ones are confidence intervals for the parameter estimates, Type 3 statistics indicating the significance of individual factors in the model fit, and chi-squared or F tests to compare the fit of nested models.

For these diagnostics to be reliable, the statistical assumptions of the GLM must be validated. This validation process does require some statistical expertise.

Portability of output

It is generally straightforward to convert parameter estimates into code and/or tables for implementation outside the modelling software.

Ability to deal with trends and other adjustments

Good; time trends can be investigated in isolation of any portfolio mix effects. Adjustments can be applied either at the data stage or through using the offset in the modelling process.

Run-times and scalability

With specialist software packages, run-times are generally very fast even on large datasets. Other packages may exhibit significant run-time increases or failure if the modelling dataset is too large to fit in PC memory.

Data volume requirements

Volume requirements are modest, except for factors with many different categories.

Opportunity to refine model

There is considerable scope for refinement of GLMs. We can add factors to a model without concern about double-counting effects, since GLM corrects for correlations of exposure between factors. It is straightforward to include interactions, and we can also fit formulas and splines to continuous variables. There are also diagnostics to assess the relative merits of two models, which helps in the refinement process.

4.4 Generalised Additive Model

Ease of setting up model

Currently, GAMs are – as far as we know – only implemented in specialist statistical software packages. These do not generally have a straightforward user interface, so a bit more manual effort is required to set up the model. However, a GAM is structurally quite similar to a GLM and, in fact, the design decisions are easier if anything – because we do not have to specify the formulas or spline points for variables that we are modelling as continuous variates.

Form and interpretation of output

The output is similar to a GLM, i.e. a set of 'pure effect' parameter estimates and supporting statistics, except for factors which have been modelled using a smoother function. The output from these is non-parametric in nature, and so difficult to write out. On the other hand, graphical output of the fitted smoother can be produced and demonstrated (see Figure 4 below). It is also true that, as with a GLM, understanding the individual factor effects is equivalent to understanding the whole model, since the underlying additive structure remains in place.

The output is not really any more difficult to interpret or communicate than GLM.

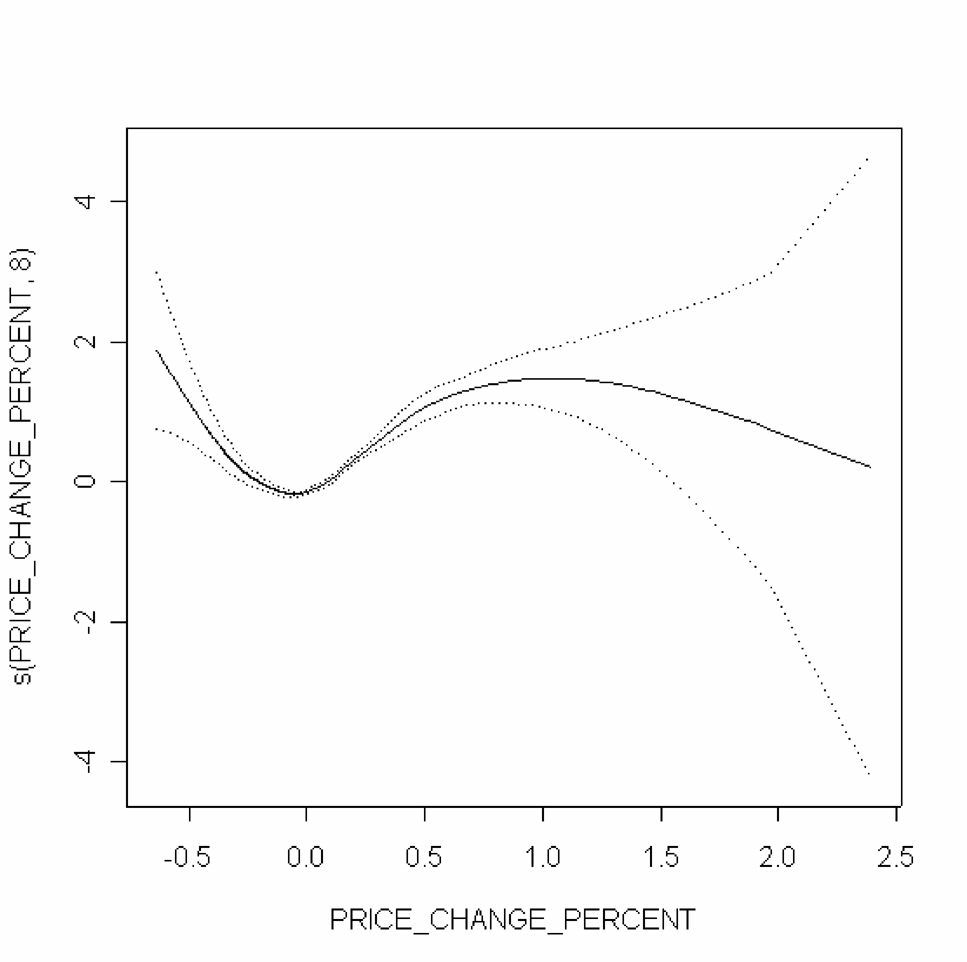

Figure 4: Illustrative output for a smoother function

Supporting information available with output

GAM produces a similar set of supporting statistics to GLM, with some additional diagnostics around the smoother functions.

Portability of output

The non-parametric nature of the smoother estimates does mean that it is much less straightforward – although not impossible – to 'export' GAM results as code for use outside the modelling software.

Ability to deal with trends and other adjustments

Similar to GLM. The ability to fit smoother functions to time trends probably does not confer any advantage here, though, as they are not really appropriate for extrapolation purposes.

Run-times and scalability

Compared to GLM, a GAM will run more slowly since there is, in effect, an iteration within each normal GLM iteration to fit the smoother functions. However, on the datasets we were using – approximately 100,000 records in size – run-times were only a few seconds for both GLM and GAM.

Scalability may be an issue in the current software implementations. We used the GAM implementation in R, which will only run if sufficient RAM is available. We had difficulties in getting the GAM implementation in SAS 9.1 to work even with the relatively small datasets we were using. We would see this as being a software issue, though, rather than a problem with the method itself.

Data volume requirements

Volume requirements are similar to those for GLM, arguably smaller if it is possible to represent complex relationships through a smoother function rather than a categorical factor with a large number of levels.

Opportunity to refine model

The comments applying to GLM are equally valid here. The main advantage of GAM over GLM is that the functions fitted to continuous factors are optimised within the algorithm, rather than relying on the user to refine the model manually.

4.5 CART

Ease of setting up model

CART/decision tree models are easy to set up given specialist software, and is primarily a question of deciding which factors are to be included in the investigation since no real structure is assumed.

Form and interpretation of output

The output is in the form of a tree, which is effectively a set of If… then… else… rules leading to a set of discrete segments of the data. The average of the target response within the segment is then the predicted outcome from the model for that segment.

The tree can be represented visually, and colour-coded to highlight 'bad' segments and 'good' segments, for example. This is quite appealing, but does require a very large sheet of paper if the tree becomes complex! It is also interesting to look at the hierarchy of variables selected for each split, as this gives some indication of the relationships between factors in the data.

The fact that the end output is a set of discrete segments can be fairly appealing for many applications. Interpretation in this sense is straightforward. Nonetheless, except for very simple trees, actually interpreting the route through the factors to the end segments – a large series of nested if statements – can be quite difficult.

Supporting information available with output

CART produces good supporting information to go with the output, including cross-validation against a test dataset.

Portability of output

The tree is simply a set of If… then… else… rules that can be easily converted to code.

Ability to deal with trends and other adjustments

CART does not really provide any insight into trends in itself. The best approach may be to investigate trends as a separate exercise, and then pre-adjust the data so that it is a priori representative of a future expectation.

Run-times and scalability

CART runs very quickly – in a matter of seconds – on the datasets we used for testing. In the software implementation we are using, there is a memory limit that restricts the size of dataset, so scalability may be an issue.

Data volume requirements

CART is somewhat profligate with data, in effect throwing away half the data for each split produced. As such, it needs fairly large volumes of data to produce a reasonably complex model. Our tests have indicated that it is the most sensitive of the methods to the volume of available data.

Opportunity to refine model

CART is quite automated, so opportunities to refine the model are limited, although additional factors can be included in the model specification.

4.6 Neural Net

Ease of setting up model

Neural nets are quite easy to set up given specialist software. Again, no particular structure to the data is assumed, and so the main decisions are:

- which factors are to be included in the investigation
- how many hidden nodes are to be used

Deciding the appropriate number of hidden nodes is by no means obvious. We have used 5, which is really quite few given the number of input nodes (i.e. continuous factors and categorical levels) we had, but nonetheless seemed to give reasonably predictive results within just about reasonable run times.

Form and interpretation of output

The output is a set of weights attached to each link in the network. This has no meaningful interpretation.

The model therefore becomes something of a 'black box' – its only useful output is the prediction itself. We can investigate these predictions and get some sort of feel for the model, but only in broad terms.

One feature of neural nets is that the algorithms for fitting the weights do not generally find a globally optimal solution, and you will normally get a different solution emerging each time you run the model. It is therefore necessary to run a number of trial model fits from different starting points, and then – of those trials that converge on a solution – select the one that gives the best fit. For one of our models, the motor accident claim frequency model, we were not able to achieve a sensible result within twenty trial models.

Supporting information available with output

Minimal supporting information is available; most packages allow validation of the predictiveness against a testing dataset.

Portability of output

The neural net weights can be implemented outside the modelling software, although this is certainly the least portable of the methodologies.

Ability to deal with trends and other adjustments

Neural nets could be somewhat unpredictable if used for extrapolation, so it is not a good framework for looking at trends.

Run-times and scalability

Run-times are the longest of all the models, taking hours rather than minutes. We have not tested the scalability, but we would suspect that neural nets on large datasets are not practical on current PCs, and special hardware might be required.

Data volume requirements

There is more structural complexity to a neural net than a GLM, but – unlike CART – all the data is used throughout. Our view is that data volume requirements will be intermediate between GLM and CART.

Opportunity to refine model

Neural nets are entirely automated, so there is no opportunity to refine the model, although additional factors can be included in the specification.

4.7 Results of the Ballgown Round

In the traditional reverse order, we have:

- Neural nets
- CART
- Multi-way analysis
- GAM
- One-way analysis
- GLM

GLM wins over one-way and multi-way analysis mainly because it avoids the risk of double-counting effects without being significantly more difficult to apply in practice. GAMs are similar to GLMs in many ways, but the non-parametric element does mean they are significantly less portable. Also the software technology is less mature. CART has some appealing features, but the results of a complex tree can be difficult to interpret meaningfully. However, they are not nearly as impenetrable as Neural Nets, the clear loser of this round.

5 The Swimsuit Round

5.1 Gains Charts and Area Under Curve

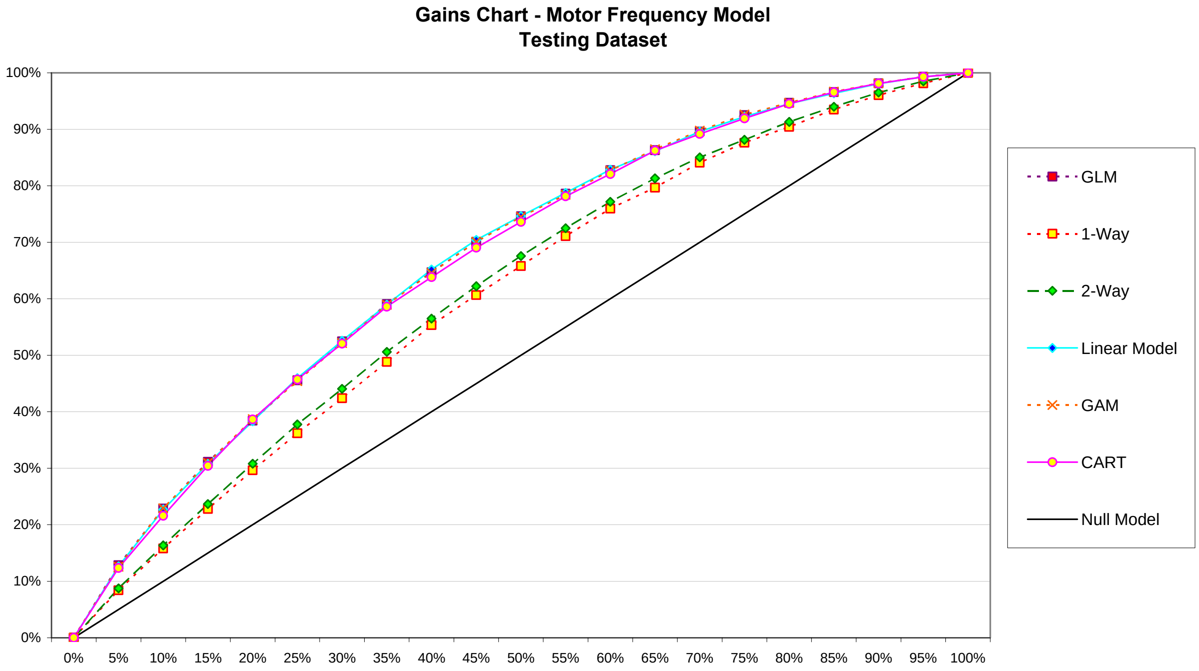

Figure 5: Example gains chart result

The gains chart results generally show that the models fitted achieve similar levels of gain. This is not surprising, given the constraints we have imposed on them – in particular, all the models are using exactly the same set of factors to explain the response.

A gains chart only shows differences between models where they have ranked observations differently. This may mean that structural problems with models may not show up in the gains chart, at least not as direct effects. Such problems might include:

- an ill-fitting distribution of results
- absurd results at the extreme ends of the range

We have not reproduced all the charts here, but table 1 below shows the area under the curve (AUC) for all the models fitted, on each of the training, testing, and predicting datasets. The best-performing model in each test has been highlighted.

		Motor - Retention	Motor - Accident Claim Frequency	Workers - Return to work probability	CTP - Claim Severity
Predicting	One-way	72.5%	58.8%	-	-
	Two-way	72.4%	60.1%	-	-
	Simple Linear Model	72.9%	65.1%	-	-
	GLM - Poisson Log	73.2%	65.3%	-	-
	GLM - Binomial Logistic	73.2%	-	60.4%	-
	GLM - Gamma Log	-	-	-	56.2%
	GAM	73.4%	65.3%	60.5%	56.8%
	CART	71.3%	64.5%	59.3%	56.2%
	Neural Nets	73.2%	-	58.4%	56.3%
Testing	One-way	72.9%	60.6%	-	-
	Two-way	72.9%	61.6%	-	-
	Simple Linear Model	73.4%	67.0%	-	-
	GLM - Poisson Log	73.8%	67.0%	-	-
	GLM - Binomial Logistic	73.7%	-	57.8%	-
	GLM - Gamma Log	-	-	-	57.4%
	GAM	74.0%	67.0%	57.8%	57.9%
	CART	72.2%	66.6%	57.2%	56.7%
	Neural Nets	74.3%	-	55.2%	57.6%
Training	One-way	73.2%	60.8%	-	-
	Two-way	73.6%	61.5%	-	-
	Simple Linear Model	73.9%	67.5%	-	-
	GLM - Poisson Log	74.2%	67.5%	-	-
	GLM - Binomial Logistic	74.2%	-	61.8%	-
	GLM - Gamma Log	-	-	-	57.5%
	GAM	74.4%	67.4%	62.0%	58.0%
	CART	72.6%	67.1%	59.4%	56.9%
	Neural Nets	74.6%	-	64.5%	58.9%

Table 1 – Area Under Curve statistics from Gains Charts

From table 1, some patterns emerge:

- The Neural Net is the best when matched against the training dataset, but it performs relatively less well on the testing and predicting datasets; this may indicate a problem with over-fitting.
- GAM outperforms GLM in most tests, although not generally by a large margin. The models are structurally quite similar so this is not surprising. On the frequency model, there was only one available factor (driver age) that could be sensibly modelled with a smoother function, so there is no real advantage from using GAM over GLM here.
- Both GAM and GLM show consistent results between the training, testing and predicting datasets. GAM gives the best result on all four models with the predicting dataset, although we would not read too much into this.
- The one-way and two-way models underperform the GLM/GAM models. The difference is not large on the retention model, which is dominated by just two factors (change in price since last year, and payment frequency). In contrast, on the frequency model – where there are a number of variables with strong explanatory power that are correlated in terms of exposure – there is a significant difference.
- The two-way model shows very little, if any, improvement over the one-way model on the gains chart basis, despite the significant additional complexity of the model. GLM is clearly more efficient at dealing with correlations of exposure. A two-way might have an advantage over GLM if there is a strong interaction effect between two variables.
- CART is generally the weakest performer. It may be somewhat handicapped because it is the most ‘data-hungry’ of the models tested, and the data volumes in these tests are not very large. It does perform noticeably better on the frequency test, which has the largest data volumes.
- CART’s performance is reasonably consistent between the training, testing and predicting datasets.

5.2 Misclassification Matrices and κ-statistics

Table 2 below shows the κ-statistics for the two classification problems.

κ Statistics from Misclassification Matrices

		Motor - Retention	Workers Comp - Return to Work
Predicting	One-way	0.176	-
	Two-way	0.168	-
	Simple Linear Model	0.191	-
	GLM - Poisson Log	0.191	-
	GLM - Binomial Logistic	0.193	0.125
	GAM	0.195	0.126
	CART	0.191	0.112
	Neural Nets	0.205	0.102
Testing	One-way	0.186	-
	Two-way	0.186	-
	Simple Linear Model	0.214	-
	GLM - Poisson Log	0.207	-
	GLM - Binomial Logistic	0.206	0.047
	GAM	0.210	0.046
	CART	0.184	0.046
	Neural Nets	0.232	0.024
Training	One-way	0.183	-
	Two-way	0.193	-
	Simple Linear Model	0.215	-
	GLM - Poisson Log	0.218	-
	GLM - Binomial Logistic	0.216	0.096
	GAM	0.219	0.095
	CART	0.202	0.083
	Neural Nets	0.235	0.142

Table 2 - κ-statistic results

The Neural Net appears to show the best results here on the retention model. However, on the return to work model, it has the highest κ-statistic on the training model but the lowest on the two validation datasets. This would tend to indicate overfitting.

Among the other models, GAM and GLM appear to perform similarly, and better than CART.

5.3 Sum of Squared Residuals

Table 3 shows the sum of squared residuals for the motor accident claim frequency model and the CTP claim severity model.

	Motor Accident Claim Frequency			CTP Claim Severity		
Models	Predicting x10^{4}	Testing x10^{3}	Training x10^{3}	Predicting x10^{14}	Testing x10^{14}	Training x10^{14}
1-Way	1.236	5.270	3.303			
2-Way	1.231	5.258	3.300			
Linear Model	1.228	5.220	3.293			
GLM - Poisson Log	1.217	5.180	3.273			
GLM - Gamma Log				1.536	2.249	6.672
GAM	1.217	5.180	3.273	1.533	2.244	6.658
CART	1.229	5.224	3.292	1.534	2.248	6.654
Neural Nets				1.538	2.247	6.588

Table 3 – Sum of Squared Residuals for Regression Model

A smaller value for the sum of squared residuals indicates a better fit. The best-ranking model in each case has been highlighted.

For the frequency model, the results for the GLM and GAM are almost identical – which we would expect since the only difference is that the GAM has a smoother spline fitted to driver age. However, they outperform all the other models – including the linear model and CART, which had very similar gains chart results to GLM and GAM. This may indicate that while the linear model and CART are effective at ranking the outcomes, they may be less effective at the actual quantitative prediction.

With the CTP severity model, the Neural Net again shows the best result on the training dataset, but this advantage is lost on the two validation datasets. GAM is the best performer on the validation datasets.

5.4 Overall Results

So, while there is no runaway winner in this round, we can form a view on the predictive power of the models across the four different modelling problems. In the traditional reverse order, we have:

- One-way analysis
- Multi-way analysis
- CART
- Neural Nets
- GLM
- GAM

Neural Nets clearly have predictive potential, but – at least in the applications we have investigated here – it has shown a tendency to overfit.

CART was generally the weakest performer in these tests, but it is the most data-hungry of the techniques, and the relatively small data volumes used in these tests perhaps do not work in its favour.

GLM exhibited solid performance throughout, but GAM had a small but consistent edge over it. GAM is particularly useful where there are continuous factors that have strong but complex (or at least non-linear) effects on the response.

The simple linear model appeared to perform well in comparison with the GLM and GAM on the gains chart, but did exhibit a significantly higher sum of squared residuals. This indicates that, while it might be effective at ranking observations, it is less good at modelling the actual distribution of results.

6 Final Results and Conclusions

The results of the individual rounds are shown below:

Ballgown Round ("Elegance")	**Swimsuit Round ("Predictiveness")**
Neural Nets	One-way analysis
CART	Multi-way analysis
Multi-way analysis	CART
GAM	Neural Nets
One-way analysis	GLM
GLM	GAM

Depending on the application, predictiveness may be more or less important relative to elegance. Arbitrarily, then, awarding a points scale from 0 to 5 for the Ballgown round and double points from 0 to 10 for the Swimsuit round, we arrive at the following results:

4th place CART

A middling performance in both rounds, but CART does produce attractive visual output and is clearly the model to choose for applications where a discrete segmentation is the desired outcome. While its predictiveness was not the highest, the volumes of data used in the test were perhaps not in its favour.

3rd place Neural Nets

Despite accusations of looking too much like an ugly black box to be a viable beauty contest winner, the neural net does have significant predictive potential – albeit with some risk of over-fitting. Certainly it is not the right choice where interpretation of the model is important, but where accurate prediction is the main requirement, and where there may be complex relationships between factors, neural nets may prove to be effective.

Joint 1st place GLM and GAM

Although getting on in age for a beauty contestant, GLM's combination of elegance and solid predictive powers means it is still a good choice. However, GLM has to share the honours with GAM, a younger contestant with a bit more added flexibility.

While GAM has some disadvantages relative to GLM, these are largely a result of the maturity of the software implementations rather than deficiencies of the technique itself. Even so, it is still a practical technique and it does have an edge in terms of predictive power.

GAM is likely to be the better choice for modelling situations where there are some key continuous variables exhibiting complex progressions that are difficult to represent within the more restrictive GLM framework.

7 Software Packages

The software packages used for this paper were as follows:

Table analysis and GLM	Glean	(http://www.prophet-web.com/Products/Glean)
GAM	R	(http://www.r-project.org)
CART	CART	(http://www.salford-systems.com/cart.php)
Neural Nets	JMP	(http://www.jmp.com)