Soft Comput (2013) 17:2185–2201
DOI 10.1007/s00500-013-1102-y

METHODOLOGIES AND APPLICATION

A genetic type-2 fuzzy logic based system for the generation of summarised linguistic predictive models for financial applications

Dario Bernardo · Hani Hagras · Edward Tsang

Published online: 13 August 2013

Abstract The global economic meltdown of the late 2000s exposed many organisations around the world, this drove the need to build robust frameworks for predicting and assessing risks in financial applications. Such predictive frameworks helped organisations to increase the quality and quantity of their transactions hence increasing the revenues and reducing the risks. Many organisations around the World still use statistical regression techniques which are well established for many problems such as fraud detection or risk analysis. However, recent years have seen the application of computational intelligence techniques to develop predictive models for financial applications. Some of the computational intelligence techniques like neural networks provide good predictive models, nevertheless they are considered as black box models which do not provide an easy to understand reasoning about a given decision or even a summary of the generated model. However, in the current economic situation, transparency became an important factor where there is a need to fully understand and analyze a given financial model. In this paper, we will present a Genetic Type-2 Fuzzy Logic System (FLS) for the modeling and prediction of financial applications. The proposed system is capable of generating summarized models from a specified number of linguistic rules, which enables the user to understand the generated financial model. The system is able to use this summarized model for prediction within financial applications. We have performed several evaluations in two distinctive financialdomains, one for the prediction of good/bad customers in a financial real-world lending application and the other domain was in the prediction of arbitrage opportunities in the stock markets. The proposed Genetic type-2 FLS has outperformed white box models like the Evolving Decision Rule procedure (which is a white based on Genetic Programming and decision trees) and gave a comparable performance to black box models like neural networks while the proposed genetic type-2 FLS provided a white box model which is easy to understand and analyse by the lay user.

1 Introduction

The global economic meltdown of the late 2000s exposed many organisations around the world where every financial indicator was on a downward trend. As companies begin their slow recovery, they are increasingly looking for ways to reduce the risk associated with their business. This led to the realisation of a number of advanced products and techniques that aim to help organisations to reduce risk or take better decisions. As a matter of the fact, when the quality of the possible investments decreases or the risk associated with investments increases, being able to fully understand the faced risks and reduce them while avoiding bad investments can make the difference between dying, surviving or expanding.

Nowadays organisations have access to a quantity of data and information that was not available 20 years ago. Looking at the current trend, in a few years the amount of information will even be more. In addition, nowadays everything is online and in seconds it is possible to have huge amounts of information. Besides, it is also becoming much easier to store and maintain large amounts of data. Hence, different financial organisations are moving towards generating mod-

Communicated by G. Acampora.

D. Bernardo · H. Hagras (✉) · E. Tsang
The Computational Intelligence Centre, School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK
e-mail: hani@essex.ac.uk

els based on data where these models are trying to predict the future by looking at the past.

Many organisations around the World still use statistical regression models which capture only information that can be refined into mathematical models to generate two outputs (0/1 or Good/Bad). Statistical regression analysis include many techniques (linear, multiple, logistic) for modelling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables. One of the simplest and most popular modelling methods is linear regression. Linear regression is the most used technique in finance. For example the "capital asset pricing model" uses linear regression (Cohen et al. 2003) as well as the concept of "Beta" for analyzing and quantifying the systematic risk of an investment (Levinson 2006). Linear regression is also often used in financial time series modelling (Cohen et al. 2003). In addition, linear regression is also an important empirical tool in economics, for example, it is used to predict consumption spending (Deaton 1992) fixed investment spending, inventory investment, purchases of a country's exports (Krugman and Obstfeld 1988) spending on imports (Krugman and Obstfeld 1988) the demand to hold liquid assets (Laidler 1993) labour demand (Ehrenberg and Smith 2008) and labour supply (Ehrenberg and Smith 2008). Logistic regression is a variant of nonlinear regression that is appropriate when the target (dependent) variable has only two possible values (e.g., live/die, buy/don't-buy, infected/not-infected). However regression techniques in general are often considered black box models which cannot be easily understood and analyzed by the normal user.

Some advanced machine learning and artificial intelligence techniques have been applied in the financial domain. For example Support Vector Machines (SVMs) have been applied in Kim (2003) to forecast financial time series and in Kim and Sohn (2010) to effectively manage governmental funds to small and medium enterprises by identifying those likely to default. Another machine learning technique is Neural Networks (NNs) which have been applied successfully in big number of financial applications such as Giacomini (2003), Lawrence (1997), Kwong (2001). However, the drawback of such advanced machine learning techniques is that although they can give good prediction accuracies, they provide black box models which are very difficult to understand and analyse by a financial analyst where it is now becoming a common requirement to have an explanation or the reasoning behind a given financial decision.

There are actually a number of reasons why models that we can understand are important; the main reason is trust. No matter how sophisticated our economy has become all transactions still comes down to trust, we have to trust the person that we are trading with. This requires transparency so that we can see what the other party is doing. This need of transparency is reflected in legislations that force financial institutions to disclose the reasoning behind their financial decisions and models.

There exist various white box transparent models, one of these models is decision trees Decision trees are well suited to modeling target variables with binary values, but—unlike logistic regression—they also can model variables with more than two discrete values, and they handle variable interactions. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. Decision trees can provide an explanation for the output class chosen. Various works have been reported using decision trees in financial applications such as Garcia-Almanza (2008); Garcia-Almanza and Tsang (2008).

Fuzzy Logic Systems (FLSs) provide white box models which could be easily analyzed and understood by the layman user. However FLSs suffer from the curse of dimensionality problem which causes the FLS-based system to generate a big number of rules in order to give good model accuracy. Most recently type-2 FLSs that are capable of handling high uncertainty levels have been employed for the generation of classification models (Sanz et al. 2010, 2011). However, the existing type-2 fuzzy classification systems are not suited for the financial domain where such type-2 FLSs generate big rule bases; besides, they make the assumption that all the possible rules are represented in the existing models which is impossible for systems with big number of inputs where the generated model will only cover a small subset of the search space. Furthermore, FLSs have a high number of parameter to tune, which sometimes require some time to be chosen in the optimal way.

In this paper, we will present a genetic type-2 FLS for the modeling and prediction in financial applications. The proposed system avoids the drawbacks of the existing type-2 fuzzy classification systems in that the proposed system is able to carry prediction based on a relativity small pre-specified rule base size even if the incoming data vector does not match any rules in the FLS rule base. The proposed type-2 FLS aims to increase the understandability of the generated model by achieving the best performance possible with a limited and summarized number of rules in order to achieve simplicity and comprehensibility for the user. We have carried various evaluations where we are going to present in this paper results from two distinctive financial domains one for prediction of good/bad customers in a financial real-world lending application and the other domain was in the prediction of arbitrage opportunities in the stock markets. The proposed system was able to use the generated summarized models for the prediction within financial applications. The proposed Genetic type-2 FLS has outperformed white box models like the Evolving Decision Rule (EDR) procedure (which is a white based on Genetic Programming (GP) Garcia-Almanza and Tsang 2008 and decision trees) and gave a comparable performance to white box models like neural networks while the proposed genetic type-2 FLS provided a white box model which is easy to understand and analyse by the lay user.

In Sect. 2, we will present a brief overview on type-2 FLSs. Section 3 will present an overview on the fuzzy classification systems. Section 4 will present the proposed genetic type-2 fuzzy based modeling and prediction system for financial applications. Section 5 will present the experiments and the achieved results. Finally Sect. 6 will present the conclusions and future work.

2 Brief overview type-2 fuzzy logic systems

In the recent years type-2 FLSs have grown in popularity due to their ability to handle high levels of uncertainties. Type-2 FLSs employ type-2 fuzzy sets as shown in Fig. 1 where a type-2 fuzzy set is characterized by a fuzzy Membership Function (MF), i.e. the membership value (or membership grade) for each element of this set is a fuzzy set in [0, 1], unlike a type-1 fuzzy set where the membership grade is a crisp number in [0, 1] (Hagras 2004).

The membership functions of type-2 fuzzy sets are three dimensional and include a Footprint Of Uncertainty (FOU) (shaded in grey in Fig. 1), it is the new third-dimension of type-2 fuzzy sets and the Footprint Of Uncertainty (FOU) that provide additional degrees of freedom that make it possible to directly model and handle uncertainties (Hagras 2004; Mendel 2001). The interval type-2 FLSs use interval type-2 fuzzy sets (such as the type-2 fuzzy set shown in Fig. 1 to represent the inputs and/or outputs of the FLS). In the interval type-2 fuzzy sets all the third dimension values are equal to one. The use of interval type-2 FLS helps to simplify the computation (as opposed to the general type-2 FLS).

The proposed system in the paper is a type-2 fuzzy classification system and hence it does not follow the structure of the type-2 FLSs reported in Hagras (2004), and Mendel (2001) where the classification system process is summarized in the following section.

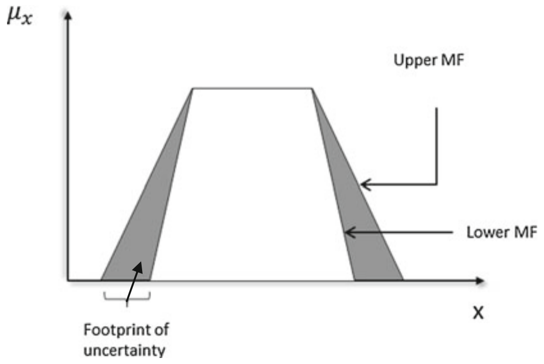


Fig. 1 A type 2 fuzzy set

An interval type-2 fuzzy set denoted $\tilde{A}$ is written as follows:

$$\mu_{\tilde{A}}(x) = \int_{x\in X} \int_{u\in\left[\bar{\mu}_{\tilde{A}}(x),\underline{\mu}_{\tilde{A}}(x)\right]} 1/u \tag{1}$$

$\bar{\mu}_{\tilde{A}}(x)$, $\underline{\mu}_{\tilde{A}}(x)$, represent the upper and lower membership functions respectively of the interval type-2 fuzzy set $\tilde{A}$. The upper membership function is associated with the upper bound of the footprint of uncertainty $FOU(\tilde{A})$ of a type-2 membership function. The lower membership function is associated with the lower bound of $FOU(\tilde{A})$ (Hagras 2004).

3 Brief overview on fuzzy classification systems

In fuzzy logic classification systems, for a given c-class pattern classification problem with n attributes (or features), a given rule in the FLS rule base could be written as follows:

$$\begin{aligned}&\text{Rule } R^j : \text{ If } x_1 \text{ is } A_1^j \text{ and} \ldots \text{and } x_n \text{ is } A_n^j \text{ then Class } C_j\\ &\quad\text{with } CF_j,\ j = 1, 2, \ldots, N\end{aligned} \tag{2}$$

where $x_1 \ldots, x_n$ represent the n-dimensional pattern vector, A_i^j is the fuzzy set representing the linguistic label for the antecedent pattern i, C_j is a consequent class (which could be one of the possible c classes), N is the number of fuzzy IF-Then rules in the FLS rule base. CF_j is a certainty grade of rule j (i.e., rule weight). Assuming each input pattern is represented by K fuzzy sets and given that we have n input patterns, the possible number of rules that will cover the whole search space is K_n. In the arbitrage application presented in this paper, we have seven inputs where each input is represented by five fuzzy sets; hence the needed number of rules to cover the whole search space for this given application is $5^7 = 78{,}125$ rules. Each rule represents all the available input patterns where each pattern is represented by one of the available fuzzy sets and “don’t care” conditions are not considered by any input feature. For our future work, we will introduce “don’t care” conditions as this will help to increase the interpretability of the rule as explained in Ishibuchi et al. (1999). In our given applications (which applies to the vast majority of financial applications), we do not have enough data to generate this huge number of rules. Hence, there will be various cases where the incoming input vector will not fire any rule in the FLS rule base.

In the design of a fuzzy rule-based system, there exist two conflicting objectives: error minimization and comprehensibility maximization. The trade-off between these two objectives has been discussed in some studies (Casillas et al.

2003a,b). Several type-1 fuzzy classification systems have been reported in the literature such as Ishibuchi (2001a,b), Ishibuchi and Yamamoto (2004, 2005, 2006) Shigeo (1995), Ahmad and Jahormi (2007), Wang (2003), and Mansoori et al. (2006). However, in the vast majority of these papers, the data was quite easy to partition, and if an input pattern does not match any of the decision areas previously labelled, the input is discharged. In financial applications this cannot be done where if a new pattern that has never been seen before is proposed, a decision needs to be made anyway, and unfortunately discharging a given pattern a priory cannot be the solution. A technique to resolve this problem was proposed in Garcia-Almanza and Tsang (2006, 2008), and, this technique keeps in a rule repository all the rules for the minority class in unbalanced data sets. All the inputs that do not match any rule in the repository are considered belonging to the majority class. This technique can work in unbalanced data set but might not work in all cases.

Most recently type-2 FLSs that are capable of handling high uncertainty levels have been employed for the generation of classification models (Sanz et al. 2010, 2011). However, the existing type-2 fuzzy classification systems are not suited for the financial domain where such type-2 FLSs generate big rule bases and make the assumption that all the possible rules are represented in the existing models which is impossible for the problems with big number of input variables where the generated model will only cover a small subset of the search space. In this paper, we will present a type-2 FLS for the modelling and prediction of financial applications. The proposed system avoids the drawbacks of the existing type-2 fuzzy classification systems where the proposed system is able to carry prediction based on a pre-specified rule base size even if the incoming data vector does not match any rules in the FLS rule base.

4 The proposed genetic type-2 fuzzy modelling and prediction system for financial applications

In fuzzy logic systems, the choice of the appropriate parameters of the fuzzy sets poses a major challenge to the design of a FLS. By simply changing the fuzzy sets parameters, it is possible to change the behaviour of a fuzzy logic system, for example in the field of managing risk in financial systems it is possible to build *riskier* or *risk-averse* fuzzy systems by changing the parameters of the fuzzy sets to make the FLS passing more or less customers. It is extremely difficult though to find the optimal configuration using a simple manual or heuristic approach because of the number of the variables to be optimised and the interaction of these variables. In our work Genetic Algorithms (GAs) were used to tune the parameters of the type-2 fuzzy sets of the FLS.

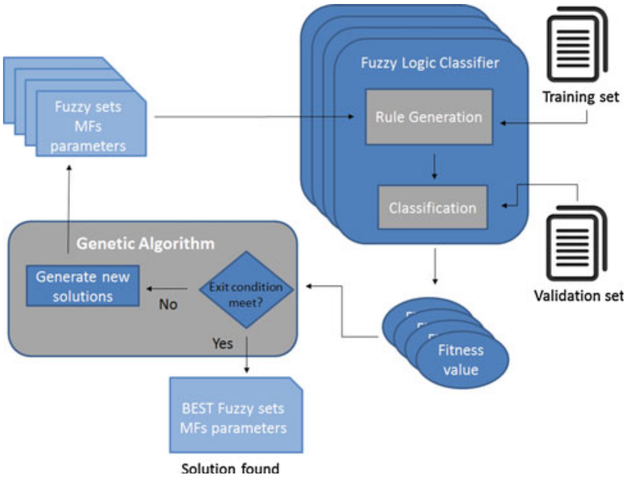


Fig. 2 An overview of the proposed genetic type-2 fuzzy logic system

The GA uses a population where each chromosome describe a fuzzy set space, in other words the size and the position of each membership function for each input. The GA starts by producing randomly an initial population, and then it evolves at each generation the previous population. In the GA each instance of the FLS is created by using each individual of the population and each instance generates a different fitness value. Using the fitness value the best individuals are selected and operators of crossover and mutation are applied to produce the new population for the next iteration. As shown in Fig. 2, the steps followed by the proposed genetic type-2 fuzzy system can be summarised as:

1. Initialize randomly the first generation.
2. Build a rule-base for each parameter configuration of the type-2 fuzzy sets as provided by a given chromosome. As the matter of the fact each chromosome describes the fuzzy membership functions configuration and this in conjunction with the training data is used to build the rule-base (the rule base generation process is discussed in Sect. 4.2).
3. Evaluate the classification ability of the generated type-2 FLS and produce the fitness value for each individual.
4. If an individual reach the desired fitness value or the max number of iteration are reached the algorithm terminates.
5. The GA uses the population and their fitness values to evolve and produce a new population of type-2 fuzzy sets.
6. Go to step 2.

4.1 The GA operation

4.1.1 The GA fitness function

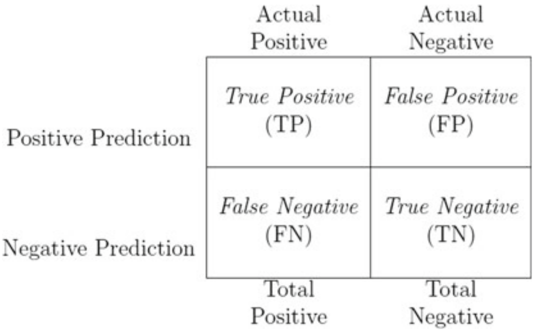
The GA tries to find the best membership function configuration to optimise the fitness function. Our work has been focused on classification problems, where the aim is to identify the correct class for a given input. In order to evaluate the performance we will use the Receiver Operating Characteristic (ROC) curve (Swets 1996). In order to explain how the ROC curve works we have first to briefly introduce the measures computed in a confusion matrix. A confusion matrix displays the data about actual and predicted classifications done by a classifier (Kohavi and Provost 1998). This information is used in supervised learning to determine the performance of classifiers. Given an instance and two classes (positive and negative) there are four possible results: The instance is positive and it is classified as positive (True Positive (TP)). The instance is negative and it is counted as positive (False Positive (FP)). The instance is positive and it is classified as negative (False Negative (FN)). The instance is negative and it is predicted as negative (True Negative (TN)). Figure 3 summarise the confusion matrix for a two class problem.

Hence True Positive (TP) is number of correct predictions in positive cases; False Positive (FP) is the number of incorrect predictions that were classified as positive when the instance is negative. False Negative (FN) is the number of incorrect predictions that were classified as negative when the instance is positive while True Negative (TN) is the number of correct negative predictions.

The ROC curve explains the performance of a classifier by plotting two measures.

- *Recall* which is also called sensitivity or true positive rate which is defined as the proportion of positive cases that were correctly identified (Kohavi and Provost 1998), it is determined by the formula:

$$Recall_{positive} = \frac{TP}{TP + FN} \tag{3}$$



	Actual Positive	Actual Negative
Positive Prediction	*True Positive* (TP)	*False Positive* (FP)
Negative Prediction	*False Negative* (FN)	*True Negative* (TN)
	Total Positive	Total Negative

Fig. 3 Confusion matrix for a two class problem

Recall is calculated on the positive class only (Swets 1996), though it is possible to extend the Eq. (3) on the negative class as well as shown in Eq. (4) below.

$$Recall_{negative} = \frac{TN}{TN + FP} \tag{4}$$

- *False positive rate* is the proportion of negative cases that were wrongly predicted as positive. It is determined by the formula:

$$False\ Positive\ Rate_{positive} = \frac{FP}{FP + TN} \tag{5}$$

False positive rate is by definition calculated on the false positive value of the confusion matrix (Swets 1996). However, in the same way the $Recall_{positive}$ was extended to $Recall_{Negative}$ by calculating it for the negative class, it is possible to extend the False Positive Rate by considering its symmetric version on the negative class, as shown in Eq. (6) below, this measure is also known as False Negative Rate.

$$\begin{aligned} False\ Positive\ Rate_{negative} &= False\ Negative\ Rate \\ &= \frac{FN}{FN + TP} \end{aligned} \tag{6}$$

The point is that on a two class problem it is possible to calculate the *recall* on both classes, so that there will be $recall_{positive}$ and $recall_{negative}$, as well as all other measures. It is interesting to note that it is possible to calculate the *false positive rate* for both classes as well using the following formulas:

$$False\ Positive\ Rate_{positive} = 1 - recall_{negative} \tag{7}$$

$$False\ Positive\ Rate_{negative} = 1 - recall_{positive} \tag{8}$$

This conclusion is important for us because in this way we can consider only the recall as a measure in the fitness function. As the matter of the fact, in order to produce a classifier that optimizes the curve on a ROC graph, the classifier could simply optimize the average of the *recall* for all classes. Hence in our GA the fitness function will be the average *recall* for all classes. In order though to produce different points on the ROC curve representing more risky or risk-averse classifier we used some weights in order to favour some recall for some classes, and hence to position the classifier in different point of the graph.

$$Fitness\ score = \frac{\sum_{i}^{N} Recall_i * w_i}{N} \tag{9}$$

N is the number of classes for the problem, and w_i is the weight defined as $w = \{w_1, \ldots, w_n\}\ with\ n = [1, N]$.

4.1.2 Employing genetic algorithms to determine the type-2 fuzzy sets parameters

This section explains how a chromosome is translated into the fuzzy sets space, describing the size and position of the membership functions. The GAs implementation of Shakya (2004) was used to encode the chromosomes using real numbers. In order to use the implementation we needed to provide the algorithm with the following parameters:

- Solution length: this is the length of a single chromosome; in our implementation this was the number of parameters describing the fuzzy set space that need to be tuned (Kassem 2012).
- Min/Max Range: this is the minimum and maximum number that can be generated in a gene (Kassem 2012).
- Fitness Function: this is the objective function that the GA tries to optimize. This function should take as an input a chromosome (a possible solution) and return fitness score (Kassem 2012).
- Population Size: the number of individuals within a population (Kassem 2012).
- Crossover Rate: every time a pair of parents are chosen from the population produced from the selection process, a random number is generated, if this number is less than the crossover rate then crossover is performed on the parents, otherwise the parents are copied without alteration as the offspring (Kassem 2012).
- Mutation Rate: for every gene within a chromosome a random number is generated, if that number is less than the mutation rate then mutation is performed on that gene, otherwise the gene is left unaltered (Kassem 2012).
- Maximum Generation: this is the maximum number of generations that if reached by the algorithm then termination is forced (Kassem 2012).
- Elite Solutions: this is the number of elite solutions that are copied from one generation to another (Kassem 2012).

Once the above parameter have been chosen for the GA, the algorithm starts by creating an initial generation of individuals randomly. An individual or *gene* is a membership function parameter for the FLS.

Each input for the FLS is represented by five type-2 fuzzy sets, which need 17 parameters to be represented as shown in Fig. 4. Each of the 17 parameters represented in the chromosome does not represent the absolute coordinate in the universe of discourse, but the relative distance from the previous parameter. The translation process of a chromosome into a fuzzy set space will be explained in more detail later. To fully build the fuzzy sets needed by the system, the total number of parameters (genes) to be optimised can be found as follows:

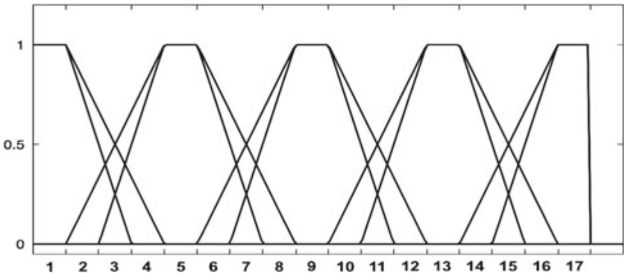


Fig. 4 The number of type-2 fuzzy sets 17 parameters to be tuned for each input

Table 1 The GA parameters

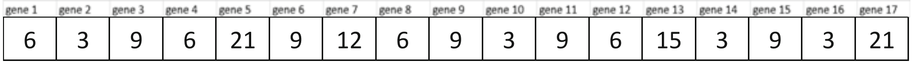
Population size	400
Solution length	119
Elite size	10
Crossover probability	0.8
Crossover type	One point
Mutation probability	0.04
Mutation type	One bit
Selection type	Tournament

$$\mathit{Number\ of\ parameters} = 17 \times F \qquad (10)$$

where F is the number of inputs (or features). For a dataset with 7 input features the total number of parameter to tune will be 119 parameters thus creating a chromosome composed of 119 genes. The fuzzy partitions derived by the chromosome are specified under the constraint that the upper membership function of a given fuzzy set starts at the same point as the right hand vertex of the previous membership function. For this reason, upper membership functions always intersect at the membership value of 0.5. In addition, the sum of the upper membership values is equal to 1.

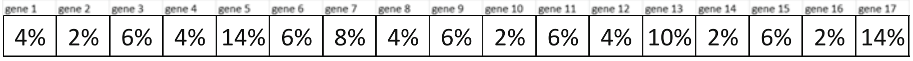
The used GA parameters are listed in Table 1.

If we consider a single input, in order to shape the interval type-2 fuzzy sets for an input we need 17 parameters (as shown in Fig. 4). Each gene contains a percentage representing the distance between parameter i and $i-1$. Let's take for example the segment of the chromosome shown in Fig. 5. This segment contains all parameters needed to build a fuzzy set space for one input. The sum of all genes within this segment is 150, so $gene_5$ (21) is equivalent to (21/150) which is 14 %, this means that the distance between V_4 and V_5 is 14 % of the total universe of discourse. Let's take another example; $gene_1$ (6), this gene would be equivalent to 4 %. This means that the distance between V_1 and the starting point of the fuzzy set would be 4 % of the total universe of discourse, considering that in this example the universe of discourse start from zero and ends at 50 as shown in Fig. 8, the core of the first membership function ends in a decimal value of 2. If the starting point of the universe of discourse would have



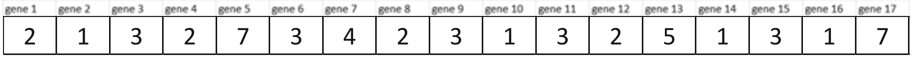
gene 1	gene 2	gene 3	gene 4	gene 5	gene 6	gene 7	gene 8	gene 9	gene 10	gene 11	gene 12	gene 13	gene 14	gene 15	gene 16	gene 17
6	3	9	6	21	9	12	6	9	3	9	6	15	3	9	3	21

Fig. 5 A segment of a chromosome



gene 1	gene 2	gene 3	gene 4	gene 5	gene 6	gene 7	gene 8	gene 9	gene 10	gene 11	gene 12	gene 13	gene 14	gene 15	gene 16	gene 17
4%	2%	6%	4%	14%	6%	8%	4%	6%	2%	6%	4%	10%	2%	6%	2%	14%

Fig. 6 Percentages derived from the chromosome



gene 1	gene 2	gene 3	gene 4	gene 5	gene 6	gene 7	gene 8	gene 9	gene 10	gene 11	gene 12	gene 13	gene 14	gene 15	gene 16	gene 17
2	1	3	2	7	3	4	2	3	1	3	2	5	1	3	1	7

Fig. 7 Distances derived from the percentages

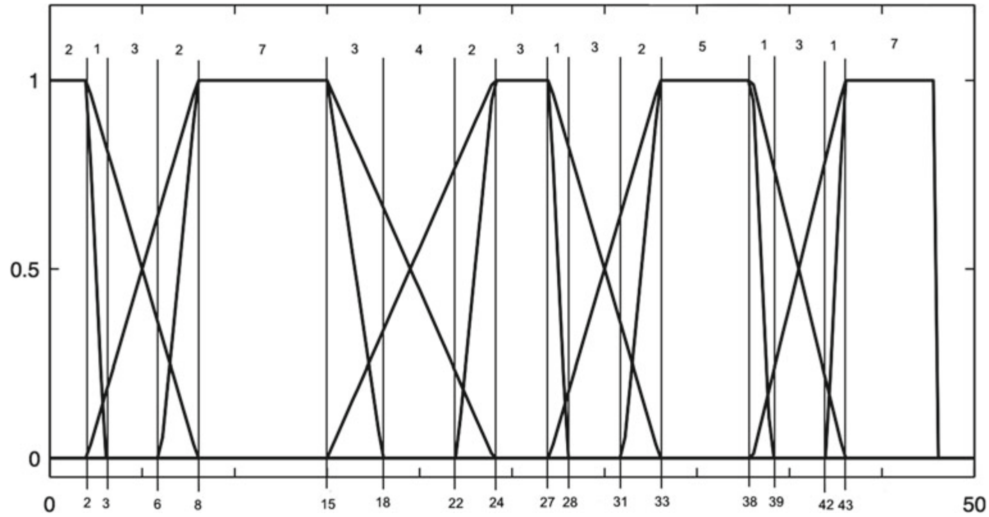


Fig. 8 Final values of the derived membership functions

been S, then the core would have been ending in $S+2$. Figure 6 shows the equivalent percentages for each of the genes shown in Fig. 5. After these percentages are derived, they are applied to the universe of discourse in order to determine the distance for each part of the membership functions. As mentioned, in the example considered in Fig. 8, the universe of discourse start in zero and end in 50, the equivalent distances for the membership functions can be shown in Fig. 7. So if we take $gene_2(2\,\%)$, so 2 % of 50 is 1, which is the second component in Fig. 7. As we have mentioned earlier these components are the distances between the parameters, hence the distance between the second and first parameter is 1 and the decimal value is $2+1=3$ from the beginning of the type-2 fuzzy sets universe of discourse. The final values of the parameters and the distances are shown in Fig. 8.

In order to build a fuzzy set space for 11 variables, 11 different chromosome segments will be selected and used to build the fuzzy set space, needing a chromosome of size $11 * 17 = 187$ genes.

4.2 Rule generation in the proposed genetic type-2 FLS

The previous section showed how to employ genetic algorithms to learn the parameters of the type-2 fuzzy sets. This subsection will show how the rules of the type-2 FLS are modelled taking as an input a dataset and the fuzzy sets whose parameters were optimised by the GA. This is called modelling phase. In the modelling phase the rule base of the type-2 fuzzy classification system is constructed from the existing training dataset. Once the model has been built the FLS can be used to predict new inputs. This is called prediction phase. In the prediction phase, the generated rule base is used to predict the incoming input vectors. Figure 9 shows an overview on the modelling and prediction phases.

4.2.1 The modeling phase

The modeling phase operates according to the following steps (as shown in Fig. 9):

Step 1: Raw rule extraction: For a fixed input–output pair $(x^{(t)}, C^{(t)})$ in the dataset, $t = 1, \ldots T$ (T is the total number of data training instances available for the modeling phase) compute the upper and lower membership values $\bar{\mu}_{A_s^q}$, $\underline{\mu}_{A_s^q}$ for each antecedent fuzzy set $q = 1, \ldots K$ (K is the total number of fuzzy sets representing the input pattern s where $s = 1 \ldots n$). Generate all rules combining the matched fuzzy sets A_s^q (i.e. either $\bar{\mu}_{A_s^q} > 0$ or $\underline{\mu}_{A_s^q} > 0$) for all $s = 1 \ldots n$. Thus the rules generated by $(x^{(t)}, C^{(t)})$ will have different antecedents and the same consequent class $C^{(t)}$ Thus each of the extracted raw rules by $(x^{(t)}, C^{(t)})$ could be written as follows:

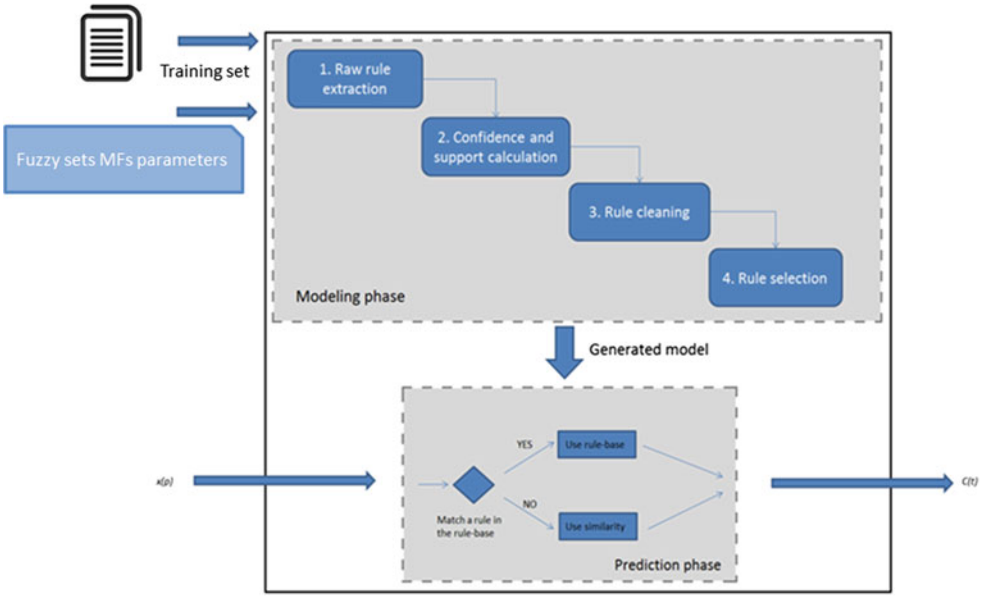


Fig. 9 An overview on the modelling and prediction phases

$$R^j : \textit{If } x_1 \textit{ is } \tilde{\boldsymbol{A}}_{\boldsymbol{1}}^{\boldsymbol{qjt}} \textit{ and } \ldots \textit{ and } x_n \textit{ is } \tilde{\boldsymbol{A}}_{\boldsymbol{n}}^{\boldsymbol{qjt}} \textit{ then Class } C_t, \quad t = 1, 2, \ldots, T \tag{11}$$

For each generated rule, we calculate the firing strength F^t. This firing strength measures the strength of the points $x^{(t)}$ belonging to the fuzzy region covered by the rule. F^t is defined in terms of the lower and upper bounds of the firing strength $\underline{f}^{(t)}$, $\overline{f}^{(t)}$ of this rule which are calculated as follows:

$$\overline{f^{jt}}(x^{(t)}) = \overline{\mu_{A_1^{qjt}}}(x_1) * \cdots * \overline{\mu_{A_n^{qjt}}}(x_n) \tag{12}$$

$$\underline{f^{jt}}(x^{(t)}) = \underline{\mu_{A_n^{qjt}}}(x_1) * \cdots * \underline{\mu_{A_n^{qjt}}}(x_n) \tag{13}$$

The $*$ denotes the minimum or product t-norm. Step 1 is repeated for all the t data points from 1 to T to obtain generated rules in the form of Eq. (11).

The financial data is usually highly imbalanced (for example in a lending application it is expected that the majority of people will be good customers and a minority being bad customers and usually the interesting class is the minority class). Hence, we will present a new approach called "weighted scaled dominance" which is an extension of our previous work "scaled dominance" and the "weighted confidence" work introduced by Ishibuchi and Yamamoto (2005). This method tries to handle imbalanced data by trying to give minority classes a fair chance when competing with the majority class. In order to compute the scaled dominance for a given rule having a consequent Class C_j, we divide the firing strength of this rule by the summation of the firing strengths of all the rules which had C_j as the consequent class. This allows handling the imbalance of data towards a given class. We scale the firing strength by scaling the upper and lower bounds of the firing strengths as follows

$$\overline{fs^{jt}} = \frac{\overline{f^{jt}}}{\sum_{j\in Classj} \overline{f^{j}}} \tag{14}$$

$$\underline{fs^{jt}} = \frac{\underline{f^{jt}}}{\sum_{j\in Classj} \underline{f^{jt}}} \tag{15}$$

Step 2: Scaled support and scaled confidence calculation: Many of the generated rules will share the same antecedents but different consequents. To resolve this conflict, we will calculate the scaled confidence and scaled support which are calculated by grouping the rules that have the same antecedents and conflicting classes. For given m rules having the same antecedents and conflicting classes. The scaled confidence $(\tilde{A}_q \Rightarrow C_q)$ (defined by its upper bound $\bar{c}$ and lower bound $\underline{c}$, it is scaled as it involves the scaled firing strengths mentioned in the step above) that class C_q is the consequent class for the antecedents $\tilde{A}_q$ (where there are m conflicting rules with the same antecedents and conflicting consequents) could be written as follows:

$$\bar{c}(\tilde{A}_q \Rightarrow C_q) = \frac{\sum_{x_s\in ClassC_q} \overline{fs^{jt}}(x_s)}{\sum_{j=1}^{m} \overline{fs^{jt}}(x_s)} \tag{16}$$

$$\underline{c}(\tilde{A}_q \Rightarrow C_q) = \frac{\sum_{x_s\in Classc_q} \underline{fs^{jt}}(x_s)}{\sum_{j=1}^{m} \underline{fs^{jt}}(x_s)} \tag{17}$$

The scaled confidence can be viewed as measuring the validity of $(A_q \Rightarrow C_q)$. The confidence can be viewed as a numerical approximation of the conditional probability (Ishibuchi 2001b). The scaled support (defined by its upper bound $\bar{s}$ and lower bound s, it is scaled as it involves the scaled firing strengths mentioned in the step above) is written as follows:

$$\bar{s}(\tilde{A}_q \Rightarrow C_q) = \frac{\sum_{x_s \in ClassC_q} \overline{fs^{jt}}(x_s)}{m} \tag{18}$$

$$\underline{s}(\tilde{A}_q \Rightarrow C_q) = \frac{\sum_{x_s \in Classc_q} \underline{fs^{jt}}(x_s)}{m} \tag{19}$$

The support can be viewed as measuring the coverage of training patterns by $(A_q \Rightarrow C_q)$. The scaled dominance, (defined by its upper bound $\bar{d}$ and lower bound $\underline{d}$) can now be calculated by multiplying the scaled support and scaled confidence of the rule as follows:

$$\bar{d}(\tilde{A}_q \Rightarrow C_q) = \bar{c}(\tilde{A}_q \Rightarrow C_q) \cdot \bar{s}(\tilde{A}_q \Rightarrow C_q) \tag{20}$$

$$\underline{d}(\tilde{A}_q \Rightarrow C_q) = \underline{c}(\tilde{A}_q \Rightarrow C_q) \cdot \underline{s}(\tilde{A}_q \Rightarrow C_q) \tag{21}$$

The "weighted scaled dominance" (defined by its upper bound $\overline{wd}$ and lower bound $\underline{wd}$) is calculated as follows:

$$\overline{wd}(\tilde{A}_q \Rightarrow C_q) = \bar{d}(\tilde{A}_q \Rightarrow C_q) - \overline{d_{ave}} \tag{22}$$

$$\underline{wd}(\tilde{A}_q \Rightarrow C_q) = \underline{d}(\tilde{A}_q \Rightarrow C_q) - \underline{d_{ave}} \tag{23}$$

where d_{ave} is the average dominance (defined in terms of $\underline{d_{ave}}$ *and* $\overline{d_{ave}}$) over fuzzy rules with the same antecedent $\tilde{A}_q$ but different consequent classes.

For rules that share the same antecedents and have different consequent classes, we will replace these rules by one rule having the same antecedents and the consequent class which will be corresponding to the rule that gives the highest average "weighted scaled dominance value" = $(\frac{\overline{wd}+\underline{wd}}{2})$

In Sanz et al. (2010), the rule generation system generated only the rule with the highest firing strength, however in our method, we generate all rules that are generated by the given input patterns, and this allows covering a bigger area in the decision space.

Step 4: Rule selection: As fuzzy based classification methods generate a large number of rules, this could cause major problems for financial applications where the users need to understand the system. Hence, in our method, we will reduce the rule base to a relatively small pre-specified size of rules that generates a summarized model which could be easily read, understood and analyzed by the user. In this step, we select only the top Y rules per class (Y is pre-specified by the given financial application) which has the rules with the highest average weighted scaled dominance values. This selection is useful because rules with low weighted scaled dominance may not actually be relevant and possibly introduce errors. This helps to keep the classification system more balanced between the majority and minority classes. By the end of this step, the modeling phase is finished where we have $X = nC \cdot Y$ rules (with nC the number of classes) ready to classify and predict incoming patterns as discussed below in the prediction phase.

4.2.2 Prediction phase

When an input pattern is introduced to the generated model, two cases will happen: the first case is when the input $x^{(p)}$ matches any of the X rules in the generated model, in this case we will follow the process explained by *case 1* below. If $x^{(p)}$ does not match any of the existing X rules, we will follow the process explained by *case 2*.

4.2.2.1 Case 1: The input matches one of the existing rules In case the incoming input $x^{(p)}$ matches any of the existing X rules, we will calculate the firing strength of the matched rules according to Eqs. (12) and (13), this will result in $\overline{f^j}(x^{(p)})$, $\underline{f^j}(x^{(p)})$. In this case, the predicted class will be determined by calculating a vote for each class as follows:

$$\bar{Z}Class_h(x^{(p)}) = \frac{\sum_{j\in h} \overline{f^j}(x^{(p)}) * \overline{wd}(A_q \to C_q)}{\max j \in h(\overline{f^j}(x^{(p)}) * \overline{wd}(A_q \to C_q))} \tag{24}$$

$$\underline{Z}Class_h(x^{(p)}) = \frac{\sum_{j\in h} \underline{f^j}(x^{(p)}) * \underline{wd}(A_q \to C_q)}{\max j \in h(\underline{f^j}(x^{(p)}) * \underline{wd}(A_q \to C_q))} \tag{25}$$

In the above equations max $j \in$ h$(\overline{f^j}(x^{(p)}) * \overline{wd}(A_q \to C_q))$ and max $j \in$ h$(\underline{f^j}(x^{(p)}) * \underline{wd}(A_q \to C_q))$ represent taking the maximum of the product of the upper and lower firing strengths and the weighted scaled dominance respectively among the "K" rules selected for each class. The total vote strength is then calculated as:

$$ZClass_h = \frac{\bar{Z}Class_h(x^{(p)}) + \underline{Z}Class_h(x^{(p)})}{2} \tag{26}$$

The class with the highest $ZClass_h$ will be the class predicted for the incoming input vector $x^{(p)}$.

4.2.2.2 Case 2: The input does not match any of the existing rules In case the incoming input vector $x^{(p)}$ does not match any of the existing X rules, we need to decide the output class for the input. The first step is to build all the rules that are possible to be generated from the given input, using the matched fuzzy sets. Let's suppose we have a classification problem with two inputs x_1 and x_2 Let's suppose that a given input $x^{(p)}$will match overall four different fuzzy sets as shown in Fig. 10. Let $MR(x^{(p)})$ the set of rules obtained by combining the matched fuzzy sets. In the example shown in Fig. 10, there will be four matching fuzzy sets which will generate four different rules: $R_1 = \{Small, Medium\}$ $R_2 = \{Small, Large\}$ $R_3 = \{Medium, Medium\}$ $R_4 = \{Medium, Large\}$. Each rule will have an associated a firing strength but not an output class.

The following step is to find the closest rule in the rule base for each rule in $MR(x^{(p)})$. In order to do this, we need to calculate the similarity (or distance) between each of the

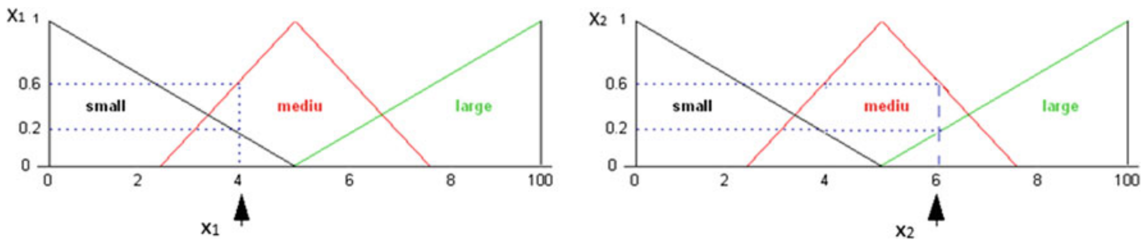


Fig. 10 An example to illustrate the similarity measure

fuzzy rules generated by $x^{(p)}$ and each of the X rules stored in the rule base. Let's define "k" to be the number of *rules* generated from the input $x^{(p)}$ ($k = 4$ in the example shown in Fig. 10). Let the linguistic labels that fit $x^{(p)}$ be written as $v_{inputr} = (v_{input1r}, v_{input2r}, \ldots, v_{inputnr})$ where r is the index of the r-th rule generated from the input. Let the linguistic labels corresponding to a given rule in the rule base be $v_j = (v_{j1}, v_{j2}, \ldots, v_{jn})$. Each of these linguistic labels (*Low, Medium, etc*) could be decoded into an integer. Hence the similarity between the rule generated by $x^{(p)}$ and a given rule in the rule base could be calculated by finding the distance between the two vectors as follows:

$$Similarity_{inputr\leftrightarrow j} = \left((1 - \left|\frac{vinput1r - vj1}{V1}\right|\right) * \left(1 - \left|\frac{vinput2r - vj2}{V2}\right|\right) * \cdots * \left(1 - \left|\frac{vinputnr - vjn}{Vn}\right|\right) \tag{27}$$

where $V_1 \ldots V_n$ represents the number of linguistic labels representing each variable. Each rule in the rule-base will have at this point a similarity associated with the r-th rule generated form the input. For each rule in $MR(x^{(p)})$ the most similar rule in the rulebase, using Eq. (27) will be found to decide on the output class. There will be "k" rules (the most similar rules to the k rules in $MR(x^{(p)})$) selected to decide for the $x^{(p)}$ input the output class. The predicted class will be determined as a vote for each class as follows:

$$\bar{Z}Class_h(x^{(p)}) = \frac{\sum_{j\in \mathrm{h}} \overline{wd}(A_q \to C_q) * \overline{f^j}(x^{(p)})}{\max\ j \in \mathrm{h}(\overline{f^j}(x^{(p)}) * \overline{wd}(A_q \to C_q))} \tag{28}$$

$$\underline{Z}Class_h(x^{(p)}) = \frac{\sum_{j\in \mathrm{h}} \underline{wd}(A_q \to C_q) * \underline{f^j}(x^{(p)})}{\max\ j \in \mathrm{h}(\underline{f^j}(x^{(p)}) * \underline{wd}(A_q \to C_q))} \tag{29}$$

where $\underline{f^j}(x^{(p)})$ and $\overline{f^j}(x^{(p)})$ are the lower and upper firing strength of the most similar rule in the rule base and $\overline{wd}(A_q \to C_q)$ and $\underline{wd}(A_q \to C_q)$ are the upper and lower interval of the weighted scaled dominance of the most similar rule of the rule considered in $MR(x^{(p)})$. In the above equation $\max\ j \in h(\overline{f^j}(x^{(p)}) * \overline{wd}(A_q \to C_q))$ and $\max\ j \in h(\underline{f^j}(x^{(p)}) * \underline{wd}(A_q \to C_q))$ represent taking the maximum of the product of the upper and lower firing strengths and the weighted scaled dominance respectively among the most similar rules to the "k" rules, this measure is used to scale the lower and upper voting strength of each class. The total vote strength is then calculated as:

$$ZClass_{hr} = \frac{ZClass_h(x^{(p)}) + \underline{Z}Class_h(x^{(p)})}{2} \tag{30}$$

The class with the highest $ZClass_h$ will be the class associated $x^{(p)}$.

5 Evaluations and results

The classification ability of the proposed system has been tested with two different datasets. The first sets of experiments are based on the data which has been used for spotting arbitrage opportunities in the London International Financial Futures Exchange (LIFFE) market (Garcia-Almanza and Tsang 2006).

The second one is a credit approval dataset. This dataset was achieved from a real-world credit reference agency identifying good and bad customers where good customers are profitable customers and bad customers are non-profitable customers.

5.1 Performance on arbitrage dataset

We have tested the proposed genetic type-2 FLS to model and predict arbitrage opportunities. Computers today are able to spot in milliseconds the stock misalignment in the market. This would allow them to make almost risk-free profits. There are two main challenges in this type of operation. Firstly, arbitrage situation do not occur very often. Secondly, the operator must act ahead of others, so the competition is reduced to how fast a computer is, and how fast its connection to the stock exchange is. Garcia-Almanza and Tsang (2006) showed that arbitrage opportunities do not appear instantaneously. There are patterns in the market which can be recognized 10 min ahead.

The proposed system is trained to identify ahead of time arbitrage opportunities (Garcia-Almanza 2008). The data reported in this paper was further developed in Garcia-Almanza (2008), Garcia-Almanza and Tsang (2008) in order

Table 2 The seven input features (variables) for the arbitrage data set (Garcia-Almanza and Tsang 2008)

No	Name	Description
1	Moneyness	Strike price/underlving index level
2	Basis % (×100,00)	Futures price minus spot index level.divided by futures price, multiplied by 10,000
3	Und (×10)	Spot index level divided by futures price, multiplied by 10
4	Interest ask %	The LIBOR ask rate for the maturity closest to the maturity of futures contract, multiplied by 100
5	Futures (T-t)	The nave trigger, profit after transaction costs, divided by futures price, multiplied by 1,000,000
6	C-P % (×l00)	The difference between the call and the put prices, divided by futures price
7	Profit after TC (×1,000,000)	The nave trigger, profit after transaction costs, divided by futures price, multiplied by 1,000,000

to identify arbitrage situations by analyzing option and futures prices in the London International Financial Futures Exchange (LIFFE) market. The pre-processed data comprised 1,641 instances of which only 401 representing arbitrage opportunities and the rest representing non-arbitrage opportunities. The data was split into 2/3 for modelling and 1/3 for testing.

According to Garcia-Almanza and Tsang (2008), the information used from the option and future prices in the London International Financial Futures Exchange (LIFFE) market have been manipulated, selected and reduced to just seven features. Those are described in Table 2.

We have compared the proposed genetic type-2 FLS approach with one of the most powerful white box modelling and prediction systems for spotting arbitrage opportunities which is Evolving Decision Rule (EDR) procedure (Garcia-Almanza and Tsang 2008). The EDR method evolves a set of decision rules by using Genetic Programming (GP) and it receives feedback from a key element that is called *repository*. The repository is a structure whose objective is to store a set of rules. The resulting rules are used to create a range of classifications that allows the user to choose the best trade-off between misclassifications and false alarms cost.

We have compared as well the proposed genetic type-2 FLS approach against Neural Networks which was found to give a better performance than any other black box model available for this data set.

The proposed genetic type-2 FLS aims to fulfil two objectives: The first one is to get good results on both RECALL and false positive rate, the second objective is to use small number of rules to model and predict the arbitrage opportunities, thus presenting a white box model which could be easily understood and analyzed by the lay user. The perfect ideal classifier is able to have a RECALL (True Positive Rate) of 1 and a False Positive Rate (FPR) of 0 (where the area under the ROC curve will be equal to 1), thus the more predictive the given model is, the higher is its ROC curve which means that the ROC curve for the given prediction model approaches the ideal classifier. In general, this means having the highest RECALL possible and the lowest false positive rate possible. Hence, the more predictive a given model is, the more the area under its ROC curve will approach the ideal classifier whose area under its ROC curve equals to one.

Moving along the Receiver Operating Characteristic (ROC) curve (which plots the true positive rate, vs. false positive rate) means increasing the FPR at the expenses of the RECALL or vice versa.

In the following evaluations, we have employed the proposed Genetic type-2 FLS with different fuzzy set space configurations in order to move along the ROC curve. In order to do so the fitness function of the GA was weighed using different weights in Eq. (9). Figure 11 shows the ROC curve obtained over testing data by the proposed genetic type-2 FLS plotted against the ROC curves obtained by the EDR procedure (Garcia-Almanza and Tsang 2008) and the Neural Networks respectively. From Fig. 11, it is obvious that the proposed genetic type-2 FLS gives a better ROC curve than the EDR procedure and the Neural Networks while the type-2 FLS presents the user with a small number of rules which summarizes the model and explains the system behaviour to the lay user in an understandable and comprehensible way. Figure 11 shows the results obtained when employing the proposed genetic type 2 FLS with only 200 and 40 rules. The best results are obtained using 200 rules; the genetic type-2 FLS with just 40 rules has slightly worst performance than the performance with 200 rules but still produces much better results than the EDR procedure and also producing a slightly better performance when compared to Neural Networks. The selected 40 or 200 rules have been selected taking those with highest average weighted scaled dominance.

In order to compare the classifier on their overall behaviour (including all range of riskier and risk-averse classifiers) the area under the curve (AUC) technique has been used. The machine learning community can use the ROC AUC statistic for model comparison (Hanley and McNeil 1983). The area under curve (AUC), when using normalized units, is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative') (Fawcett 2006). Table 3 summarise the AUC results for the classifier

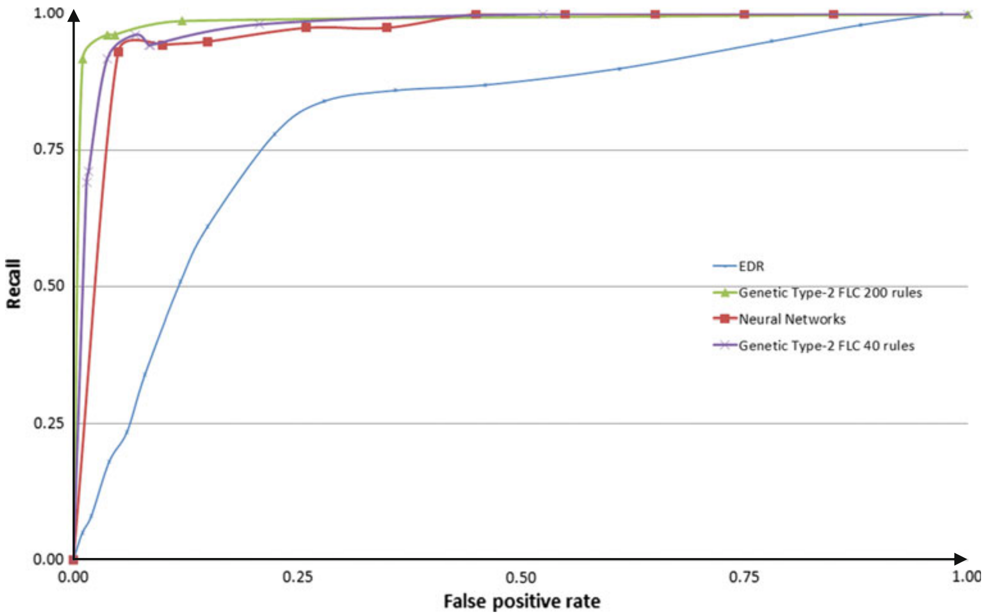


Fig. 11 ROC graph over testing data for the arbitrage prediction comparing EDR, NN and genetic type-2 FLSs

Table 3 AUC results for the arbitrage data

Classifier	AUC
GA T2 FLS 200 rules	0.9849
GA T-2 FLS 40 rules	0.9755
NN	0.9607
EDR	0.8039

used where it is shown that the AUC for the genetic type-2 FLS with 200 rules gives the best AUC of 0.9849 followed by the AUC of 0.9755 for the genetic type-2 FLS with 40 rules which gave better performances than the Neural Networks classifier which gives an AUC of 0.9607 followed by the AUC 0.8039 for EDR.

The best average recall overall obtained on this dataset is 96.22 % with the Genetic Type-2 FLS with 200 rules. The Genetic FLS with 40 rules best average recall performance was 94.64 %. The best average recall obtained by the neural network was 94.04 %. The EDR best average recall was around 78 %.

These results shows that the proposed genetic type-2 FLS gives a better performance when compared to a white box model like the EDR procedure. The proposed genetic type-2 FLS gave also better performance when compared to black box models which shows that the proposed genetic type-2 FLS can achieve a similar or even better performance than black box models while providing a transparent white box model with a summarised number of rules which are easy to understand and analyse by the lay user.

It should be emphasised that all the inputs for the arbitrage data were continuous inputs which allowed the FLSs to give the best results. However in data employed in the next subsection the inputs will be both continuous and categorical/discrete (like gender) which will not allow the FLSs to give its best performance, however the FLSs will be able to give good comparable results to black box models which giving a white box model.

5.2 Performance on credit approval dataset

With the economic crisis of the recent year, prime and subprime credit requests continuously expanded. Unfortunately with the increase of the number of people asking for credit, the number of people not being able to repay increased as well. The ability of finding and declining bad credit request has been crucial for lenders and provide a huge money saving. Lots of effort is nowadays put in finding the best techniques to reduce the risk in the lending market. The proposed system has been tested using data gathered from a credit lender company. The dataset built includes the information that a customer would provide when asking for credit. The dataset is composed by 123,116 records and 10 features divided into 3/4 for the training set (92,397 records) and 1/4 for the testing set (30,720 records). The 40 % of the training set is used during the GA tuning as validation set. The system need to be able to find bad credit requests but as well need not to decline too many requests. As the matter of the fact a simple approach to avoid all bad credit requests would be simply of not accepting any requests at all, but of course this extreme scenario is not feasible because this does not match with

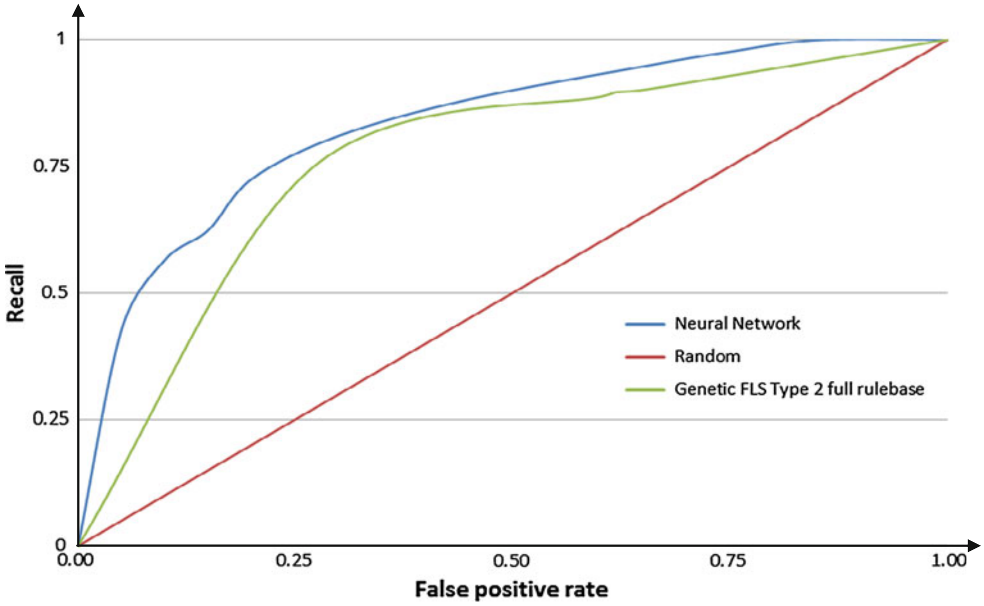


Fig. 12 Credit approval dataset ROC curve over testing data

Table 4 Credit approval AUC and best average recall

Classifier	AUC	Best AVG Recall
Neural network	0.8303	76.25 %
GA T2 FLS full rule-base	0.748	73.84 %

the business model of credit lenders. The data is extremely unbalanced as it presents 98.54 % of customer belonging to class 0 (Good Customers) inputs and 1.46 % of customers belonging to class 1 (Bad Customers). Different types of our classifiers have been built for this reason, providing a more *risk averse or riskier* approach thus accepting more or less requests. It is worth mentioning that the data is quite different from the Arbitrage data set mentioned in the previous subsection where this data set is very noisy and sparse the hence the prediction accuracy of any prediction system will be rather limited.

The genetic type 2 FLS was compared to neural networks which were found to be the best black box model suited for this data set. Figure 12 shows the ROC curve over the testing data of the proposed genetic type-2 FLS plotted against the Neural Networks ROC curve. Table 4 summarise the best average recall and the area under the curve for both techniques.

The best performances on this dataset are obtained by the neural network. The reason is mainly because half of the features on this dataset are categorical, and the FLS at the moment reduce its performance when dealing with these kinds of features. However, it can be seen that over this unbalanced noisy data, the proposed genetic type-2 FLS produced comparable results to black box models like Neural Networks while the proposed type-2 FLS produced a white box model that could be easily understood and analysed by the lay user.

5.3 Comparison between the proposed weighted scaled dominance and other measures employed in Fuzzy classification systems

In this paper, we have presented a new measure called ***weighted scaled dominance*** which is based on the extension of the “scaled dominance” measure introduced in our previous work and the weighted confidence measure introduced by Ishibuchi and Yamamoto (2005). The aim of this new measure is to give more weight to associations that do not occur very often in the dataset, especially those associations affiliated with the minority classes and hence described in the decision space by few samples. In most of the cases the minority classes are usually the relevant class to identify, and scarcity of the samples makes the problem more challenging. There exist in the literature other measures that aim to give more importance to infrequent but still important associations; one of these measures is the *lift*. The *lift* can be defined as the combined support of the consequent and the antecedent of a rule, over the support of the antecedent multiplied by the support of the consequent (Tufféry 2011).

$$l(\tilde{A}_q \Rightarrow C_q) = \frac{s(\tilde{A}_q \Rightarrow C_q)}{s(\tilde{A}_q) * s(C_q)} \tag{31}$$

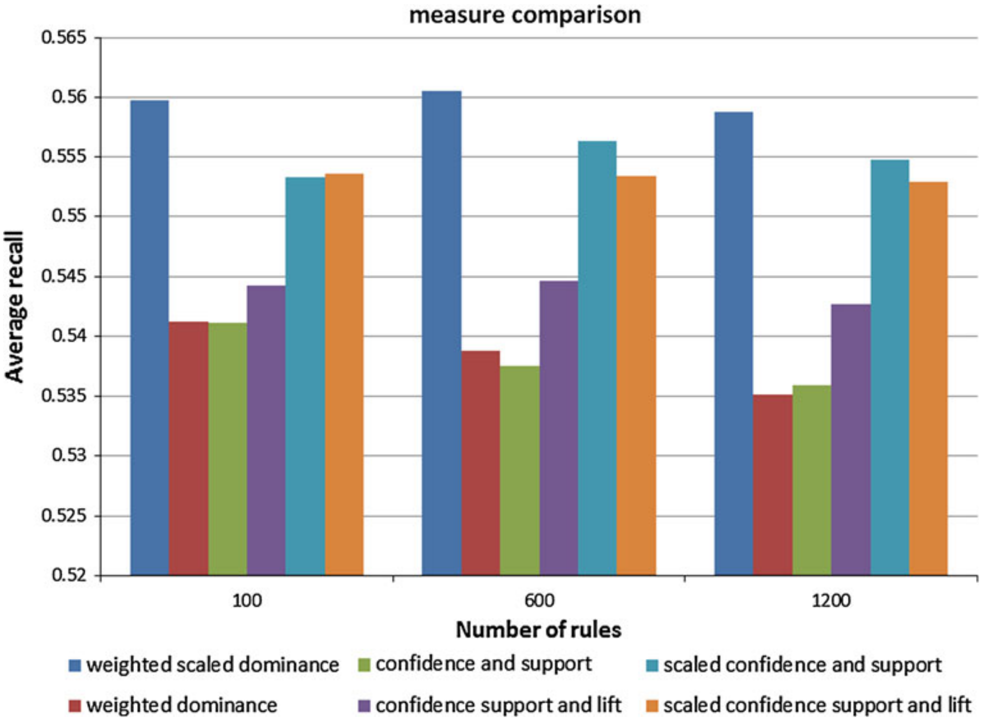


Fig. 13 Comparison between the proposed weighted scaled dominance and other measures employed in fuzzy classification systems

Equation (31) can be rewritten using the definition of confidence in Eqs. (16) and (17) as follows:

$$l(\tilde{A}_q \Rightarrow C_q) = \frac{c(\tilde{A}_q \Rightarrow C_q)}{s(C_q)} \tag{32}$$

The nominator is the confidence of the rule and the denominator of the equation represent the support of the consequent class. Another important measure is the weighted dominance which is obtained simply by the multiplication of scaled confidence and scaled support, explained in Ishibuchi and Yamamoto (2004, 2005). Those metrics are defined as follows:

$$wc(\tilde{A}_q \Rightarrow C_q) = c(\tilde{A}_q \Rightarrow C_q) - c_{ave} \tag{33}$$

$$ws(\tilde{A}_q \Rightarrow C_q) = s(\tilde{A}_q \Rightarrow C_q) - s_{ave} \tag{34}$$

$$wd(\tilde{A}_q \Rightarrow C_q) = wc(\tilde{A}_q \Rightarrow C_q) * ws(\tilde{A}_q \Rightarrow C_q) \tag{35}$$

where c_{ave} and s_{ave} are the average confidence and support over fuzzy rules with the same antecedent $\tilde{A}_q$ but different consequent classes (Ishibuchi and Yamamoto 2004, 2005).

We compared the results obtained by the different metrics using the same fuzzy sets. We have compared the metrics over various data sets but due to the space limitation, we will report only the results achieved over a complicated noisy data from the banking credit evaluation system (different from the data sets employed above). Figure 13 summarise the results obtained with the different data mining measures with varying the size of the rule bases used for classification.

As shown in Fig. 13, the best result (with best average recall) was obtained employing the suggested ***weighted scaled dominance*** measure as described in Eq. (22), (23) or in general whenever the ***scaling*** procedure is used in any technique. The scaling procedure is applied by employing the scaled firing strength described in Eqs. (14)–(15). When the scaling is not used the next best results are obtained with *lift* described by Eq. (32). This underlines the point that infrequent association can as well be very important, and this is extremely true in imbalanced datasets. The ***weighted dominance*** is described by Eqs. (33)–(35) but without using the scaling procedure it did not get good results, as the matter of the fact this measure did not use the scaled firing strength as in the ***weighted scaled dominance***. As can be seen, the comparisons have been conducted with different pre specified rule sizes. The rule selection has been performed by selecting those only with highest metrics in question.

5.4 Evaluation of the performance of the proposed similarity technique

The quality of a fuzzy logic system is positively correlated with the quality of its rule-base. But what happen though if an input does not match any rule in the rule base? In the majority of fuzzy classification systems, there are two main approaches to handle a situation when the incoming input does not match any rules from the FLS rule base. The first approach is to reject the input, and do not give a prediction for

Table 5 Confusion matrix

	Actual positive	Actual negative
Positive prediction	700 (TP)	300 (FP)
Negative prediction	0 (FN)	0 (TN)
Recall (%)	100	0

this given input, and hence do not consider it in the confusion matrix and in the calculation of the *recall* and *false positive rate*. The second approach is to build a default rule that fires every time an input does not fire any rules from the rule base.

The first approach is unacceptable solution for the financial domain where the prediction system should always be able to provide a prediction. The second approach could be acceptable in the case of highly unbalanced datasets and in case the system has only two output classes but overall it is not a strong solution as the matter of the fact it does not improve the quality of the classifier and will have problems when there is a big number of output classes. In fact on a two class problem, this approach by definition will always produce the same average recall improvement (for inputs not matching rules from the rule base) regardless of the class chosen as default option. Let's consider for example a dataset where there are 1000 inputs that do not match any rule in the rule-base. Let's suppose then that 700 of these cases are actually class 1 while 300 are actually class 0. If we create a default rule which says that all of these 1,000 unmatched inputs must be mapped as Class 1, we will have the confusion matrix in Table 5.

From the example, even though the dataset is unbalanced and we chose the rule that would have made better sense, the achieved average recall is only 50 %, as the matter of fact regardless of the default rule we choose the recall on the default class by definition will always be 100 % and the recall will be 0 % on the other class and so the average recall will always be 50 % on the cases of inputs not matching rules from the rule base. Hence, the contribution that the default rule will give to the classifier will always be the same minimal (50 %) contribution (on the recall measure) regardless the output class chosen.

Our proposed techniques on the other hand aim to find the most similar rules from the rule base (as discussed above in Sect. 4.2.2.2, and choose by using the *weighted scaled dominance* approach the output class. This section will show the results obtained on both arbitrage and credit approval datasets mentioned above by the proposed similarity approach.

On the arbitrage data, we tried various pre specified rule base sizes of 10 rules, 20 rules, 30 rules and 40 rules. The numbers of cases where the inputs did not match any rules in the rule base were 97 (in case of using a rule base of 10 rules), 62 (in case of using a rule base of 20 rules), 56 (in case of using a rule base of 30 rules) and 34 (in case of using a rule base of 40 rules). Figure 14 shows the comparison on testing data on the average recall obtained when using similarity technique and the default rule technique. As shown in Fig. 14, when employing the similarity technique with just 5 rules per class (10 rules in the rule-base) the cases of inputs not matching rules in the rule base were 97, and the achieved average recall on cases where the inputs not matching rules from the rule base was 63.9 % (compared to 50 % average recall when using any default rule approach). The similarity method gives better results when increasing the number of rules in the rule-base because the decision space is better represented by the rules in the rule-base so the similarity can find more appropriate similar rules. On a rule-base of 40 rules the similarity is able to achieve an average recall of 100 % on cases where the inputs not matching rules from the rule base, thus the similarity technique was able to correctly classify all the 34 cases where the inputs did not match rules from the rule base.

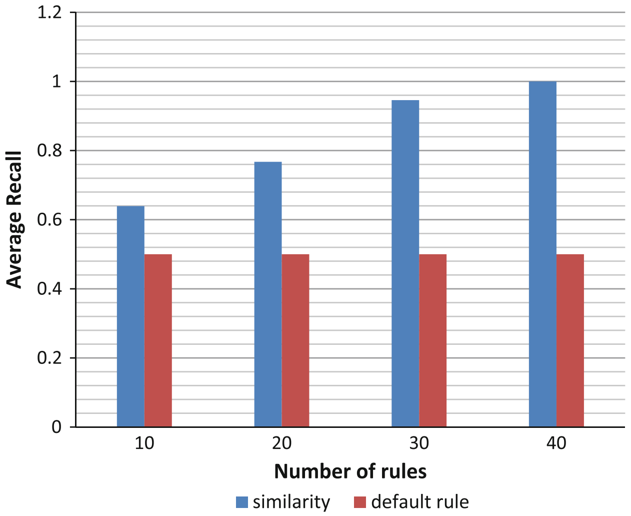


Fig. 14 Comparison between the similarity and default rule technique on arbitrage data set

The similarity measure was tested as well on the credit approval dataset which presented more complex features where the dataset have unordered categorical features on which it is difficult to create an ordered relationship (like for example credit card type). On these type of features a default distance of 1/(tot number of labels −1) was given, and the similarity formula in Eq. (27) is changed accordingly. The similarity technique on this dataset has been tested with a rule-base of 50, 100, 150, 200, 250, 300 and 350 rules. The credit approval data is a bigger dataset and highly unbalanced and the number of cases where the inputs do not match any of the rules in the rule base were 2,426, 748, 429, 319, 188, 125, 92 in case of using 50, 100, 150, 200, 250, 300 and 350 rules respectively. The dataset present 98.54 % of class 0 and 1.46 % of class 1. Figure 15 shows the average recall com-

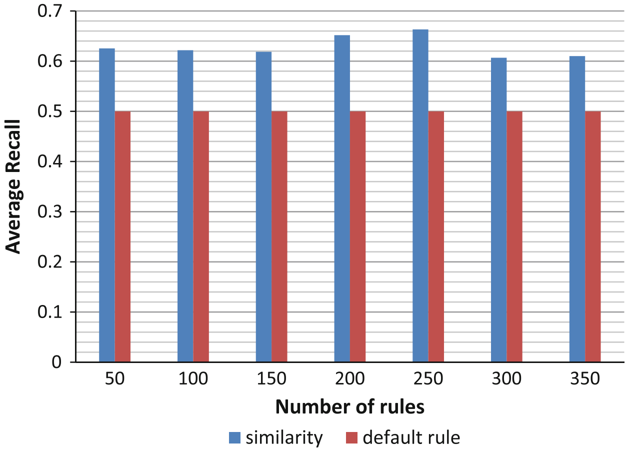


Fig. 15 Comparison between the similarity and default rule technique on credit approval data set

parison when employing the similarity and the default rule technique. It can be seen that again the similarity technique gives better average recall (compared to the default rule) on the cases where the inputs do not match any rules from the rule base

6 Conclusions and future work

The global economic meltdown of the late 2000s exposed many organisations around the world, this drove the need to build robust frameworks for predicting and assessing risks in financial applications. In the current economic situation, transparency became an important factor where there is a need to fully understand and analyze a given financial model. In this paper, we have presented a genetic type-2 FLS capable of generating summarized models from pre-specified number of linguistic rules, which enables the user to understand the generated financial model, thus generating a transparent and easy to read and analyse model. The proposed system was tested on two different datasets.

We have shown how the proposed genetic system allows learning the various parameters of the input type-2 fuzzy sets which cannot be easily designed or manually tuned.

We have presented two novel measures, the first measure is a data-mining measure called weighted scaled dominance. The performance of this novel technique was compared against other classic data-mining metrics and it was shown that the proposed weighted scaled dominance outperformed other widely used measures. The second presented measure was called similarity measure which is a technique used to be able to provide a classification even when the inputs do not match any rules from the rule base. This technique was implemented to avoid a commonly used approach of discharging any inputs that do not match any rule in the rule-base. We have also shown the improvements of the proposed similarity measures over the default rule where we have shown that for inputs not matching rules in the rule base, the proposed similarity measure result in a considerable uplift in the average recall when compared to using the default rule which cannot be easily used when the number of output classes is more than 2.

We have performed several evaluations in two distinctive financial domains one for the prediction of good/bad customers in a financial real-world lending application and the other domain was in the prediction of arbitrage opportunities in the stock markets. The proposed Genetic type-2 FLS has outperformed white box models like the Evolving Decision Rule (EDR) procedure (which is a white based on Genetic Programming (GP) and decision trees) and gave a comparable performance to black box models like neural networks while the proposed genetic type-2 FLS provided a white box model which is easy to understand and analyse by the lay user.

In financial applications, there is a need to have clear, transparent and easy to understand models which stresses the importance of increasing the interpretability of the given financial model. Hence, for our future work, we will aim to optimize also the length of the rules and use do not care conditions to make the genetic type-2 FLC easier to read by the lay user. In Ishibuchi and Nojima (2007), the trade-off between interpretability and accuracy of type-1 fuzzy systems has been discussed and how accuracy can be affected when trying to build interpretable systems. In our future work, we aim to carry out the same analysis as Ishibuchi and Nojima (2007) for fuzzy type-2 systems and investigate how this trade off affects type-2 systems.

References

Ahmad S, Jahormi M (2007) Construction accurate fuzzy classification systems: a new approach using weighted fuzzy rules. Computer graphics, imaging and visualisation, pp 408–413

Casillas J, Cordon O, Herrera F, Magdalena L (2003a) Accuracy improvements in linguistic fuzzy modeling. Springer, Berlin

Casillas J, Cordon O, Herrera F, Magdalena L (2003b) Interpretability issues in fuzzy modeling. Springer, Berlin

Cohen J, Cohen P, West S, Aiken L (2003) Applied multiple regression/correlation analysis for the behavioral sciences, 2nd edn. Lawrence Erlbaum Associates, Hillsdale

Deaton A (1992) Understanding consumption. Oxford University Press, Oxford

Ehrenberg A, Smith (2008) Modern labor economics. HarperCollins

Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27:861–874

Garcia-Almanza A (2008) New classification methods for gathering patterns in the context of genetic programming, PhD Thesis. Department of Computing and Electronic Systems, University of Essex, Essex

Garcia-Almanza A, Tsang E (January 2008) Evolving decision rules to predict investment opportunities. Int J Autom Comput 5(1):22–31

Garcia-Almanza A, Tsang E (2006) Forecasting stock prices using genetic programming and chance discovery. 12th International conference on computing in economics and, finance

Giacomini E (2003) Neural networks in quantitative finance. Master Thesis. Humboldt-Universit atzu Berlin, Berlin

Hagras H (2004) A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE Trans Fuzzy Syst 12(4):524–539

Hanley J, McNeil B (1983) A method of comparing the areas under receiver operating characteristic curves derived from the same cases. Radiology 3:148

Ishibuchi H (2001a) Effect of rule weights in fuzzy rule-based classification system. IEEE Trans Fuzzy Syst 9(4):506–515

Ishibuchi H (2001b) Three-objective genetic-based machine learning for linguistic rule extraction. Inf Sci 136(1–4):109–133

Ishibuchi H, Nojima Y (2007) Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. Int J Approx Reason 44(1):4–31

Ishibuchi H, Nakashima T, Murata T (1999) Performance evaluation of fuzzy classifier systems for multi-dimensional pattern classification problems. IEEE Trans Syst Man Cybern Part B Cybern 29(5):601–618

Ishibuchi H, Yamamoto T (2004) Fuzzy rule selection by multy-objective genetic local search algorithms and rule evaluation measures in data mining. Inf Sci 141(1):59–88

Ishibuchi H, Yamamoto T (2005) Rule weight specification in fuzzy rule-based classification systems. IEEE Trans Fuzzy Syst 13(4):428–435

Ishibuchi H, Yamamoto T, Nakashima T (2006) An approach to fuzzy default reasoning for function approximation. Soft Comput 10(9):850–864

Kassem S (2012) A type2 fuzzy logic system for workforce management in the telecommunications domain, MSc, Thesis. University of Essex, Essex

Kim K (2003) Financial time series forecasting using support vector machines. Neurocomputing 55(1–2):307–319

Kim HS, Sohn SY (2010) Support vector machines for default prediction of SMEs based on technology credit. Eur J Oper Res 3:838–846

Kohavi R, Provost F (1998) Glossary of terms. Edited for the special issue on applications of machine ;earning and the knowledge discovery process, vol 30

Krugman P, Obstfeld M (1988) International economics: theory and policy. Scott, Foresman

Kwong K (2001) Financial forecasting using neural network or machine learning techniques. University of Queensland, Queensland

Laidler D (1993) The demand for money: theories, evidence, and problems

Lawrence R (1997) Using neural networks to forecast stock market prices. University of Manitoba, Manitoba

Levinson M (2006) Guide to financial markets. The Economist. Profile Books, London, pp 145–6

Mansoori E, ZolGhadri M, Katebi D (2006) Using distribution of data to enhance performance of fuzzy classification system. Iran J Fuzzy Syst 4(1)

Mendel J (2001) Uncertain rule-based fuzzy logic systems: introduction and new directions. Prentice-Hall, New Jersey

Sanz J, Fernandez A, Bustince H, Herrera F (2010) Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning. Inf Sci 180:3674–3685

Sanz J, Fernandez A, Bustince H, Herrera F (2011) A genetic tuning to improve the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets: degree of ignorance and lateral position. Int J Approx Reason 52:751–766

Shakya S (2004) Markov random field modeling of genetic algorithms. Progress report submitted to The Robert Gordon University to make the case for transfer from MPhil to PhD, The Robert Gordon University

Shigeo A, Lan M (1995) A method for fuzzy rules extraction directly from numerical data and its application in pattern classification. IEEE Trans Fuzzy Syst 3(1):18–28

Swets J (1996) Signal detection theory and ROC analysis in psychology and diagnostics: collected papers. Lawrence Erlbaum Associates, Mahwah

Tufféry S (2011) Data mining and statistics for decision making, Chichester

Wang L (2003) The WM method completed: a flexible fuzzy system approach to data mining. IEEE Trans Fuzzy Syst 11(6):768–782