Coil 2000 Competition
The use of a learning classifier system JXCS

Andrew Greenyer
The Database Group,
Colston Tower, Colston Street, Bristol, BS1 4UH UK
Phone: +44 1179 183500, Fax: +44 1179 183501
Email: andrewg@databasegroup.co.uk

ABSTRACT: It is now widely accepted that large companies need to be able to emulate the old fashion corner shop: where the shopkeeper knew all his customers by name, what they liked, what they disliked and was able to recommend certain products based on his knowledge. The COIL 2000 competition emulates this situation with an insurance company trying to identify caravan insurance buyers from other insurance buyers. Traditionally direct marketing companies have used statistical techniques such as linear regression, decision trees such as CHAID and cart, through to neural networks to predict which customers are likely to respond or purchase a particular product. Our entry for the competition used a new system that has been developed jointly by University of the West of England and The Database Group, under a project funded by the UK Department of Trade. The system is based on a form of learning classifier system, known as XCS [1]. The system is designed to produce rule based criteria for identify best prospects for marketing activity.

KEYWORD: Direct Marketing, Learning Classifier Systems, Genetic Algorithms

1 Introduction

The Database Group is a computer services bureau supplying services to the direct marketing industry. Its analysis division worked in the traditional statistics arena for producing models to predict and describe customers who were most likely respond to a marketing activity. In 1998 The Database Group joined forces with the Faculty of Computer Studies and Mathematics at the University of the West of England, to develop a new data mining program based on genetic algorithms and learning classifier systems.

The learning classifier system used for the project was based on the XCS Classifier System, Wilson [1], which was further developed by Barry [2] into a Java based program called JXCS.

This paper will discuss in an overview form the learning classifier system, and the use of the JXCS system in approaching the COIL 2000 Competition problem.

2 The JXCS Learning Classifier System

The JXCS Learning Classifier System has the following schematic (Figure 1):

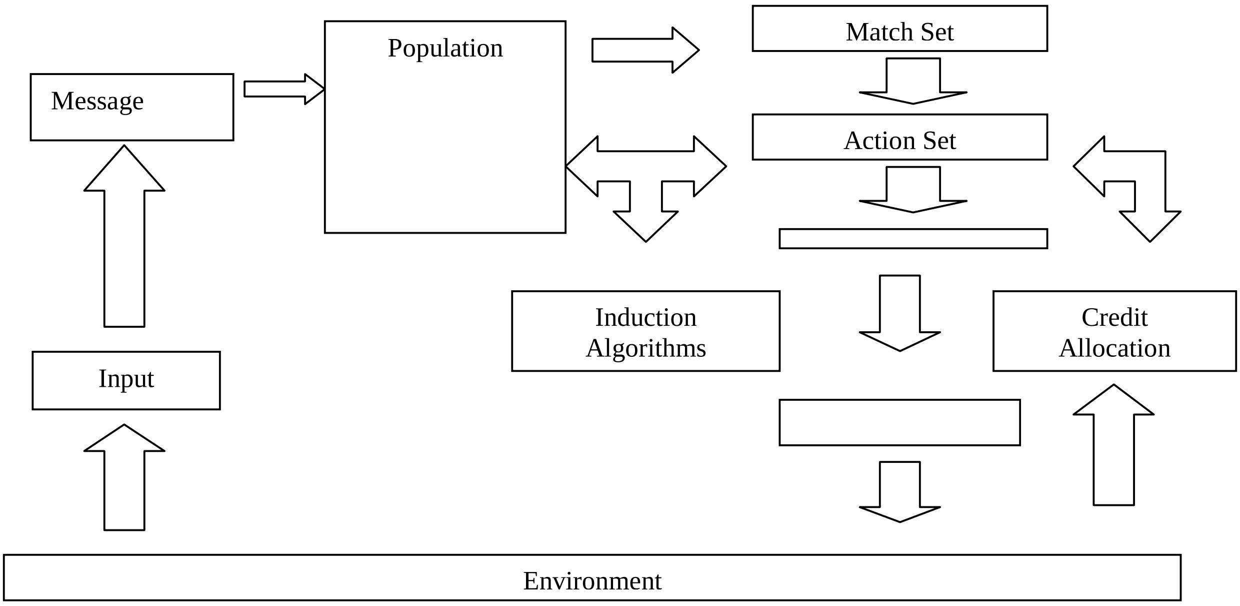

Figure 1 : Schematic of JXCS Learning Classifier System

The process of the JXCS starts with two sets of data: The first is the environment which contains the records extracted from the database for analysis split into a training and test set with a variable defined as an action (known as the dependent variable in statistical terms) and the condition (independent variables). The second set of data is the population, which contains the classifiers. The population may be empty at the start of the analysis run or randomly populated with records extracted from the environment. The population has the maximum number of classifiers it may contain defined. Each classifier is defined by a condition, resultant action, numerosity (number of multiple occurrences of classifier) and various predictive accuracy measures.
Each iteration of the JXCS starts by randomly selecting a record from the environment training set - this is known as the input message. The structure of this message is an ordered vector of attribute values, each of which has one of the following data model types:

Nominal – a discrete value chosen from a finite set of unique unordered values
Ordinal – a discrete value chosen from a finite set of unique ordered values
Continuous – a fixed interval in a continuous range
Binary String – a fixed length string of binary digits

The environment provides a reward back to the JXCS. This is a function of both actual action of the current record and the predicted action by JXCS. The default scheme is to give maximum reward to a correct prediction and a minimum reward to an incorrect prediction.
Each classifier is associated with a record of its performance; values are kept that are measures of ability to predict the reward for the environment state and action pairs. There are four estimates maintained, the first of which is prediction, this is an estimate of the reward that the system will receive from the environment given that the action specified by the classifier has been performed when the environment is in one of the states given by the classifiers condition. In other words it is an estimate of the proportion of examples that the classifier classifies correctly. Prediction Error can then be calculated from the absolute difference between the predicted and the actual reward a classifier receives. Accuracy is derived from the prediction error and fitness is the accuracy of a classifier relative to those classifiers with the same action that are active at the same time.

An important aspect of the representation of the population within JXCS is the use of classifier numerosity. Classifiers with the same condition and the same action are considered to be the same classifier and share the performance measure described above. The numerosity of a classifier is effectively the number of copies of a single classifier present in the population, so the population is stored as a set of unique macro-classifiers with numerosities. Therefore, when calculating population size and averages of classifier parameters the numerosities are taken into account. This distinguishing feature of JXCS necessitates the use of two new terms: macro-classifiers and micro-classifiers. A macro-classifier has the constituent parts of a classifier as described above including a numerosity (n) and it is interpreted as consisting of n copies of a micro-classifier with the given condition, action and performance statistics. Using this representation eases the identification of 'good' classifiers, which would manifest themselves as macro-classifiers with large numerosities.

An iteration begins by comparing the selected record from the environment with the macro classifiers from the match set (M). If the match set is empty a new macro classifier is created based upon the input message and a random action with numerosity 1. The classifiers in the match set are partitioned by action to form action sets (A), and for each action set the system prediction of the action set is calculated as a fitness-weighted measure. The system prediction is the sum of products of the prediction and fitness of each classifier divided by the sum of fitnesses.

The JXCS runs in two modes of operation, explore and exploit. In the explore mode, the system seeks to broaden its search for classifiers so it chooses an action at random from those proposed within the current action sets, while in the exploit mode the system seeks to find the best classifiers so it chooses the action set with the highest prediction from within the match set.

The selected action is sent to the environment to test its classifying performance and a reward is returned dependent on a reward function and the performance values of all classifiers within the selected action set are updated.

If the system is in exploit mode the system can then report on the overall system performance as learning doesn't occur. This reporting also occurs in a third test mode, which is used for testing the system on the preclassified examples that were not in the training set, namely the test set. The standard measures reported are moving averages of the system prediction, the system error (the absolute difference between the system prediction of the action set and the received reward), the number of macro-classifiers in the population and, if a unique set of optimally general classifiers is available, the percentage of this set present in the population.

If the system is in explore mode then a number of rule induction operators are run if they are triggered. These operators are described in more detail below.

The rule induction operators may create more micro-classifiers than the population allows, so a deletion technique is applied to remove the surplus classifiers. The classifier chosen for deletion is a macro-classifier and it is removed from the population if its numerosity is one, otherwise the numerosity is reduced by one.

Once the system has updated the population after receiving a reward for the random action set chosen on an exploration step, the system will use none, either or both of two induction operators depending on the current strength of the classifiers. The Create Detector operator is the third induction algorithm, which is triggered by a failure to match the current message to any of the existing classifiers. There follows a description of each these algorithms and the conditions under which they are triggered.

On both exploration and exploitation steps the Create Detector operator is fired when no classifier in the current population matches the environmental message. The operator creates a new classifier, which matches this message, and which has been generalised using a fixed user-defined generality.

When the mean prediction of matched classifiers falls below a fraction of the initial prediction given to new classifiers then the Create Effector operator is triggered. This mostly works the same as the Create Detector operator by producing a new classifier that matches the current message and has been generalised using a fixed generality. The difference is that the action is guaranteed to be different to the action used in the current action set which ensures that the operator is, in fact, creating a new effector in order to enlarge the exploration space. In fact, in the kinds of problems that JXCS encounters the Create Effector operator is rarely, if ever, invoked.

One of the user-defined parameters of JXCS is the GA invocation frequency. A count of each time a micro-classifier is updated by the credit allocation algorithm is kept and is reset for all those classifiers, which are in the current action set when the GA is triggered. The GA is invoked when the average count for the current action set (allowing for numerosity) is greater than the threshold invocation frequency.

The GA consists of two operators: crossover and mutation. Two classifiers from the current Action Set are selected with replacement (allowing duplicates) using the macro-classifiers' fitnesses as weights in Roulette Wheel Selection. Single point crossover is applied with a fixed probability (usually 0.8) to the conditions of these classifiers to produce two child classifiers. If crossover is not triggered the parents are copied to produce the children. Mutation then occurs on both the condition and the action. Each attribute is mutated with a fixed user-defined probability and, in general, increases or decreases the number of values that the attribute matches by a number between 0 and a fixed maximum. The action, a single-valued ordinal attribute, is mutated by generating a uniformly random value. The prediction for each child is either set as the average of parents' predictions if crossover has been applied or is set to the prediction of a different parent. The child error is set to one quarter of the average population error and the fitness is set to one tenth of the average population fitness so that newly created classifiers do not contribute greatly to the calculation of system prediction or the operation of the GA until they have been evaluated a sufficient number of times.

3 COMPETITION METHODOLOGY

The evaluation file of known caravan insurance buyers was first processed through commercially available modelling software, Model 1, available from Group 1 Software. This allows large numbers of variables to be evaluated in terms of their significance and sensitivity in prediction. The modelling process used a variety of traditional statistics such as regression, CHAID, CART, Bayesian probability and neural networks. The most predictive model generated was a cross validated neural network with the objective set to identifying as many caravan buyers in the top 20% of the database. The resultant model produced contained 1 hidden layer with 9 nodes and 38 inputs; this resulted in 193 known caravan insurance buyers being identified within the top 20% of the file, 55.5% of the total buyers.

The most significant variables used in the model were:

Variable	Variable Description	Importance %	Sensitivity %
APERSAUT	Number of Car Policies	1.95	6.87
PPERSAUT	Contribution Car Policies	1.99	6.79
PBRAND	Contribution Fire Policies	1.95	5.06
MINKGEM	Average Income	1.91	4.98
PWAPART	Contribution Third Party Insurance	1.90	4.87
AWAPART	Number Third Party Insurance	1.89	4.84
MKOOPKLA	Purchasing Power Class	1.91	4.70
MOSHOOFD	Customer Main Type	1.92	4.58
MOSTYPE	Customer Sub Type	1.92	4.45
MHHUUR	Rented House	1.89	4.30
MHKOOP	Home Owners	1.89	4.30
MAUT0	No Car	1.89	4.15
MOPLLAAG	Lower Level Education	1.90	4.10
MINKM30	Income < 30,000	1.90	3.85
ABRAND	Number of Fire Policies	1.88	3.68
MINK4575	Income 45-75,000	1.89	3.64
MAUT1	1 Car	1.89	3.48
MOPLHOOG	High Status	1.89	3.41
MRELGE	Married	1.89	2.97
MBERHOOG	High Level Education	1.88	2.95
MSKA	Social Class A	1.89	2.88
MRELOV	Other Relation	1.88	2.88
MSKC	Social Class C	1.88	2.74
MBERARBG	Skilled Labourers	1.88	2.15
APLEZIER	Number of Boat Policies	1.88	0.70
PPLEZIER	Contribution Boat Policies	1.89	0.68

These variables were input into the JXCS software to further refine the modelling process. After multiple passes through the software the best results were achieved using the variables; PBRAND, MOSHOOFD, MOSTYPE, PPERSAUT and APERSAUT, the resultant accuracy curves generated are shown in figure 2.

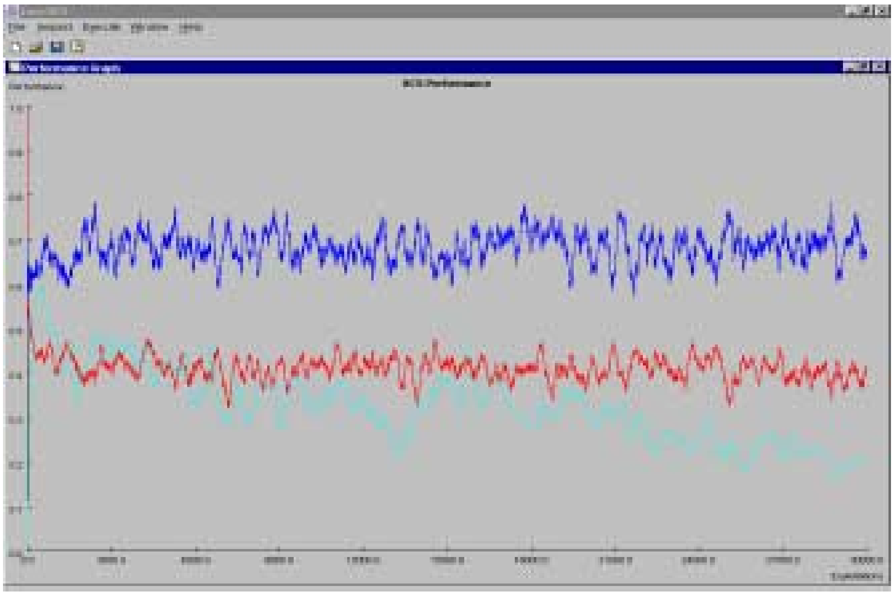

Figure 2: Accuracy Curves from JXCS

The accuracy curve (dark blue) shows the model levels out after approximately 3,000 iterations and remains around the 70% level, with the corresponding error curve (red) around the 40% level. The light blue curve represents the percentage of the maximum number of classifiers contained within the classifier population, we can observe that the number of classifiers required decreases over time although the accuracy remains level. The number of classifiers finally settled at 22. Out of the 22 rules generated 9 were significant in their prediction of a caravan buyer, by applying these rule to the total file identified 193 caravan insurance buyers out of 1159 records giving a 16.7% response rate.

To score the selection file both the neural network and the JXCS rules were applied and 800 records were selected to be submitted based on being in the top 30% of records according to the neural network and identified by the JXCS rules as being likely caravan insurance buyers.

This resulted in 112 policy records being identified from the 800 records submitted giving a response rate of 14% and placing our submission in joint third place in the competition.

4 CONCLUSION

In the application of JXCS to this real world problem we have been able to illustrate an improvement in the overall accuracy of applying the rule set generated, when JXCS is used in conjunction with the learning achieved through the statistical methods.

Further research is now being undertaken in the use of continuous data variables within JXCS, processing large volumes of commercial data, conversion of rules generated into an easily readable form and being able to use directly, learning achieved through our traditional statistic routines. We are also examining other fitness functions that smooth out the predicted results and associated errors achieved by adding in extra rules.

References

[1] Wilson, S.W., *Generalisation in the XCS classifier system*, GP98, 1998

[2] Barry, A, *The XCS Classifier, Technical report*, Faculty of Computer Studies and Maths, UWE, UK, 1998

[3] Greenyer, A, *The use of learning classifier systems in the UK direct marketing industry*, Data Mining 2000 Conference