Institute of Actuaries of Australia

Application of Soft-Computing Techniques in Accident Compensation

Prepared by

Peter Mulquiney
Taylor Fry Consulting Actuaries

Presented to the Institute of Actuaries of Australia
Accident Compensation Seminar 28 November to 1 December 2004.

This paper has been prepared for the Institute of Actuaries of Australia's (IAAust) Accident Compensation Seminar, 2004. The IAAust Council wishes it to be understood that opinions put forward herein are not necessarily those of the IAAust and the Council is not responsible for those opinions.

© 2004 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.a

Abstract

In this paper, soft-computing methods are applied to some aspects of loss reserving and pricing for a motor bodily injury (CTP) portfolio. In particular, the performance of a GLM model of the average size of finalised claims is compared to models developed using the soft-computing techniques, neural networks, MARS and MART.

Both the neural network and MART models were found to have better prediction accuracy on past experience periods than the GLM model. Predictive accuracy was measured by both the sum of squares, and the average absolute error, in a separate test data set. However, both the neural network and MART models had features which made them less suitable than the GLM model for projecting claim sizes into future periods.

Table of Contents

1 Introduction

Accident compensation data often exhibit features which make loss reserving and pricing difficult when using traditional actuarial techniques such as the chain ladder method. Typical features observed in the data of accident compensation schemes which complicate the analysis include:

- changes in the rate of claim finalization ;
- legislative changes;
- seasonality; and
- superimposed inflation which varies by experience year and age of claims.

One method of dealing with these features is through conventional statistical modelling techniques such as Generalised Linear Modelling ("GLMs"). Indeed this is the topic of Taylor and McGuire's paper "Loss reserving with GLMs" (also presented at this conference). An alternative group of techniques that are also potentially useful are those based on the ideas of soft-computing.

Soft-computing techniques include methods such as neural networks, MARS ("Multiple Adaptive Regression Splines), and decision tree based methodologies like MART ("Multivariate Additive Regression Trees"). A strength of these techniques is their ability to model non-linear relationships. What distinguishes them from more traditional approaches in this respect is that they can identify and model nonlinearities almost automatically. In other words, the modeller does not need to define the nonlinearities and interactions explicitly as is necessary with conventional techniques such as GLMs.

In this paper, I will discuss the application of soft-computing methods to the problems of reserving for a motor bodily injury (CTP) portfolio. In particular I will compare the performance of a GLM model with the soft-computing techniques, neural networks, MARS, and MART, and will discuss some of the potential advantages and disadvantages of these methods.

The application of soft-computing in actuarial science is not new. The review papers by Shapiro (2001, 2003) provide an overview of the published actuarial applications. The applications are wide ranging and include data mining (e.g., Kolyshkina and Brookes, 2002), underwriting and risk classification, as well as insolvency modelling. However to date, little work has been devoted to these methods for pricing and reserving in longer tailed classes of business such as accident compensation portfolios. Note that the current paper only considers some issues in relation to aggregate pricing and loss reserving; for example, risk rating is not considered.

2 Overview of soft computing techniques

In the following section I give an overview of the theory behind neural networks, MART, and MARS and compare these methods to GLMs. This overview is intended to be brief with the main motivation to give the reader some insight into:

- the differences in the architectures of the models that are produced by each of these methodologies; and

- how each of these methodologies deals with the problem of overfitting. In other words, how each of these methodologies attempts to fit just the underlying trends in the data, and not the “noise”.

For readers wishing to gain a greater understanding of these methods the textbook by Hastie, Tibshirani, and Friedman (2001) is recommended. More detail on each of the individual methods can be found in the following sources: Bishop (1995) and Ripley (1995) for neural networks; Friedman (2001) for MART; and Friedman (1991) for MARS.

2.1 Model architectures

All the models discussed in the present paper are types of regression models. That is, they attempt to predict an outcome measurement, $\boldsymbol{Y}$, from a vector of p predictor measurements, $\boldsymbol{X}$. Here, the outcome measurement is often referred to as the dependent or response variable, while the predictor measurements are often referred to as independent variables, inputs or covariates. In other words, each of these methods gives us a function of the predictor measurements, $f(\boldsymbol{X})$, for predicting $\boldsymbol{Y}$.

In this section, the general form (or architecture) of the regression functions produced by each of these methodologies is presented.

2.1.1 GLMs

Given our vector of inputs $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$, the GLM has a regression function of the form

$$f(\boldsymbol{X}) = g^{-1}(\eta) \qquad [2.1]$$

$$\text{where} \quad \eta = \beta_0 + \sum_{i=1}^{p} \beta_i X_i$$

with β_i being unknown parameters and the variables X_i being:

- direct quantitative inputs such as accident quarter, quarter of finalisation, etc.

- transformations of quantitative inputs such as X_i^2, X_i^3, $X_i^{1/2}$, log (X_i), and (X_i - c)$_+$. The last function in the list is known as a linear spline and the "+" subscript means that the function is zero when X_i - c is negative.

- numeric coding of the levels of a qualitative input. For example, for a two level qualitative input such as sex we could create $X_1 = I$(sex = male) and $X_2 = I$ (sex = female). Here I(.) is the indicator function which is 1 when the statement within the parentheses is true and 0 when not. Using this coding, the effect of sex is modelled as by two sex-dependent constants.

- Interactions between input variables such as $X_3 = X_2.X_1$.

The function g(.) is known as the link function and for many insurance applications, the log function is used for the link function. η is often referred to as the linear predictor.

As indicated by Eqn [2.1], the GLM regression function has a large amount of flexibility. The link function, input transformations, and interaction terms allow one to construct regression functions for quantities which are complicated and non-linear functions of their inputs. This flexibility is one reason for the widespread use of GLMs in actuarial applications.

However, determining the appropriate input transformations and interactions to include in a GLM model can be difficult to do in practice. This is an area where the skill of the model builder can play a large part in determining how well the regression function will model the data.

2.1.2 Neural networks

In the previous section, we saw that the basic approach of GLMs was for the model builder to match the architecture of the regression function to the data. The approach of neural networks is somewhat different. Instead of matching the model to the data, the neural network regression function is given an initial architecture that is so flexible it can model almost anything. Careful fitting is then used to constrain the function so that it will only describe the underlying features of the data.

Starting with our vector of inputs $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$ we can construct a neural network regression function as follows. First we create M linear combinations of inputs

$$h_m = \sum_{i=1}^{p} w_{mi} X_i \qquad [2.2]$$

The actual value that we choose for M will be determined in the tuning/fitting process (see section 2.2). These M linear combinations are then passed through a layer of activation functions $g(h_m)$ (Fig. 2.1) to produce the outputs Z_m

$$Z_m = g(h_m) = g(\sum_{i=1}^{p} w_{mi} x_i) \qquad [2.3]$$

These first steps correspond to the middle (or hidden) layer of the neural network (Fig. 2.2).

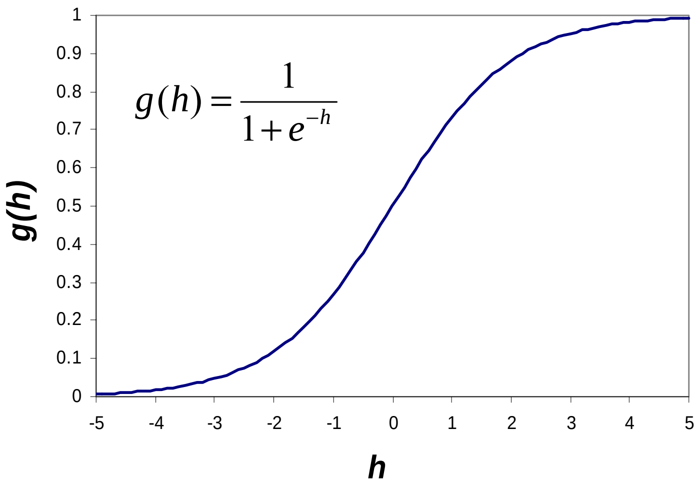


Figure 2.1 A sigmoidal activation function. A sigmoidal curve is usually chosen as it introduces non-linearity into the regression function while keeping responses bounded.

The regression function is then taken to be a linear combination of the outputs from the hidden layer.

$$f(X) = \sum_m W_m Z_m = \sum_m W_m g(\sum_{i=1}^{p} w_{mi} x_i) \qquad [2.4]$$

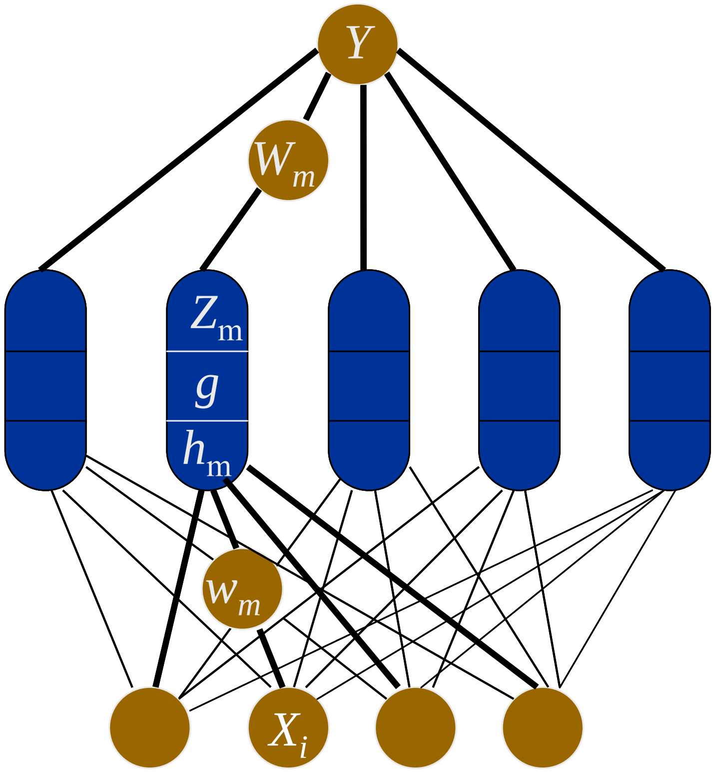


Figure 2.2 The structure of a neural network. This neural network has a single hidden layer with 5 hidden units ($M = 5$). Figure adapted from Gershenfeld (1999).

The parameters of this regression model are the weights. In their simplest form these regression functions will have $(p+1) \times M$ parameters. Typically there are many more parameters in a neural network regression function compared to a GLM regression function.

As might be expected, this architecture produces a regression function that is very flexible. Indeed, it has been shown that a neural network regression function with a single hidden layer and enough hidden units can describe any continuous function to any desired degree of accuracy. Further, if you introduce a second hidden layer, it can be shown that the neural network can describe any function with a finite number of discontinuities.

2.1.3 MART

"Multiple Additive Regression Trees" (MART) was first developed by Jerome Friedman in 2001. This technique is also known as gradient boosting and is the basis of the Salford Systems data mining product "Treenet".

Before we discuss the architecture of the MART regression function, it is necessary to have a basic understanding of regression trees. Regression trees are regression functions which partition the predictor variable values into disjoint regions and model the response of each region by the average response observed in the region. For example, if our vector of inputs was $\boldsymbol{X} = (X_1, X_2)$, then a regression tree with 4 regions (or terminal nodes) would partition the predictor space into 4 regions (Fig. 2.3). The response from each region would then be modelled by a constant as follows

$$f(X) = \sum_{m=1}^{4} c_m I\{(X_1, X_2) \in R_m\} \qquad [2.5]$$

$$\text{with } c_m = average(Y_i \mid X_i \in R_m)$$

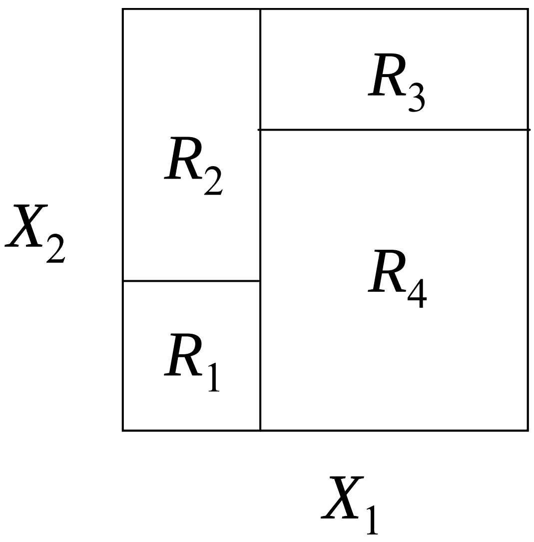


Figure 2.3 A vector of 2 inputs divided into 4 regions.

The idea of MART is to form a regression function out of a “committee” of small regression trees. Small regression trees, for the purposes of MART, divide the input space into between 2 to 8 regions. Hence, each regression tree on its own is a very poor regression function. However by forming a committee of these trees the predictive power of the resultant regression function is greatly improved.

The committee of regression trees is constructed in an automated stagewise manner. In other words, the regression function is automatically grown by adding new regression trees one at a time. At each addition, only the parameters of the newly added tree are estimated, with the parameters of the existing trees remaining the same. Thus as new trees are added, the features of the data set become progressively better represented by the regression function.

Hence, the overall regression function for a MART model is the sum of a number of individual piecewise constant functions. This means that they are well suited to modelling discontinuities. In addition, because of the large number of trees usually involved in the regression function, they are still able to well approximate smooth curves, albeit in a piecewise manner.

2.1.4 MARS

Multivariate adaptive regressions splines (“MARS”) is an adaptive regression method that builds up a regression function automatically in a forward stepwise manner using linear splines. Linear splines have the functional forms $(X_i - c)_+$ or $(c - X_i)_+$ where the constant c is called the knot.

The MARS algorithms adds linear splines to the regression function one at a time. The particular linear spline that is chosen at each stage is determined by computational brute force and is simply the spline that gives the biggest decrease in the residual sum of squares when added to the regression function. Note that the new spline may be added alone or as an interaction term with one or more of the linear splines already present in the regression function. In this way the model architecture automatically adapts to match the features of the data.

As can be noted from the above description, the overall architecture of a MARS regression model is less flexible than a GLM since:

- linear splines are the only input transformation that is allowed; and
- the regression function does not explicitly include a link function.

However, MARS has the advantage over GLMs that it automatically and adaptively determines the architecture of the regression function.

2.2 The problem of overfitting

A goal of the previous section was to give some insight into the different regression function architectures that are possible with the different modelling methods. For all the methods discussed, it was seen that each method could yield a regression function with a large amount of flexibility, although still subject to the limitations of its underlying building blocks (or basis functions).

However, for all of the methods discussed, if the regression function is equipped with a sufficient number of inputs and parameters, it is possible for the regression function to model the observed responses exactly. In this case one has modelled not only the underlying features of the data but also the noise inherent in the data - the model has been overfitted. Choosing a regression function which has not been overfitted is a problem with which all the methods discussed in this paper must address.

In the following subsections, I briefly discuss how this is done for each of the methods.

2.2.1 GLMs

When fitting a GLM model to data, it is necessary for the modeller to specify the inputs, interactions, and transformations to use in the regression function (as discussed in section 2.1.1). It is also necessary to specify the assumed statistical distribution of the response variable. Having done this, the parameters of the regression function can be estimated by maximum likelihood estimation.

By using a statistical approach to parameter estimation, one is able to construct statistical tests. These can be used to assess whether the addition or removal of terms to the regression function has led to a statistically significant improvement in the model or to see whether the estimated coefficient of a particular input is statistically significant. By using these tests, and along with other considerations, the modeller attempts to construct a regression model which contains as few parameters as is necessary.

So for GLMs the modeller uses statistical reasoning to choose a model architecture sufficient to model the underlying features of data without overfitting.

2.2.2 Neural networks

For neural networks we have seen that from the outset, the regression function has an architecture that is so flexible it is capable of overfitting the data. To prevent against overfitting, it is necessary to constrain the fitting so that the model only describes the underlying features of the data.

Before we discuss the approach used to protect against overfitting, it is important to realise some other distinctions between parameter estimation for neural networks as opposed to GLMs.

Firstly, when applying neural networks (as well as MART and MARS) no assumption is usually made about the statistical distribution of the response. This means that the statistical tests that are used to protect against overfitting in GLMs are not available.

In addition, by not adopting a statistical distribution for the response, it is not possible to estimate parameters by maximum likelihood estimation. For these methods, the parameters are typically estimated by specifying a loss function that needs to be minimised. This is typically the squared error loss function (or "sum of squares").

Given these considerations, the way that overfitting is prevented in neural networks is by adding a penalty function to the sum of squares error function which becomes larger as the regression function becomes less smooth. The penalty function is typically defined by

$$sum\ of\ squares + \lambda(\sum_m W_m{}^2 + \sum_m \sum_p w_{mp}{}^2) \qquad [2.6]$$

where the W_m and w_{mp} are the weight parameters from the neural network regression function (Eqn [2.4]). It is seen that the weight decay parameter, λ, controls the magnitude of the penalty. So by choosing a larger λ, we cause the fitted regression function to be smoother.

A question still remains about how to best choose the weight decay parameter, λ. A typical way of determining this is by cross-validation. For cross-validation, we randomly divide our data into a training data set and a test data set. We then fit a number of neural network models to the training data using a number of values of λ. The sum of squares in the test data set is then determined for each of the models. The λ value that minimises the sum of squares in the test set, is the λ value that is chosen.

The rationale behind cross-validation is that as the value of λ gets smaller, the regression function will become less smooth and start to fit the underlying features of the data. Because the underlying features of the data should be common to both the training and the test set, the sum of squares in both sets will decrease. However, as the value of λ continues to decrease, the function will begin to model the noise in the training set. Because the noise will be different in both the training and test data sets, the sum of squares in the test data set will start to increase. At this point we have begun to overfit the data.

Note that cross-validation is generally used to fit both λ and the number of units in the hidden layer, *M* (Section 2.1.2).

2.2.3 MART and MARS

For MART the problem of overfitting is addressed by specifying an appropriate size for each component regression tree, the number of regression trees that are added to the regression tree, as well as another tuning parameter termed the shrinkage parameter (for more details see Hastie et al., 2001). As for neural networks, the appropriate values of these tuning parameters are determined using cross-validation.

For MARS, the problem of overfitting is addressed by choosing the appropriate number of terms to keep in the regression model. This too, can be determined by cross-validation. However, it is usually determined by a computationally more efficient method known as generalised cross validation (for more details see Hastie et al., 2001).

3 Case study

The architectures and features of the soft-computing methods described above indicate that they may be useful for modelling accident compensation data, particularly where the data exhibit features that are difficult to model using traditional actuarial techniques such as the chain ladder.

In their paper, "Loss Reserving with GLMs", Taylor and McGuire (2004) present one such data set from a CTP portfolio. This data set was shown to have features such as

- changes in the rate of claim finalisation;
- legislative changes;
- seasonality; and
- superimposed inflation which varies by experience year and age of claims.

In the paper, the authors comment that these features are not uncommon in accident compensation data and demonstrate how the traditional chain ladder has difficulty in coping with these features. They then go on to demonstrate how the architecture of the GLM provides an effective framework for dealing with these features.

In the present paper, I investigate the possibility of using soft computing methods as an alternative to GLMs to model this data set.

3.1 Data

The data set relates to CTP insurance in one state of Australia. Following Taylor and McGuire we have restricted our analysis to a model of the average size of finalised claims. The justification for this choice can be found in their paper.

The data set consists of a claim file consisting of approximately 60,000 claims. For each claim various items are recorded, including, the date of injury, date of notification, and histories of paid losses, case estimates and finalised/unfinalised status including dates of change of status.

For this analysis, all paid loss amounts have been converted to 30 September 2003 values in accordance with past wage inflation in the state concerned. A summary of the average sizes of finalised claims is provided in the Appendix. This is the usual triangular summary of data with rows representing accident quarters, columns development quarter, and diagonals calendar quarter of finalisation. In this triangle, each cell (i, j) represents the average size of all claims finalised in accident quarter, i and development quarter j.

For our regression models we are interested in modelling the size of the rth finalised claim, Y_r in terms of:

- i_r = accident quarter = 1, 2, 3, …, 37
- j_r = development quarter = 0,1, 2, …, 36
- k_r = calendar quarter of finalisation = $i_r + j_r$
- t_r = operational time = proportion of claims incurred in accident quarter i_r which have been finalised at development quarter j_r
- s_r = season of finalisation = March, June, September, and December

Hence for each of the different methods our regression function will have the general form:

$$Y_r = \mathrm{f}(i_r, j_r, k_r, t_r, s_r) \qquad [3.1]$$

3.2 Methodology

All analysis was performed using the software "R". This software is freely available at http://www.r-project.org/foundation/ and is widely used by academic statisticians. The algorithm packages nnet, gbm, and polspline were used for the neural network, MART, and MARS algorithms, respectively.

For the analysis, individual finalised claim data were used rather than aggregated data. The tuning parameters of each of the soft-computing methods were determined by cross-validation. This involved constructing a training data set by randomly selecting 2/3 of the data, with the remaining 1/3 forming the test data set. The final models presented below were, however, fitted to the full data set.

3.3 Results

3.3.1 Summary of the GLM model from Taylor and McGuire (2004)

The GLM model of the average size of finalised claims that was determined in Taylor and McGuire (2004) was

$$\begin{aligned} E[Y_r] = \exp\{\alpha &+ \beta^d_1 t_r + \beta^d_2 \max(0,10-t_r) \\ &+ \beta^d_3 \max(0,t_r-80) + \beta^d_4 I(t_r < 8) && [\textit{Operational time effect}] \\ &+ \beta^s I(k_r\text{=March quarter}) && [\textit{Seasonal effect}] \\ &+ \beta^f_1 k_r + \beta^f_2 \max(0,k_r - 2000Q3) \\ &+ \beta^f_3 I(k_r<97Q1) && [\textit{Finalisation quarter effect}] \\ &+ k_r [\beta^{tf}_1 t_r + \beta^{tf}_2 \max(0,10-t_r)] && [\textit{Operational time x finalisation quarter interaction}] \\ &+ \max(0,35-t_r) [\beta^{ta}_1 + \beta^{ta}_2 I(i_r > 2000Q3)]\} && [\textit{Operational time x accident quarter interaction}] \end{aligned}$$

[3.2]

with the response assumed to follow an exponential dispersion family distribution with a variance power of 2.3 (Taylor and McGuire, 2004). A plot of the log of the regression function (the linear predictor) is shown in Figure 3.1.

Eqn [3.2] and Figure 3.1 illustrate the complex features that are present in the finalised claim data. There are 5 main features:

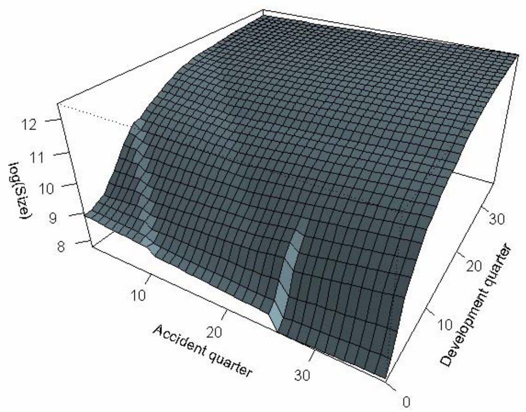
- Operational time effect: Because of changes in the rate of claims finalisation, the regression function includes an operational time effect rather than a development quarter effect. This effect shows that the average size of finalised claims increases with operational time.

- Seasonal effect: Claims finalised in the March quarter tend to be slightly lower than other quarters.

- Finalisation quarter effect: This represents superimposed inflation and indicates that there is a change in the rate of superimposed inflation before 1997 and at the end of the September 2000 quarter.

- Operational time and finalisation quarter interaction: This brings out the feature that smaller and larger finalised claims are subject to different rates of superimposed inflation.

- Operational time and accident quarter interaction: This feature resulted from legislative changes that came into effect in September 2000. This legislation placed limitations on the payment of plaintiff costs and effectively eliminated a certain proportion of smaller claims in the system in all subsequent accident quarters.



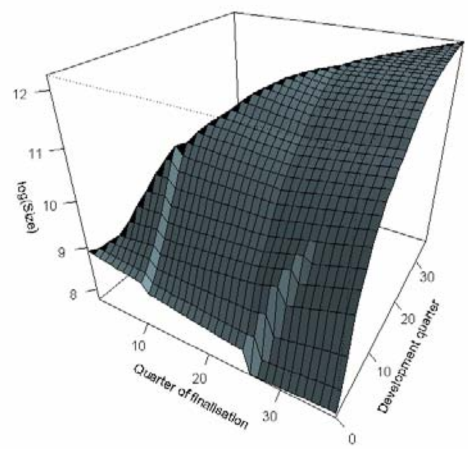


Figure 3.1 Plot of the linear predictor of Taylor and McGuire's GLM model. To smooth these plots I have assumed that the rates of finalisation in each accident quarter are equivalent, and I have ignored the effect of seasonality.

3.3.2 Comparison of models

The results of the soft-computing model fitting exercises are shown in the following six figures. Figures 3.2 and 3.3 show one-way plots of observed and fitted values for quarter of accident and development quarter, respectively. These plots show the average of all observed and fitted values at each value of quarter of accident or development quarter. These plots show that there seems to be no systematic bias in the model fits across accident quarter and development quarter, except for the latest few accident quarters where the data is sparser. Similar plots can be shown for quarter of finalisation and operational time.

Even though there appeared to be no systematic biases in one-dimension, it is still possible that pockets of cells in a two dimensional plot will show systematic differences between observed and fitted values. To test for this possibility, the ratios of observed to fitted values for the accident quarter/development quarter triangles were constructed (Fig. 3.4). In each of these figures, the ratios are colour coded so that ratios greater than 100% are red, and those below 100% are blue.

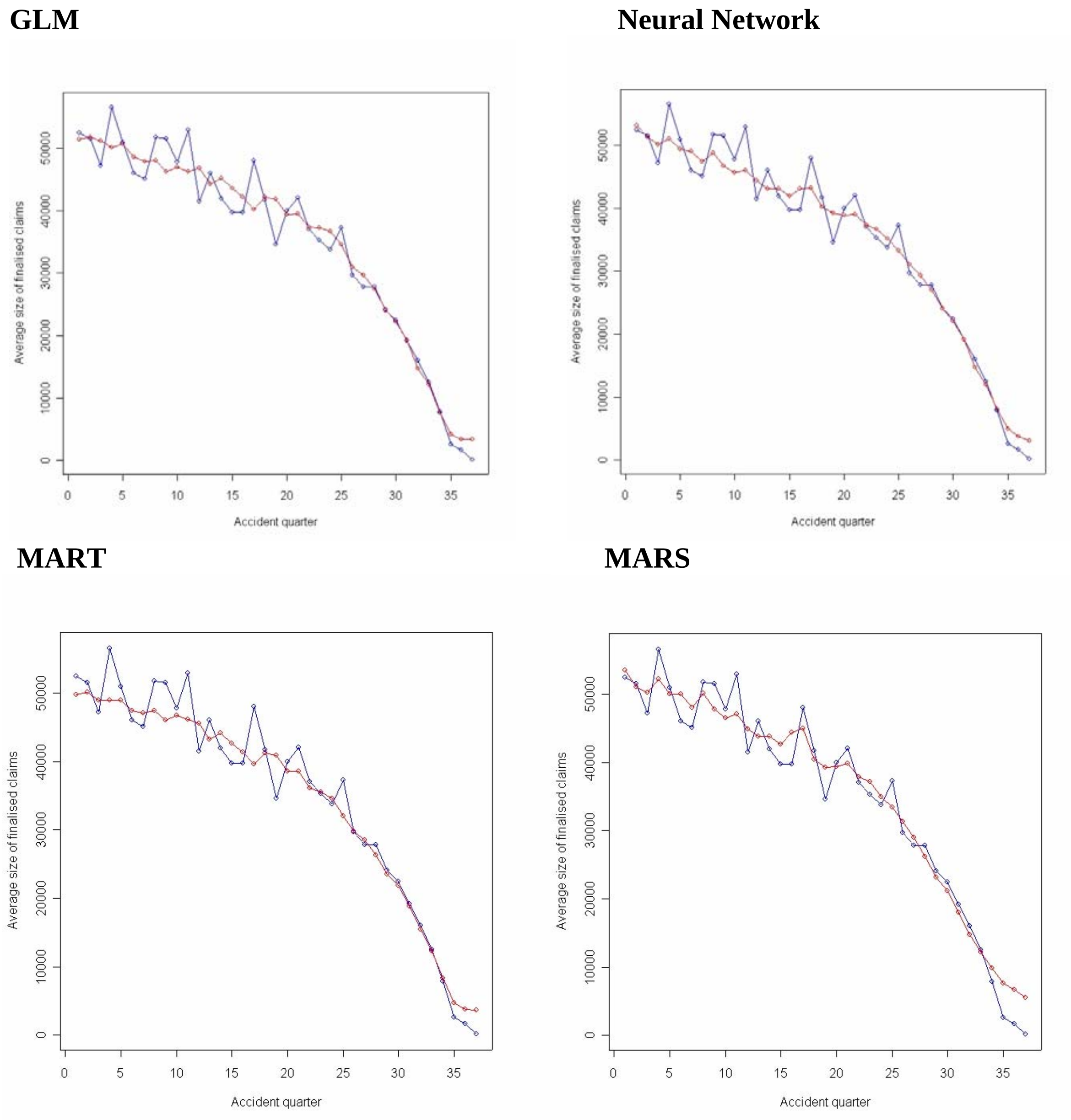
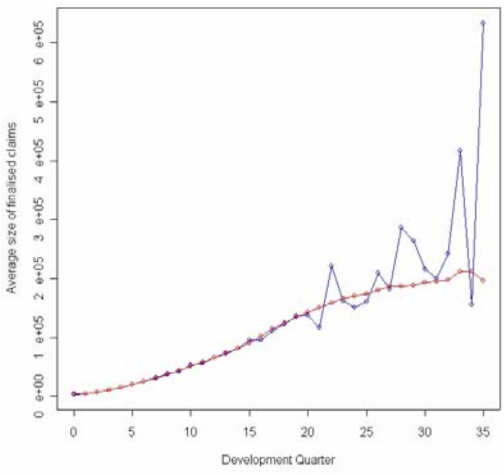
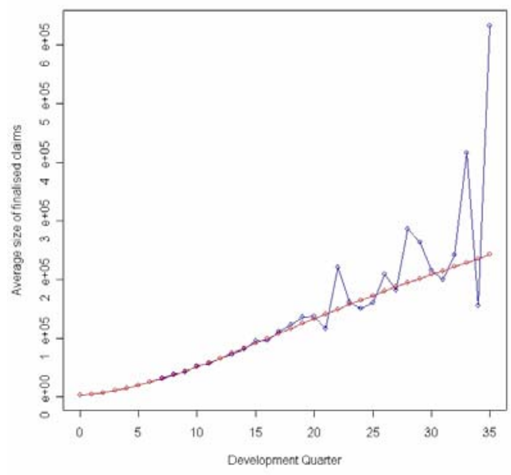


Figure 3.2 One-way tabulations by accident quarter of observed and fitted average finalised claim sizes. All figures: red points = fitted; blue points = observed.

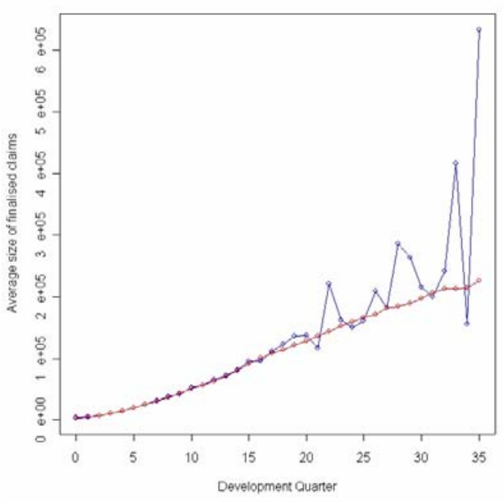
GLM



Neural Network



MART



MARS

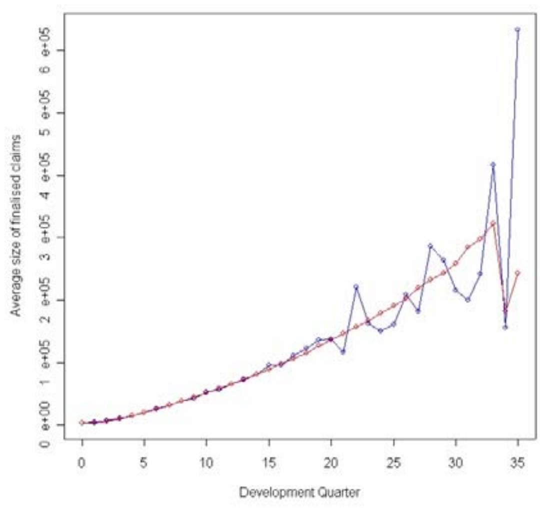


Figure 3.3 One-way tabulations by development quarter of observed and fitted average finalised claim sizes. All figures: red points = fitted; blue points = observed.

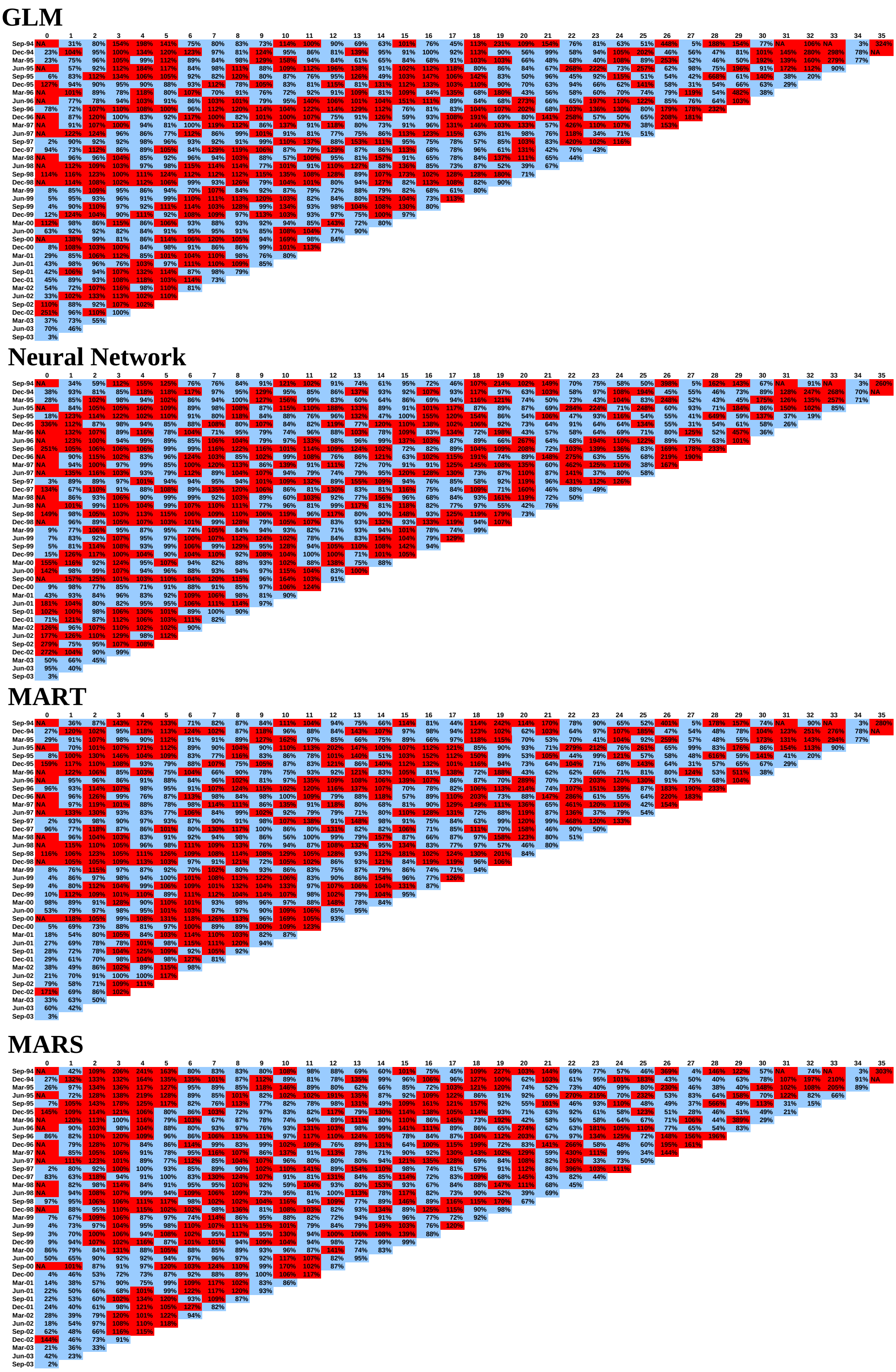


Figure 3.4 Colour coded tabulations of observed to fitted average claim sizes. Tabulations are accident quarter by development quarter. All figures: red squares indicate observed greater than expected; blue squares indicate observed less than expected.

Both the GLM and neural network models show a reasonable random scatter of colour indicating no systematic deviations in model fit. This is less so for the MART and MARS models. For the MART model, the region in the bottom left hand quarter of the triangle shows poor model fit, while for MARS, the entire triangle below development period 6 is a region of poor model fit.

In order to better appreciate the features of the data set that have been modelled by each of the methods, 3 dimensional plots of the regression functions were plotted (Fig. 3.5). For each of the models, two plots were produced. The plot on the left hand side shows the logarithm of the average size of finalised claims plotted as a function of accident quarter and development quarter. This is effectively a three dimensional accident quarter/development quarter triangle. However, note that the plot is not a triangle as the missing part of the triangle has been filled in by projecting with the models.

The plots on the right hand side show the logarithm of the average size of finalised claims as a function of quarter of finalisation and development quarter. These plots are effectively a transformation of the left hand side plots that were created by taking the top left hand corner of each plot and dragging it to the top right hand corner. Note that in these plots only the historical region of the triangle is observed as the projected region has been rotated out of view. The two types of plot allow the features of the regression function to be viewed from different perspectives.

These plots show that the regression functions all have a similar overall shape: however the actual form in each case is constrained by the underlying architecture of the model. For example:

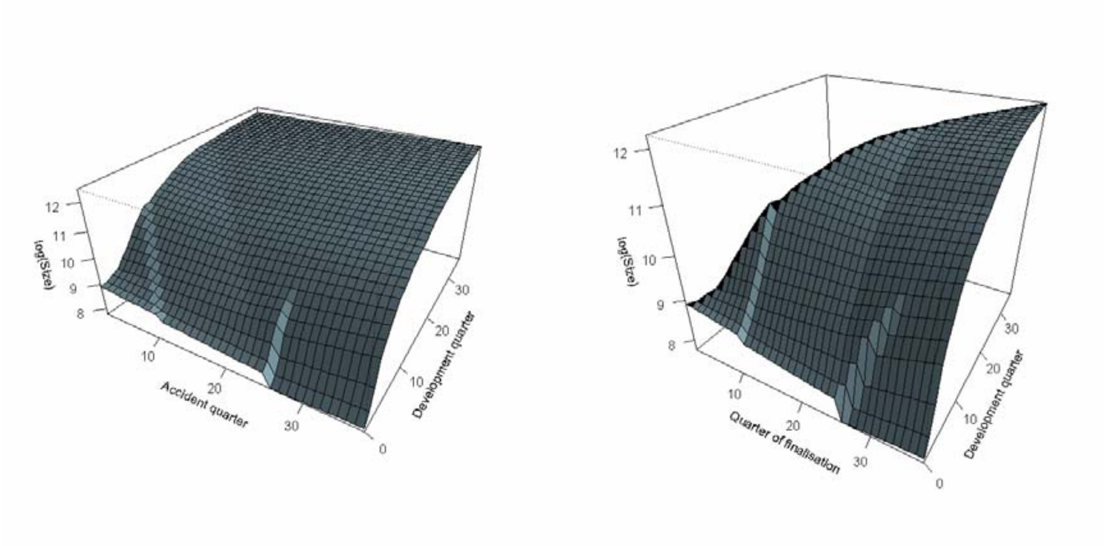
- The linear predictor for the GLM model has been constructed using a mixture of linear splines, interaction terms, and other input transformations. This produces a regression function containing smooth surfaces, discontinuities, and broken trends.

- The neural network model has a single-hidden layer so is constrained to being a smooth continuous surface.

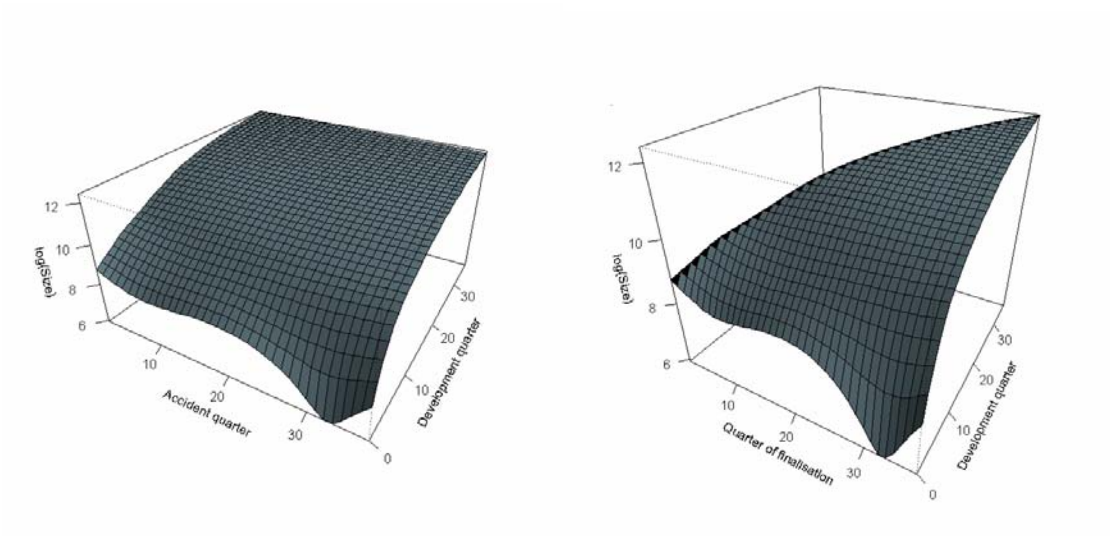
- The MART model is the sum of a number of individual piecewise constant functions and hence is constrained to producing a piecewise constant regression function.

- The MARS model is constrained to a mixture of liner splines and interaction terms constructed out of those splines. Note for space reasons we have not shown the plots of the MARS regression function in Fig. 3.5.

GLM



Neural Network



MART

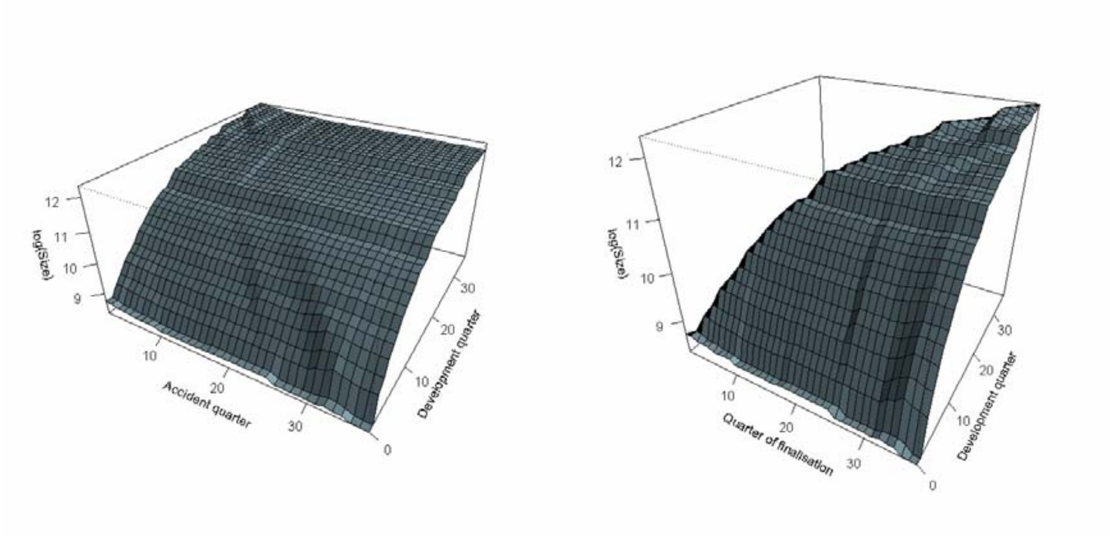


Figure 3.5 Comparison of log(average size of finalised claims) from three models

As a test of the predictive accuracy of the models, each model was fitted to a training data set which consisted of 2/3 of the data. The remaining 1/3 of the data was then used to test the predictive accuracy of each model. Two measures of predictive accuracy were used: the sum of squares of the differences between observed and fitted values in the test set, and the average absolute error of these differences (Table 3.1). The results indicate that with the exception of MARS, the soft-computing techniques outperformed the GLM in predictive accuracy by both measures.

Table 3.1 Test errors for the four regression models

Model	Sum of squares	Average Absolute Error
GLM	2.000×10^{14}	33,777
Neural Network	1.996×10^{14}	33,476
MART	1.999×10^{14}	33,290
MARS	1.994×10^{14}	33,806

3.3.3 Projections of claim size

An important part of any reserving or pricing analysis is to project estimates into future periods. For example, in the Taylor and McGuire paper, the GLM model was used to project the average size of finalised claims into future finalisation quarters for each historical accident quarter. By combining these projections with a model of claims finalisation, estimates of incurred loss by quarter of accident were made.

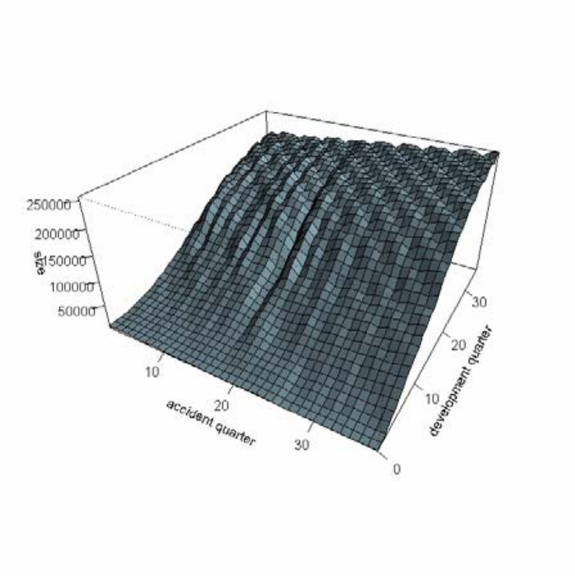
Figure 3.6 shows the projections of the average size of finalised claims for the four models. It is apparent that the projections made by each of the models are quite different; Both the GLM and MARS model project continued superimposed inflation, while both the neural network and MART appear to project negative superimposed inflation.

3.3.4 Use of neural networks in GLM modelling

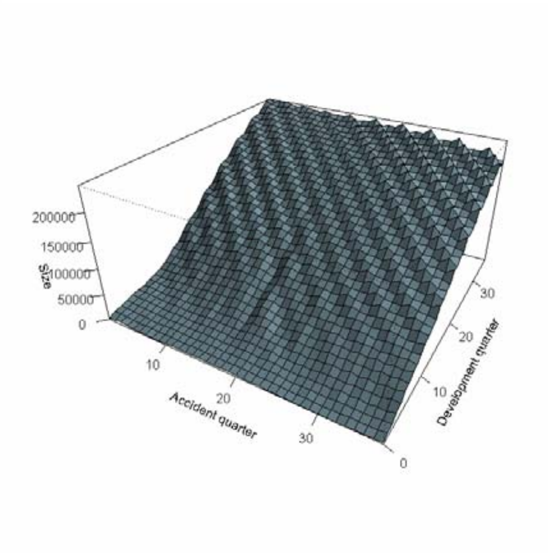
One of the difficulties of GLM modelling is determining the appropriate interactions to include in the GLM regression function. This is an area where the skill of the model builder can play a large part in determining how well the regression function will model the data.

To see whether the adaptive non-linear modelling capability of neural networks could help identify which interactions to include in a GLM model, a neural network was fitted to a residuals from a main effects GLM model. A main effects model is one in which no interaction terms have been included. The results of the analysis are shown in Fig. 3.7. In these plots I have assumed that the rates of finalisation in each accident quarter are the same.

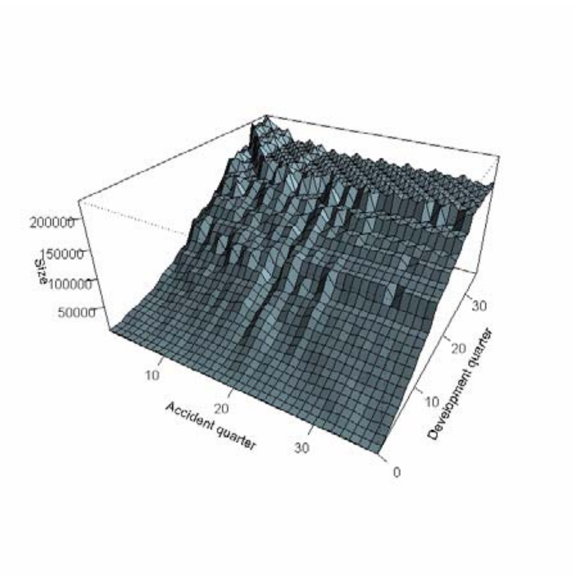
GLM



Neural Network



MART



MARS

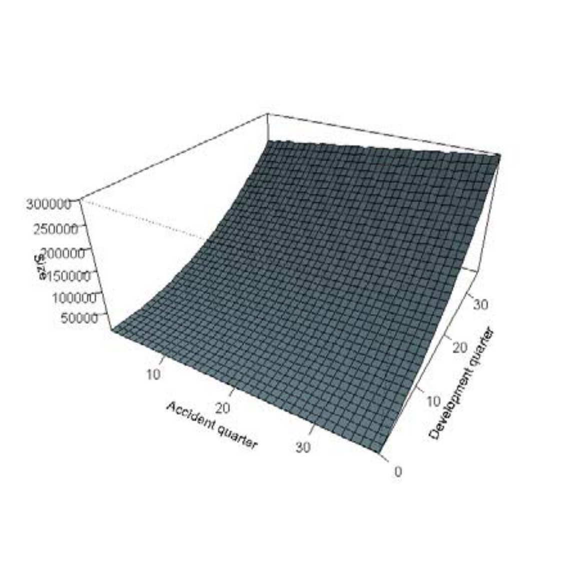


Figure 3.6 Comparison of projections of the average size of finalised claims

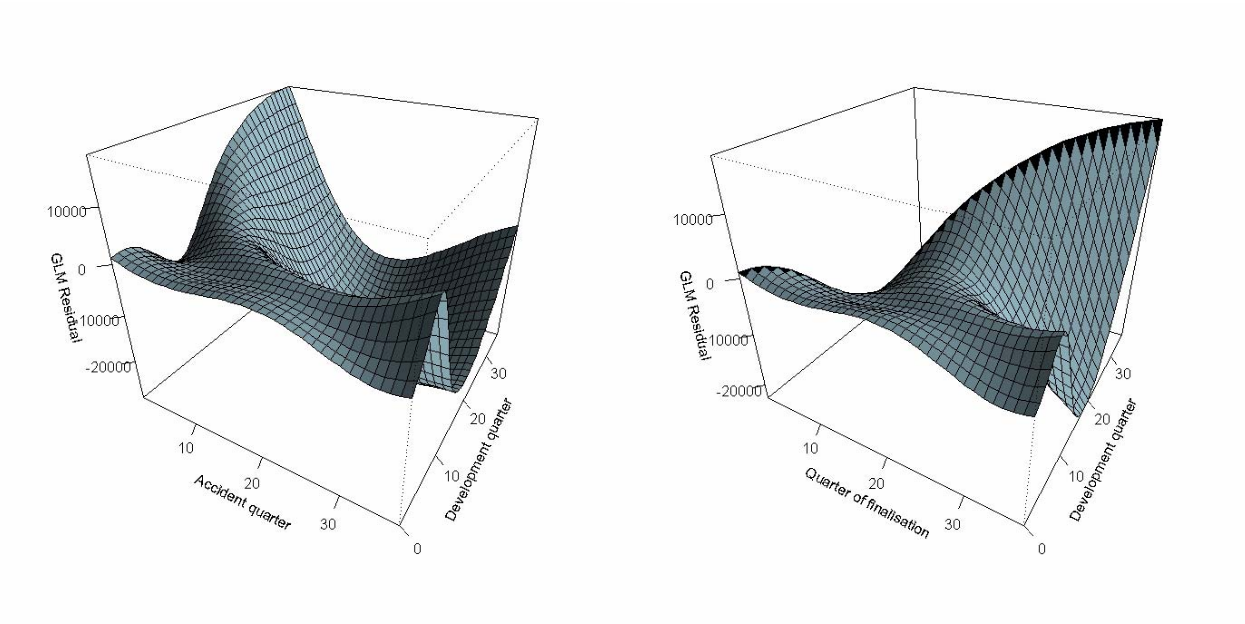


Figure 3.7 Neural Network fit to the residuals from the main effects GLM model

The figure clearly shows that there are some discernable features left in the GLM residuals. The clearest feature is that it appears that there is a strong interaction between quarter of finalisation and development quarter (or operational time). This is clearly seen in the right hand figure. The interaction between development quarter (or operational time) and accident quarter is also seen in the front corner of the left hand figure.

However, while the neural network allows one to visualise the features left in the residuals of a main effects model, it does not translate this into the specific interactions that need to be included in the GLM model. This requires judgement from the modeller and may not always be obvious from plots such as Figure 3.7.

4 Discussion

4.1 Performance of soft-computing methods for the data

Both neural networks and MART were effective in modelling the complex features of the motor injury data set. Both these methods were able to produce sum of squares and average absolute errors of the test data set that were lower than those produced by the GLM model. However, I found MARS to be somewhat less effective.

Although I did not have as much success with the MARS algorithm for this exercise, others, on different problems have found more success (e.g., Kolyshkina et al., 2004). This illustrates, that the success of a particular method depends to a large extent on how appropriate the method's architecture is to the problem. This will not always be apparent at the outset and it often desirable to try a number of different methods.

The regression functions were produced by the soft-computing algorithms in a largely automated manner greatly increasing the speed of model construction. I was able to produce each of the soft computing models in about half a day compared with the one to one and a half days work required for the GLM model. However, the soft computing methods were not *completely* automated. I found that some skill/experimentation was required to get optimal performance out of each algorithm.

A disadvantage of using these largely automated algorithms is that it can be difficult to incorporate external information into model construction. An example of this is the change of legislation that came into effect in the September 2000 quarter. The knowledge of this change influenced the construction of the GLM model and the resultant model showed an abrupt change in the average claim size at early operational times after September 2000. While these changes were detected in the neural network and MART models, these methods did not model the effects of the legislation as effectively as the GLM. Part of the reason for the poor performance appears to be model architecture. For example, the single layer neural network has an architecture which cannot model abrupt changes.

4.2 Projection with soft-computing methods

An area where neural networks and MART performed poorly was projection. An important part of any reserving or pricing analysis is to project estimates into future periods. However, a feature of the neural network and MART regression functions that makes this very difficult is that they are very complex. For example, the neural network regression function that was fitted to the finalised size data had the form of Eqn [2.4] with 161 weight parameters while the MART regression function consisted of 86 regression trees each with 4 parameters. This compares to the 13 parameters in the GLM model.

The complexity of these functions has led some to label these methods as "black box" methods. This "black box" nature makes it difficult to discern what features of the data are being extrapolated and also gives less control over this extrapolation. Also as the regression functions are only fitted over the range of the input values in the data set, the complex nature of the functions means that their behaviour outside the input data ranges will often be hard to predict. In other words, the complex models tend to be less robust for projections.

Hence, projection is an area where GLMs have a clear advantage. The process of manually constructing the regression function for a GLM gives the modeller more control over how the features of the data should be extrapolated into the future. Thus, any the features and trends included in any GLM projection are transparent and explicit.

4.3 GLMs vs soft-computing methods in loss reserving and pricing

Because of the limitations specified above, it seems preferable to use GLM models as the primary tools for performing reserving and pricing projections. However, as demonstrated above, the ability of soft-computing methods to automatically model the complex features of a data set, mean that soft-computing methods may play important roles in model verification and checking.

One way soft-computing methods could be used in model verification is as a general check on the GLM model. If the GLM was giving sums of squares or average absolute errors that were significantly larger than those obtained with the soft-computing techniques, there might be reason to believe that the GLM regression model needed some refinement. A second possible use is to help visualise some of the remaining features in the data after a GLM model has been fitted. This was illustrated in section 3.3.4 and could assist in determining the interaction terms to include in a GLM model.

A final advantage of using GLMs for reserving and pricing projections is that GLMs makes it easier to perform meaningful experience analysis. Because GLMs make specific distributional assumptions about the response variable, it is relatively easy to determine confidence intervals about predictions, and hence to make statistical assessments of whether experience has been significantly different to expected. This kind of analysis is not so readily available when using the soft computing methods described above.

5 Acknowledgements

Greg Taylor and Gráinne McGuire are thanked for valuable discussions and advice.

6 References

Bishop, C. (1995) Neural networks for pattern recognition. Clarendon Press, Oxford.

Friedman (1991) Multivariate adaptive regression splines (with discussion). Annals of Statistics 19, 1-141.

Friedman, J (2001) Greedy function approximation: the gradient boosting machine, Annals of Statistics. 39, 1189-1232

Hastie, T., Tibshirani, R. and Friedman, J. (2001) The elements of statistical learning. Springer-Verlag, New York.

Gershenfeld, N. (1999) The nature of mathematical modelling. Cambridge University Press.

Kolyshkina, I. and Brookes, R. (2002) Data mining approaches to modelling insurance risk. IXth Accident Compensation Seminar.

Kolyshkina, I, Wong, S and Lim, S. (2004) Enhancing generalised linear models with data mining. Casualty Actuarial Society 2004 Discussion Paper Program.

Ripley, B.D. (1996) Pattern recognition and neural networks, Cambridge University Press.

Shapiro, A.F. (2001) Soft computing applications in actuarial science. ARCH 2001.

Shapiro, A.F. (2003) Insurance applications of neural networks, fuzzy logic, and genetic algorithms. In *Intelligent and Other Computational Techniques in Insurance: Theory and Applications*. Shapiro, A.F. and Jain, L.C. eds., World Scientific, Singapore.

Taylor, G. and McGuire, G. (2004) Loss reserving with GLMs: A Case Study. Casualty Actuarial Society 2004 Discussion Paper Program.

7 Appendix – Average sizes of finalised claims

accident	development quarter of finalisation									
quarter	0	1	2	3	4	5	6	7	8	9
Sep-94		2,382	5,594	14,548	23,662	20,845	13,393	15,952	20,187	25,363
Dec-94	1,735	7,483	8,005	9,761	16,670	18,494	21,625	20,482	22,610	36,551
Mar-95	1,636	5,401	8,415	11,250	12,939	16,427	15,306	19,423	23,650	38,433
Jun-95		4,201	8,235	10,865	21,174	15,658	15,338	19,380	26,883	26,601
Sep-95	433	5,741	8,290	12,863	11,326	16,024	14,984	15,849	32,440	27,282
Dec-95	9,060	5,734	6,634	8,514	10,810	11,168	14,994	24,945	21,316	35,263
Mar-96		6,532	7,028	8,476	13,478	12,324	21,493	16,797	26,858	29,239
Jun-96		4,820	6,896	8,456	11,891	13,291	15,259	23,460	30,592	27,762
Sep-96	5,307	4,384	7,214	10,427	13,090	14,603	17,752	28,077	34,566	42,847
Dec-96		3,967	7,915	9,696	9,805	14,188	25,250	24,684	28,015	39,882
Mar-97		4,351	6,578	8,504	11,132	13,294	19,350	32,097	37,282	34,748
Jun-97		6,340	7,701	9,328	12,174	13,587	25,875	25,577	36,860	45,901
Sep-97	73	4,063	6,393	9,849	13,056	16,439	19,525	25,478	33,226	44,113
Dec-97	5,013	3,749	8,501	9,059	12,652	19,397	20,228	39,553	47,096	49,009
Mar-98		4,069	6,720	11,608	11,671	17,624	23,852	30,306	39,476	43,025
Jun-98		5,032	7,769	10,571	13,827	17,887	26,926	31,734	40,262	32,682
Sep-98	5,828	5,832	7,420	10,149	14,871	21,627	23,007	31,026	37,829	45,701
Dec-98		5,181	6,660	11,127	16,982	20,827	28,255	33,628	56,021	37,895
Mar-99	401	3,986	8,292	11,595	14,073	21,779	21,256	39,558	35,930	45,572
Jun-99	111	4,363	7,990	11,984	15,102	21,380	30,718	35,818	43,731	53,315
Sep-99	97	4,207	7,420	11,354	14,234	21,926	25,532	29,806	44,528	43,578
Dec-99	547	5,663	8,785	11,578	18,106	17,596	26,086	31,767	35,942	48,759
Mar-00	5,050	4,509	6,763	14,539	13,408	20,166	22,155	27,151	34,228	43,820
Jun-00	1,940	3,922	6,948	10,001	13,678	18,518	26,127	31,930	39,958	45,756
Sep-00		6,157	6,876	9,850	13,739	23,564	26,500	41,375	45,000	48,456
Dec-00	147	2,464	3,807	7,235	10,462	16,988	22,767	27,905	35,336	47,889
Mar-01	396	2,231	4,251	9,510	11,317	20,115	29,005	39,125	41,250	39,670
Jun-01	1,271	3,060	5,628	7,615	15,559	20,652	33,051	40,138	49,832	43,731
Sep-01	898	3,317	4,878	11,581	21,188	25,352	25,003	37,460	35,387	
Dec-01	678	2,463	4,966	10,511	18,989	21,111	35,324	28,008		
Mar-02	1,594	2,429	6,579	14,210	15,408	26,188	26,690			
Jun-02	1,017	3,443	7,947	11,497	17,380	23,825				
Sep-02	1,394	3,072	5,600	14,098	18,272					
Dec-02	8,102	2,905	6,081	10,007						
Mar-03	1,013	2,392	2,652							
Jun-03	2,327	1,400								
Sep-03	59									

accident quarter	development quarter of finalisation									
	10	11	12	13	14	15	16	17	18	19
Sep-94	38,981	45,863	43,155	40,948	35,908	79,834	65,785	37,000	102,906	262,773
Dec-94	33,065	32,703	37,365	78,315	69,108	72,480	94,207	93,147	139,178	114,821
Mar-95	57,613	41,147	42,031	37,687	45,913	68,820	64,777	104,722	134,404	141,225
Jun-95	41,032	49,272	101,466	78,637	64,070	75,045	97,381	117,429	92,234	110,545
Sep-95	33,960	36,892	53,606	82,564	36,676	89,167	149,679	123,365	178,941	108,499
Dec-95	34,086	40,123	65,550	53,308	93,931	94,785	131,331	106,164	123,663	115,566
Mar-96	30,093	47,875	50,320	73,979	60,599	95,555	79,094	153,636	84,868	216,397
Jun-96	39,983	67,740	60,913	67,250	78,248	121,485	105,823	89,606	100,151	81,515
Sep-96	42,509	60,787	64,080	84,732	81,474	69,821	82,547	93,824	121,783	142,849
Dec-96	46,771	58,639	49,350	64,319	104,134	57,695	100,423	115,642	218,638	92,438
Mar-97	66,464	51,580	70,439	57,971	58,495	82,164	87,609	141,490	168,964	130,546
Jun-97	48,392	45,413	52,464	58,820	78,146	109,606	137,301	142,201	83,567	109,264
Sep-97	55,427	83,651	57,813	114,059	88,783	75,094	75,506	88,514	65,035	119,652
Dec-97	48,775	48,073	89,093	63,252	72,469	101,136	73,730	93,165	134,379	76,378
Mar-98	31,026	63,288	60,886	57,705	128,024	86,192	62,702	95,530	108,612	174,022
Jun-98	46,875	47,891	63,811	76,160	63,667	106,801	81,854	81,477	108,414	69,581
Sep-98	55,025	54,074	71,146	55,322	74,652	134,479	91,283	126,607	141,252	217,591
Dec-98	58,810	63,307	57,124	72,602	116,566	86,935	130,781	122,363	110,631	134,531
Mar-99	46,377	49,490	51,105	75,895	80,459	92,833	83,014	81,526	118,238	
Jun-99	52,992	47,720	57,234	62,371	126,934	96,335	82,298	141,547		
Sep-99	67,518	59,729	69,221	83,279	94,824	138,272	94,214			
Dec-99	56,099	57,395	69,656	55,780	89,280	94,104				
Mar-00	53,568	55,077	101,005	55,766	74,750					
Jun-00	65,777	70,131	59,887	77,433						
Sep-00	93,585	64,453	60,465							
Dec-00	58,789	73,588								
Mar-01	49,258									
Jun-01										

accident quarter	development quarter of finalisation									
	20	21	22	23	24	25	26	27	28	29
Sep-94	129,349	195,633	101,579	121,257	95,978	84,158	712,264	8,764	316,169	282,728
Dec-94	79,351	135,363	90,855	145,021	172,933	335,296	84,477	104,109	87,098	146,445
Mar-95	97,834	70,355	110,518	65,824	172,169	148,164	452,372	84,431	84,166	78,663
Jun-95	121,116	98,383	411,325	352,690	121,433	431,789	104,799	174,005	128,565	374,070
Sep-95	73,437	144,988	71,009	152,943	193,406	91,006	98,609	70,139	1,273,300	116,584
Dec-95	94,241	90,905	141,854	100,275	102,849	230,998	102,517	59,253	103,639	105,413
Mar-96	56,632	84,235	75,801	97,496	114,575	126,450	144,496	227,986	91,890	780,581
Jun-96	371,135	90,167	98,013	304,316	188,162	210,869	156,466	140,817	124,941	202,313
Sep-96	277,303	97,799	153,110	215,742	225,600	139,336	259,511	339,882	362,793	
Dec-96	110,092	209,851	426,546	93,455	87,416	116,867	323,074	355,008		
Mar-97	168,714	85,802	692,725	190,969	181,670	51,736	300,118			
Jun-97	153,029	124,170	205,154	58,125	118,166	87,164				
Sep-97	159,195	130,970	644,990	179,757	206,863					
Dec-97	207,985	65,860	135,046	61,946						
Mar-98	159,924	97,588	74,291							
Jun-98	54,887	107,701								
Sep-98	94,013									
Dec-98										

accident quarter	development quarter of finalisation						
	30	31	32	33	34	35	36
Sep-94	139,507		201,849		6,200		633,545
Dec-94	191,107	276,459	537,824	608,937	166,449		
Mar-95	366,509	265,796	297,888	595,605	165,077		
Jun-95	174,706	320,567	228,614	192,673			
Sep-95	285,658	81,822	41,975				
Dec-95	123,129	56,756					
Mar-96	73,749						
Jun-96							