Paper category: Real-World Applications (RWA)

"Evolutionary, my dear Watson"
Investigating Committee-based Evolution of Fuzzy Rules for the Detection of Suspicious Insurance Claims

Peter J. Bentley

Department of Computer Science
University College London,
Gower Street, London WC1E 6BT
P.Bentley@cs.ucl.ac.uk
Tel. +44 (0)20 7679 1329

Abstract

Fraud is causing huge losses to our financial world. This work describes the use of a committee decision system using genetic programming to evolve fuzzy logic rules capable of detecting suspicious home insurance claims. Details of the evolutionary-fuzzy system, the committee decision maker and the data preprocessing are provided. Finally, a series of experiments are described, showing that the complete system is capable of attaining good accuracy and intelligibility levels for real home insurance data.

1 INTRODUCTION

Despite his prowess at solving crimes, and despite the fact that 221b Baker Street is only a short walk from University College London, even the world famous detection skills of Sherlock Holmes would surely be unable to keep up with the problems of fraud in today's world. Looking at home insurance claims alone, with tens of thousands of new claims every month, it is simply not humanly possible to check every one thoroughly. And when around one in twenty[1] claims may be "suspicious", this inevitably results in losses of significant sums.

The only viable solution to problems of this scale is automation by computer. Just as computers are used for credit scoring, risk assessment and customer profiling, it is possible to use computers to assess the likelihood of insurance claims being "suspicious".

Such automated detection can be performed by using simple statistical techniques, or by applying 'rules of thumb' to claims. However, the fingerprints of fraudulent activity may be diverse and complex, resulting in the failure of these traditional methods. This motivates the use of newer techniques, called machine learning or *pattern classification*, which are capable of finding complex non-linear 'fingerprints' in data.

This paper investigates one such technique: the use of genetic programming to evolve fuzzy logic rules capable of classifying home insurance claims into "suspicious" and "non-suspicious" classes. The paper follows on from (Bentley, 1999), describing extensions to the system in the form of parallel processing and committee decision making. The paper also focuses on the use of this system for classification of real home insurance data. However, the reader is warned that because of the nature of the data and evolved results, this article will not report details that may compromise confidentiality agreements.

The next section provides a brief background to the topics described in this paper. Section three summarises the evolutionary-fuzzy system described in (Bentley, 1999), and section four describes the new committee decision extensions and other additions to the system. Section five outlines the extensive data cleaning and preprocessing necessary, and explains the experiments performed, showing how the results have been analysed. Finally, section six provides conclusions.

2 BACKGROUND

Machine Learning, pattern classification and data mining are huge fields in Computer Science, with countless different techniques in use or under investigation. This paper concentrates on a single approach: the use of *fuzzy logic* with *genetic programming* to classify data.

Fuzzy sets were introduced by Lofti Zadeh in 1965 (Zadeh, 1965). Designed to allow the representation of 'vagueness' and uncertainty that conventional set theory disallowed, the sets and their manipulation by logical operators led to the development of the field known as Fuzzy Logic (Bezdek and Pal, 1992). Despite the name, fuzzy techniques are actually capable of greater precision compared to classical approaches (Kosco, 1994). Fuzzy

[1] According to information provided by Lloyds TSB PLC

controllers have been used with considerable success: examples include controllers for elevators, subway trains, and even fuzzy autofocus systems for cameras (Mc. Neill and Freiberger, 1993).

Another appeal of fuzzy logic is its *intelligibility*. Fuzzy rules use linguistic identifiers such as ‘high’, ‘short’ and ‘inexpensive’. Because all humans tend to think in such vague terms, the specification and understandability of such rules becomes simple, even to someone unaware of the mechanisms behind this technique (Kosco, 1994). The combination of representation of uncertainty, precision, and intelligibility has motivated the use of fuzzy logic in pattern classification (Bezdek and Pal, 1992), and indeed, forms the motivation for its use in this research.

Fuzzy logic can be combined or hybridized with many other techniques, including evolutionary algorithms. Some have developed fuzzy-evolutionary systems (Pedrycz, 1997) where fuzzy logic is used to tune parameters of an evolutionary algorithm. Others use evolutionary-fuzzy approaches, where evolution is employed to generate or affect fuzzy rules (Mallinson and Bentley, 1999; Marmelstein and Lamont, 1998). This paper describes the latter approach, and makes use of Genetic Programming (GP).

John Koza (1992) developed GP for the purposes of automatic programming (making computers program themselves). GP differs from other EAs in three main respects: solutions are represented by tree-structures, crossover normally generates offspring by concatenating random subtrees from the parents, and solutions are evaluated by *executing* them and assessing their function.

Like all evolutionary algorithms (EAs), GP maintains *populations* of solutions. These are evaluated, the best are selected and ‘offspring’ that inherit features from their ‘parents’ are created using crossover and mutation operators. The new solutions are then evaluated, the best are selected, and so on, until a good solution has evolved, or a specific number of generations have passed.

EAs are often used for pattern classification problems (Koza et al, 1998), but although the accuracy can be impressive, it is often difficult to understand the evolved method of classification. By evolving fuzzy rules it is possible to get the best of both worlds - accurate and intelligible classification (Mallinson and Bentley, 1999).

However, despite the success of such systems, as with any technique, the performance varies as the data changes and as different control parameters are used. Previous work has shown that the choice of clusterer, fuzzy membership functions, fuzzy interpreter, genetic algorithm mutation rates, population sizes and number of generations will all affect quality of classification (Bentley, 1999). Ideally, analysis of the data would predict which system elements are likely to be most effective, however to date, reliable prediction of this kind is not available. The solution is to run different models with different setups on the same data and to use decision aggregation to find and present the best results (Bunn, 1989). This paper describes the use of such a committee decision approach, by showing how multiple versions of a fuzzy evolutionary system using different setups can be run in parallel, with decision aggregation enabling prediction reliability to be maintained for different problems.

3 SYSTEM OVERVIEW

Before describing the committee decision extensions, this section briefly describes the evolutionary fuzzy system used (with different setups) as members of the committee. Full details of this system can be found in (Bentley, 1999).

The system developed during this research comprises two main elements: a Genetic Programming (GP) search algorithm and a fuzzy expert system. Figure 1 provides an overview.

3.1 CLUSTERING

Data is provided to the system in the form of two comma-separated-variable (CSV) files: training data and test data.

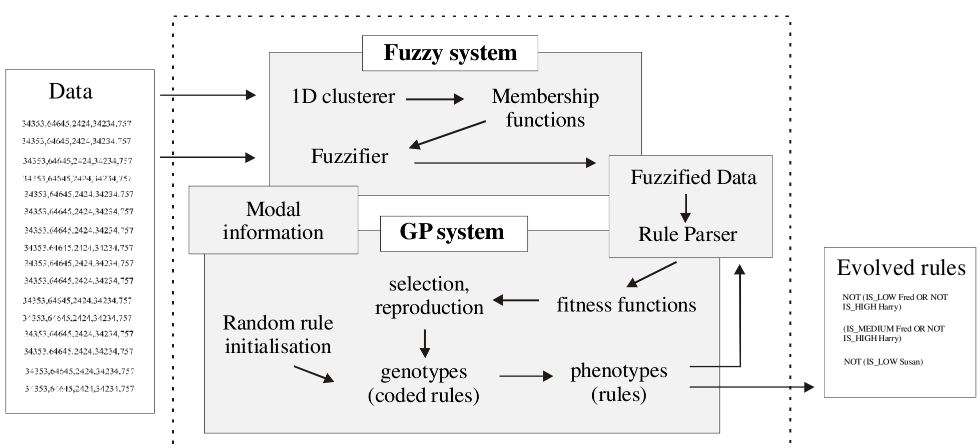

Figure 1 Block diagram of the Evolutionary-fuzzy system.

When started, the system first clusters each column of the training data into three groups using a one-dimensional clustering algorithm. A number of clusterers are implemented in the system, including C-Link, S-Link, K-means (Hartigan, 1975) and a simple numerical method (in which the data is sorted, then simply divided into three groups with the same number of items in each group).

After every column of the data has been successfully clustered into three, the minimum and maximum values in each cluster are found. These values are then used to define the domains of the membership functions of the fuzzy expert system.

3.2 FUZZY MEMBERSHIP FUNCTIONS

Three membership functions, corresponding to the three groups generated by the clusterer, are used for each column of data. Each membership function defines the 'degree of membership' of every data value in each of the three fuzzy sets: 'LOW', 'MEDIUM' and 'HIGH' for its corresponding column of data. Since every column is clustered separately, with the clustering determining the domains of the three membership functions, every column of data has its own, unique set of three functions.

The system can use one of three types of membership function: 'non-overlapping', 'overlapping', and 'smooth' (Bentley, 1999). The first two are standard trapezoidal functions, the third is a set of functions based on the arctangent of the input in order to provide a smoother, more gradual set of 'degree of memberships'.

Whichever set of membership functions are selected, they are then shaped according to the clusterer and used to fuzzify all input values, resulting in a new database of fuzzy values. The GP engine is then seeded with random genotypes (coded rules) and evolution is initiated.

3.3 EVOLVING RULES

The implementation of the GP algorithm is perhaps best described as a genetic algorithmist's interpretation of GP, since it employs many of the techniques used in GAs to overcome some of the problems associated with simple GP systems. For example, this evolutionary algorithm uses a crossover operator designed to minimise the disruption caused by standard GP crossover, it uses a multiobjective fitness ranking method to allow solutions which satisfy multiple criteria to be evolved, and it also uses binary genotypes which are mapped to phenotypes.

3.3.1 Genotypes and Phenotypes

Genotypes consist of variable sized trees, where each node consists of a binary number and a flag defining whether the node is binary, unary or a leaf, see figure 2. At the start of evolution, random genotypes are created (usually containing no more than 3 binary and 4 unary nodes). Genotypes are mapped onto phenotypes to obtain fuzzy rules, e.g. the genotype shown in fig. 2 maps onto the phenotype:

"(IS_MEDIUM (Height OR IS_LOW Age) AND IS_MEDIUM Age)".

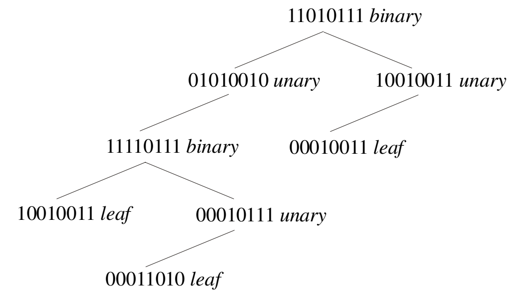

Figure 2: An example genotype used by the system.

Currently the system uses two binary functions: 'OR' and 'AND', four unary functions: 'NOT', 'IS_LOW', 'IS_MEDIUM', 'IS_HIGH', and up to 256 leaves (column labels such as "Date", "PolicyNumber", "Age", "Cost"). Depending on the type of each node, the corresponding binary value is mapped to one of these identifiers and added to the phenotype. The mapping process is also used to ensure all rules are syntactically correct, see (Bentley, 1999).

3.3.2 Rule Evaluation

Every evolved phenotype (or fuzzy rule) is evaluated by using the fuzzy expert system to apply it to the fuzzified training data, resulting in a defuzzified score between 0 and 1 for every fuzzified data item. This list of scores is then assessed by fitness functions which provide separate fitness values for the phenotype, designed to:

i. minimise the number of misclassified items (where a misclassified item is an "unknown" data item with a score > 0.5).
ii. maximise the difference between the average scores for correctly classified "suspicious" items and the average scores for "normal" items (where a correctly classified suspicious item is a data item in the first *Sn* of the training set with score > 0.5).
iii. maximise the sum of scores for "suspicious" items.
iv. penalise the length of any rules that contain more than four identifiers (binary, unary, or leaf nodes).

The first function ensures as few misclassifications as possible. The second forces evolution to distinguish between "suspicious" and "unknown" classes of data, while the third demands that 'suspicious' items are given higher scores than "unknown" ones. The final function ensures that all evolved rules are short - serving the dual purpose of preventing bloat and increasing the readability of the final output.

3.3.3 Rule Generation

Using these four fitness values for each rule, the GP system then employs the SWGR multiobjective optimisation ranking method (Bentley & Wakefield,

1997) to determine how many offspring each pair of rules should have. (Fitnesses are scaled using the effective ranges of each function, multiplied by importance values and aggregated. Rules with higher overall fitnesses are given higher ranking values, and hence have an increased probability of producing offspring.) Child rules are generated using one of two forms of crossover. The first type of crossover emulates the single-point crossover of genetic algorithms by finding two random points in the parent genotypes that resemble each other, and splicing the genotypes at that point. By ensuring that the same type of nodes, in approximately the same places, are crossed over, and that the binary numbers within the nodes are also crossed, an effective exploration of the search space is provided without excessive disruption (Bentley & Wakefield, 1996). The second type of crossover generates child rules by combining two parent rules together using a binary operator (an 'AND' or 'OR'). This more unusual method of generating offspring (applied approximately one time out of every ten instead of the other crossover operator) permits two parents that detect different types of "suspicious" data to be combined into a single, fitter individual. Mutation is also occasionally applied, to modify randomly the binary numbers in each node by a single bit.

The GP system employs population overlapping, where the worst *Pn%* of the population are replaced by the new offspring generated from the best *Pm%*. Typically values of *Pn* = 80 and *Pm* = 40 seem to provide good results. The population size was normally 100 individuals.

3.3.4 Modal Evolution

Each evolutionary run of the GP system (usually only 15 generations) results in a short, readable rule which detects some, but not all, of the "suspicious" data items in the training data set. Such a rule can be considered to define one mode of a multimodal problem. All items that are correctly classified by this rule (recorded in the modal database, see figure 1) are removed and the system automatically restarts, evolving a new rule to classify the remaining items. This process of modal evolution continues until every "suspicious" data item has been described by a rule. However, any rules that misclassify more than a predefined percentage of claims are removed from the final rule set by the system.

3.4 ASSESSMENT OF FINAL RULE SET

Once modal evolution has finished generating a rule set, the complete set of rules (joined into one by disjunction, i.e., 'OR'ed together) is automatically applied to the training data and test data, in turn. Information about the system settings, which claims were correctly and incorrectly classified for each data set, total processing time in seconds, how the data was clustered and the rule set are stored to disk.

3.5 APPLYING RULES TO FUZZY DATA

The path of evolution through the multimodal and multicriteria search space is guided by fitness functions. These functions use the results obtained by the Rule Parser - a fuzzy expert system that takes one or more rules and interprets their meaning when they are applied to each of the previously fuzzified data items in turn.

This system is capable of two different types of fuzzy logic rule interpretation: traditional fuzzy logic, and *membership-preserving* fuzzy logic, an approach designed during this research. Depending on which method of interpretation has been selected by the user, the meaning of the operators within rules and the method of defuzzification is different. Complete details of the fuzzy interpretation methods are provided in (Bentley, 1999).

4 COMMITTEE DECISIONS

As should now be apparent, the evolutionary-fuzzy system has a number of very different elements that can be used at any one time. The choice of clusterer, membership functions, fuzzy interpreter, fitness functions and GA settings can cause varying degrees of success for different input data. What may be a good setup for one data set is not so good for another. In addition, previous work has identified the need for results to be both intelligible and accurate (Bentley, 1999), so multiple results generated by multiple different system setups need to be assessed against multiple criteria.

To achieve this, the system has been extended into a multi-model decision aggregation system. The user can now set up as many as four different versions of the system and have them run in parallel on the same data set, for a user-defined number of times. On completion, the committee decision maker analyses all results written to disk by the different systems, writing the analysis and recommendation of the best evolved rules to disk, see figure 3. The separate evolutionary fuzzy systems (or committee members) have been modified to allow efficient parallel processing (e.g., reading of the data files is performed one at a time on a first-come-first-served basis to avoid excessive "disk thrashing" caused by simultaneous accessing).

4.1 AUTOMATIC DATA ANALYSIS

Three simple forms of analysis are automatically performed by the committee decision maker. First, the rule sets generated by each committee member are examined separately. The most accurate rule set(s) (measured using the number of items in the first class correctly classified) and the most intelligible rule set(s) (where fewer rules = more intelligible) are found. The most accurate *and* intelligible rule set(s) are then chosen for each committee member using decision aggregation.

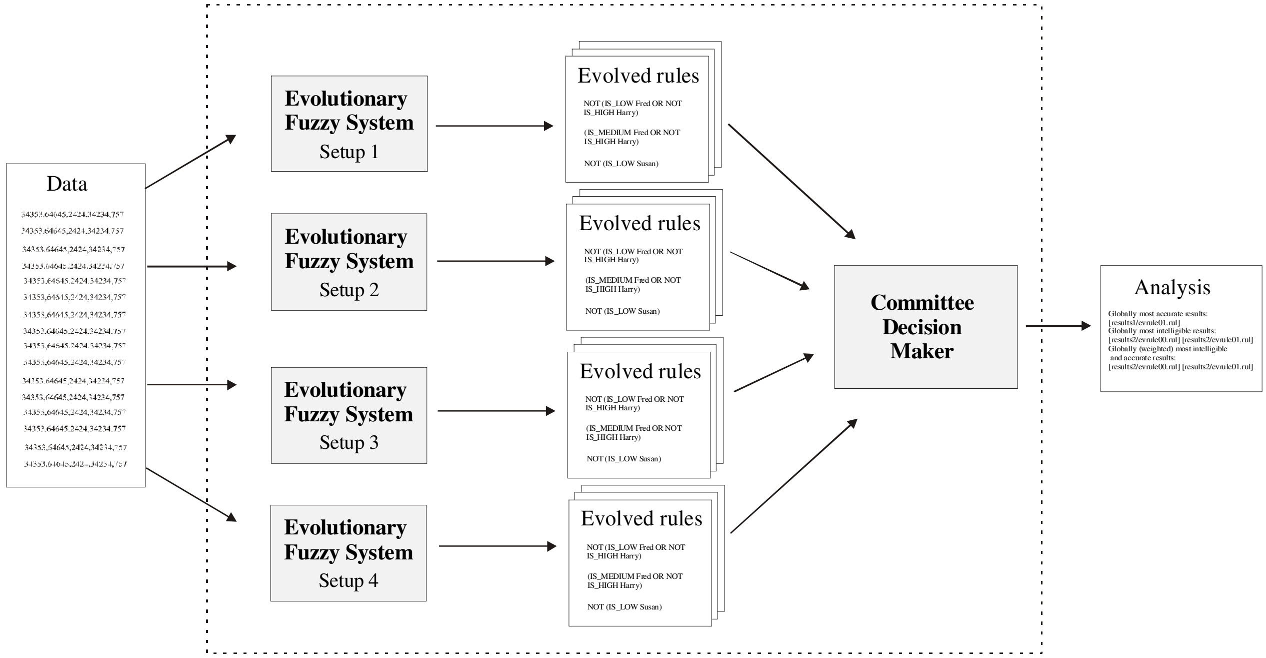

Figure 3 Block diagram of the committee decision system.

In a similar way to (Bunn, 1989), the committee decision maker employs aggregation of weighted normalised values. In other words, each rule set is given a score *s*, where:

$$s = w1\left(\frac{a-\min(a)}{\max(a)-\min(a)}\right) + w2\left[1-\left(\frac{p-\min(p)}{\max(p)-\min(p)}\right)\right]$$

w1 is the importance weighting for accuracy,
w2 is the importance weighting for intelligibility,
a is the accuracy of the current rule set (higher values are better),
p is the intelligibility of the current rule set (lower values are better).

To force the different effective ranges of the multiple criteria to be commensurable (Bentley & Wakefield, 1997), the accuracy and intelligibility values are normalised (and the intelligibility value is inverted) before being weighted. Using information provided by Lloyds TSB, the default weighting values were 0.3 and 1.0 for accuracy and importance, respectively.

Once every rule set has been assigned a score, the set(s) with the highest score for each committee member are reported to the user.

The committee decision maker then performs the same analysis globally, finding the globally most accurate and intelligible rule set(s), then assigning every rule set a score based on globally aggregated, weighted, normalised values. The best overall rule set(s) are then reported to the user.

Finally, a histogram of field occurrences in all evolved rule sets is automatically constructed. As will be shown later, this provides a clear picture of which fields are most important for classification of the data.

5 APPLYING THE SYSTEM TO INSURANCE DATA

5.1 PREPROCESSING THE DATA

As with any real-world problem, classification of real data is often far removed from the clean, perfect world of mathematical theories. Data is usually noisy, inconsistent and sometimes inadequate. Even though intelligent techniques such as GP and fuzzy logic can handle such characteristics better than many approaches, significant data preprocessing will always be required.

The insurance data used for this work was no exception. The data came from numerous sources within the bank, resulting in two somewhat incompatible files. One file contained 98 cases of "suspicious" insurance claim, each with 73 fields (this was assembled from numerous different files provided). The other contained 20,000 cases of "unknown" insurance claims (that might or might not be suspicious), each with 36 fields. The fields comprised items such as "policy number", "claim number", "date of birth", "policy type", etc. However, the two files had very few fields in common. Even after constructing some new fields by processing others in different formats, only 14 common fields in both files could be found.

Once all non-corresponding fields were removed, we were left with two files, one containing 98 claims, each with 14 fields, the other containing 20000 claims, each with the same 14 fields. The data for every pair of fields was then converted into the same format (for example, dates were initially stored in different formats, different codes were used, etc). Missing values in the files were replaced by random values within the range of normal values for each field. (Attempting to classify data with missing values is difficult, so it is simpler to fill the gaps with random values. This has the effect of adding a small level of noise to the data – in this case 1.07% overall. However, the distribution of missing values, and hence noise per field was not even – it varied from 0% to 17%. By keeping a record of the percentage noise per field, the reliability of evolved rules that use the noisiest fields can then be reduced. Note that additional noise in the form of errors within the data was also evident.)

In an attempt to extract more information from the data, and give the classifier a better chance of success, six new fields were created by processing existing fields. For example a new field called 'days before claiming' was constructed by subtracting values in the field 'accident date' from the values in the field 'notified date'.

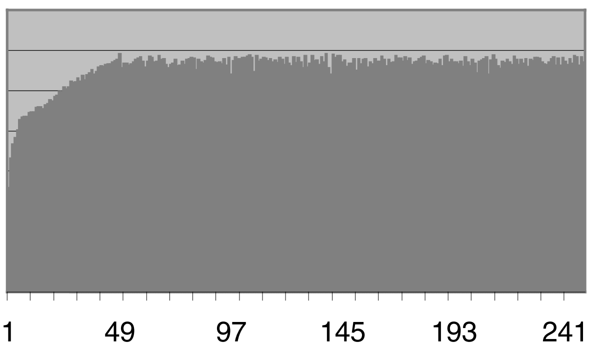

Figure 4 Chart showing the first 250 values, for a field related to the date. Note how the "suspicious" values in the first 49 are much lower on average than the "unknown" values in claims 50 upwards.

A training and test data file was constructed, each containing 49 "suspicious" claims and 10000 "unknown" (alternate claims taken from the original files). A series of experiments were then performed using the evolutionary fuzzy system. The results were suspiciously good – indeed, accuracy was 100%. From these experiments it became clear that inconsistencies in the data were proving considerably more useful as indicators of fraud than anything else. The disparity was mainly caused by the fact that the 98 cases of "suspicious" claim were gathered over a period of some years, whilst the "unknown" data was gathered over a recent period of three months. Any field that varied according to the date was therefore lower, on average, for the "suspicious" fields compared to the "unknown". By plotting charts of each field, it was simple to discover that this adversely effected six of fields, e.g. see fig. 4.

While it is possible that variations on the frequency of fraud may depend on absolute values of dates (e.g. perhaps fraud becomes more likely during a particular month of a year, or following a television programme on 'how to do fraud'), this was seen as unlikely. It was therefore more desirable to attempt to find more generic indicators of fraud, not those dependent on absolute times or specific policy numbers. Consequently, all six fields were deleted (and a seventh which had the same value for all claims was also removed), leaving thirteen fields in each data item. Information was not lost, however. The new fields mentioned earlier contained *relative* date information, so the data contained within five of the deleted fields was still available (with the benefit that the time biases were removed, as differences between fields were used, rather than absolute values).

5.2 EXPERIMENTS

As should be apparent, the task of detecting genuine patterns of fraud using the data provided was not trivial. Indeed, although the data was now in a fit state to be used by a classifier, there still remained the problem of the "unknown" data set. Lloyds TSB suggested that up to 5% of the items in this set might be "suspicious", but which claims and exactly how many was unknown. To tackle this problem, three sets of experiments were performed with the committee decision system. The first experiment assessed the ability of the system to find rules indicative of "suspicious" items, without those patterns describing any "unknown" items. The second experiment assessed how well the system could find "suspicious" rules that also detected up to 5% of the "unknown" items. The third experiment assessed the ability of the system to find rules that detected "suspicious" items and up to 10% of the "unknown" items. (Note that although the system does report which claims in the "unknown" data set were found to be suspicious, these results cannot be provided here.)

Each experiment used four setups of the system:

1. standard fuzzy logic with non-overlapping membership functions
2. standard fuzzy logic with overlapping membership functions
3. membership-preserving fuzzy logic with overlapping membership functions
4. membership-preserving fuzzy logic with smooth membership functions

(Previous work had shown that varying these aspects of the system caused the largest variation in behaviour (Bentley, 1999).)

All four committee members were trained on one file and tested on the other, then trained on the second and tested on the first. This resulted in 24 different rule sets being generated for this problem, each with different levels of intelligibility and accuracy.

5.3 RESULTS

Table 1 and 2 present the results of the experiments. It should be apparent in Table 1 that no committee member managed to find useful rules that detect 0% "suspicious" claims in the "unknown" set – indeed most failed to generate any valid rules at all. When up to 5% or 10% "suspicious" claims are assumed to exist in the "unknown" data set, accuracy rates increase dramatically. As Table 2 explains, committee members [A] and [D] provide the most accurate and intelligible classifications

Table 1 Intelligibility (number of rules) and accuracy (number of correct classifications of “suspicious” items) of rule sets for test and training data. Accuracy rates are listed as n, m where n = number out of 49 correctly classified in class 1, m = number classified out of 10,000 in class 2. Results are given for training on file1, testing on file2 and training on file2, testing on file1.

Estimate of fraud in ’unknown’		**[A] Fuzzy Logic with non-overlappingMFs**			**[B] Fuzzy Logic with overlapping MFs**			**[C] MP-Fuzzy Logic & overlapping MFs**			**[D] MP-Fuzzy Logic with smooth MFs**		
	files:	**rules**	**train**	**test**	**rules**	**train**	**test**	**rules**	**train**	**test**	**rules**	**train**	**test**
No more than 0%	1, 2	3	6, 0	5, 0	failed	0, 0	0, 0	failed	0, 0	0, 0	failed	0, 0	0, 0
	2, 1	2	6, 0	4, 0	failed	0, 0	0, 0	failed	0, 0	0, 0	failed	0, 0	0, 0
No more than 5%	1, 2	5	28, 177	23, 219	4	14, 464	44, 9347	2	3, 418	4, 358	1	30, 236	20, 168
	2, 1	3	29, 318	27, 312	4	16, 304	16, 387	3	3, 165	1, 174	1	24, 340	31, 278
No more than 10%	1, 2	4	35, 940	26, 399	5	12, 853	9, 725	1	4, 740	6, 759	1	30, 344	26, 420
	2, 1	4	32, 889	28, 931	5	21, 628	19, 622	2	11, 558	6, 583	1	24, 335	29, 258

Table 2 Best results as reported by committee decision maker.

Estimate of fraud in ‘unknown’	**Committee decision for accuracy**	**Committee decision for intelligibility**	**Committee decision for weighted intelligibility (1) and accuracy (0.3)**
No more than 0%	[A] 2nd rule set	[A] 2nd rule set	[A] 2nd rule set
No more than 5%	[A] 2nd rule set	[D] 2nd rule set	[D] 2nd rule set
No more than 10%	[A] 1st rule set	[D] 1st rule set	[D] 1st rule set

for all experiments with this data. The best accuracy overall is achieved by [A], finding 61out of 98, or 62% of the “suspicious” claims, whilst suggesting that 1339 out of 20000, or 6.7% of the “unknown” claims are also suspicious. But the most accurate and intelligible rule sets are generated by [D], with most rule sets containing just a single rule. Overall, the best rule set as reported by the committee decision maker is:

(IS_LOW Field8 OR Field3)

which can be translated as:

If either the value for field8 is low or the value for field3 is high, then in 57% of observed cases the claim will be suspicious. This rule suggests that 3.8% of the “unknown” claims are suspicious.

Further analysis can be performed by examining the occurrences of fields in the evolved rules, see table 3. In general, the tally of field occurrences in the rules as shown above indicates that suspicious claims seem to be more likely when:

Fields 1, 5, 7, 9 and 13 are medium or high
Fields 2,3,4 and 6 are high
Field 8 is low or high
Fields 11 and 12 are low or medium

Interestingly, the only field with significant levels of noise – Field10 – is hardly used for classification in the rules. The table also shows that Field3 seems to provide the single best indication of ‘suspiciousness’. Indeed, even used on its own, the rule:

IS_LOW IS_LOW Field3

which in mp-fuzy logic should be translated as:

Table 3 Frequencies of fields in all rule sets and reliability of fields (based on noise caused by filling missing values). Note that NOT IS_X field is expanded to IS_Y Field or IS_Z Field and IS_LOW IS_LOW is translated to IS_HIGH for mp-fuzzy logic.

Field Occurrences	**No more than 0% suspicious in unknown**			**No more than 5% suspicious in unknown**			**No more than 10% suspicious in unknown**			**Reliabilty (100% - % noise)**
	Low	Medium	High	Low	Medium	High	Low	Medium	High	
Field1	0	2	0	1	0	0	0	2	4	100
Field2	0	0	0	0	2	1	0	0	2	99.95
Field3	0	0	0	0	4	0	0	0	6	100
Field4	0	0	0	0	2	3	0	1	2	100
Field5	0	0	0	0	1	0	0	1	1	100
Field6	0	0	0	0	2	0	0	0	1	100
Field7	0	0	0	2	1	0	0	1	0	99.95
Field8	0	0	2	0	3	0	2	1	1	99.93
Field9	0	0	0	0	4	1	0	1	4	100
Field10	0	0	0	0	1	0	0	0	0	83.12
Field11	0	0	2	1	1	0	2	3	1	99.98
Field12	0	0	4	2	1	0	3	1	1	100
Field13	0	0	0	1	5	1	0	1	3	100

ISVERYHIGH Field3

is capable of detecting 54 out of 98 suspicious claims.

6 CONCLUSIONS

This paper has described the use of genetic programming to evolve fuzzy rules within a parallel committee decision system for the detection of suspicious home insurance claims. Attention was paid to data preprocessing, describing some of the typical problems associated with real-world data in order to show just how hard this kind of classification becomes. Nevertheless, despite having only 49 suspicious items in the first class to train the system, and an unknown number of suspicious items in the 10000-item second class, performance of the system was good. Given the quality and quantity of the data, accuracy rates of over 60% must surely be regarded as impressive. Indeed it seems very likely that better accuracy would only result in overfitting the meagre training data. In addition, intelligibility rates were excellent with many rule sets comprising a single, understandable rule.

This work shows the benefit of committee decision making. Each of the four different committee members (different setups of the evolutionary fuzzy system) provided different rates of accuracy and intelligibility. The committee decision maker was able to analyse all results and pick the best rule set.

The evolved rules and the table of field frequencies in rules have provided important and interesting information about the nature of fraud in home insurance claims. Sadly the names of the fields and the true meanings of the rules cannot be reported in this article, but Lloyds TSB have stated that "the results were sensible as confirmed by previous analysis, and support the potential for even more useful results with improved data."

As Sherlock Holmes said in *The Study of Scarlet*, "There is a strong family resemblance about misdeeds, and if you have all the details of a thousand at your finger ends, it is odd if you can't unravel the thousand and first."

Acknowledgments

Thanks to Hugh Mallinson for his implementation of the clustering algorithms and some of the fitness functions used in the system. Insurance data has been kindly provided by Lloyds TSB. This project is being performed in collaboration with Searchspace Ltd and Lloyds TSB, and is funded by the EPSRC, ref: GR/L3/708.

References

Bentley, P. J. (1999). Evolving Fuzzy Detectives: An Investigation into the Evolution of Fuzzy Rules. A late-breaking paper in Genetic and Evolutionary Computation Conference (GECCO '99), July 14-17, 1999, Orlando, Florida USA, pp. 38-47.

Bentley, P. J. & Wakefield, J. P. (1996). Hierarchical Crossover in Genetic Algorithms. In Proceedings of the 1st On-line Workshop on Soft Computing (WSC1), (pp. 37-42), Nagoya University, Japan.

Bentley, P. J. & Wakefield, J. P. (1997). Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms. Chawdhry, P.K.,Roy, R., & Pant, R.K. (eds) Soft Computing in Engineering Design and Manufacturing. Springer Verlag London Limited, Part 5, 231-240.

Bezdek, J. C. and Pal, S. K. (Ed.s) (1992). *Fuzzy Models for Pattern Recognition.* IEEE Press, New York.

Bunn, D. (1989), Forecasting with more than one model. *Journal of Forecasting* v8, pp. 161-166.

Dasarathy, B.V. (1980) Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 1, 67-71.

Hartigan, J. A (1975). *Clustering algorithms.* Wiley, NY.

Kosco, B. (1994). *Fuzzy Thinking, the new science of fuzzy logic*. Flamingo. Harper Collins Pub., London.

Koza, J. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press.

Koza, J. et al. (1998). *Genetic Programming '98: Proceedings of the Third Annual Genetic Programming Conference*. Morgan Kaufman Pub., CA.

Mallinson, H. and Bentley, P.J. (1999). Evolving Fuzzy Rules for Pattern Classification. In Proc. of the Int. Conf. on Computational Intelligence for Modelling, Control and Automation - CIMCA'99.

Marmelstein, R. E. and Lamont, G. B. (1998) Evolving Compact Decision Rule Sets. In Koza, John R. (ed.). *Late Breaking Papers at the Genetic Programming 1997 Conference*, Omni Press, pp. 144-150.

Mc. Neill, D. and Freiberger, P. (1993). *Fuzzy Logic*. Touch Stone Pub.

Pedrycz, W. (Ed.) (1997) *Fuzzy Evolutionary Computation*. Kluwer Academic Publishers, MA.

Wolberg, W. H., and Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In *Proceedings of the National Academy of Sciences*, 87,9193--9196.

Yu, T. and Bentley, P. (1998). Methods to Evolve Legal Phenotypes. In Proceedings of the Fifth Int. Conf. on Parallel Problem Solving From Nature. Amsterdam, Sept 27-30, 1998, pp. 280-282.

Zadeh, L. A. (1965). Fuzzy Sets. *Journal of Information and Control*, v8, 338-353.

Zhang, J. (1992). Selecting typical instances in instance-based learning. In *Proceedings of the Ninth International Machine Learning Conference*, pp. 470--479. Aberdeen, Scotland: Morgan Kaufmann.