Evolving Decision Rules to Discover Patterns in Financial Data Sets

Alma Lilia García-Almanza, Edward P.K. Tsang, and Edgar Galván-López

Department of Computer Science, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, U.K. algarc@essex.ac.uk, edward@essex.ac.uk, egalva@essex.ac.uk

Summary. A novel approached, called *Evolving Comprehensible Rules* (ECR), is presented to discover patterns in financial data sets to detect investment opportunities. ECR is designed to classify in extreme imbalanced environments. This is particularly useful in financial forecasting given that very often the number of profitable chances is scarce. The proposed approach offers a range of solutions to suit the investor's risk guidelines and so, the user could choose the best trade-off between miss-classification and false alarm costs according to the investor's requirements. Receiver Operating Characteristics (ROC) curve and the Area Under the ROC (AUC) have been used to measure the performance of ECR. Following from this analysis, the results obtained by our approach have been compared with those one found by standard Genetic Programming (GP), EDDIE-ARB and C.5, which show that our approach can be effectively used in data sets with rare positive instances.

Key words: Evolving comprehensible rules, machine learning, evolutionary algorithms.

1 Introduction

In this work, we propose to evolve decision rules by using Genetic Programming(GP) Koza (1992a). The result of this is a reliable classifier which is able to detect positive cases in imbalanced environments. This work is aided by Machine Learning (ML), an Artificial Intelligence field that has been successfully applied to financial problems Chen (2002),Chen and Wang (2004), Tsang and Martinez-Jaramillo (2004). ML embraces techniques to extract rules and patterns from data. These, like other forecasting techniques, extend past experiences into the future. However, in rare event detection, the imbalance between positive and negative cases poses a serious challenge to ML Japkowicz (2000),Weiss (2004),Batista et al. (2004),McCarthy et al. (2005). This is due to negative classifications are favoured given that these have a high chance of being correct, as we shall explain later.

DRAFT October 12, 2007 - 10:06

This work is organised as follows. Section 2 provides an introduction of the Machine Learning and Evolutionary Computation systems. Section 3 describes some metrics to measure the performance of a learning classifier system and provides a description of the Receiver Operating Characteristic (ROC) curve. Section 4 describes in detail our approach. Finally, Section 5 describes the results found by our approach and we draw some conclusions in Section 6.

2 Previous Work

ML, a field of Artificial Intelligence (see Mitchell (1997)), is a multidisciplinary area which embraces probability and statistics, complexity theory, Evolutionary Computation (EC) and other disciplines. ML is often used to create computer programs that form new knowledge (which is acquired by information and data analysis) for a specific application. Our approach is based on supervised learning, a branch of ML in which the system is fed by a training data set which comprises examples with inputs and the corresponding desired outputs.

In this paper, we focus on using GP to evolve rules. Other methods have also been used to evolve decision rules; for instance, Corcoran and Sen (1994) used a Genetic Algorithm (GA) to evolve a set of rules. For this purpose, they treated it as an optimisation problem. That is, the goal of the GA is to maximise the number of correct classifications of a training data set. The contribution of their approach was to evolve rules with continue variables rather than using a binary representation. Bobbin and Yao (1999) were interested in evolving rules for nonlinear controllers. In addition, their approach was able to offer rules that can be interpreted by humans.

Jong and Spears (1991) proposed an application called GA batch-incremental concept learner (GABIL). The idea is to evolve a set of rules using GAs. That is, GAs evolve fixed-length rules for attributes whose values are nominal. Each member in the population is composed by a variable number of rules which means that the individual's size is variable too.

Following the same idea, Janikow Janikow (1993) proposed Genetic-based Inductive Learning (GIL). Janikow proposed three types of operations: (a) rule set level, (b) rule level and (c) condition level. Each of them contains specific operations. For the former, the operations involve rules exchange, rules copy, new event, rule generalisation (generalise two rules picked at random) and rule specialisation. The operations in the rule level are in charge of introduce or drop conditions. Finally, there are 3 operators for the condition level: reference change, reference extension and reference restriction.

Using Evolutionary Algorithms, Kwedlo and Kretowski (1999) proposed an approach which novelty was to use multivariate discretisation. The proposed approach has the ability to search simultaneously for threshold values for all the attributes that hold continuous values.

DRAFT October 12, 2017 19:06

Fidelis et al. (2000) proposed an approach based on GAs in the hope that it can discover comprehensible IF-THEN rules. The novelty of their approach was that it allowed to have a fixed length at genotype level but the number of rules conditions (which is mapped at the phenotype level) is variable. The key ingredient that allows this feature in the genotype-phenotype mapping was to allow an element at the genotype level that the authors called "weight". As the authors explained in their paper, the results regarding to the accuracy were promising in one of the data sets but more importantly, they showed how a GA was able to find concise rules that are more understandable in terms of complexity.

Bojarczuk et al. (1999) were interested in discovering rules to classify 12 different diseases. Moreover, the authors were also interested in producing rules that were comprehensible for the final user. The latter requirement, as the authors expressed in their work, is that the resulting rule can be "readable" and so, it can be used as a complement to the user's knowledge. The authors claimed that the results found by their GP system were promising because they reported high accuracy in classifying the diseases.

Using GP, Falco et al. (2001) also evolved classification rules that were comprehensible for the final user. This approach evolves decision trees that classify a single class and so, if there are more than one class to classify the GP is executed as many times as the number of classes. The authors stated that the tree represents a set of rules which is composed by disjunctions and conjuctions.

Niimi and Tazaki (2000) proposed a discovery rule technique by means of generalisation of association rules using an apriori algorithm, then the rules are converted to decision trees which will be used as the initial population in GP. After evolving the potential solutions, the best individual is converted into classification rules.

The works described above suggest that ECs could perform very well on classification tasks. Our approach, as it will be explained in Section 4, has been designed in such a way to produce rules, which detect rare cases in imbalanced environments. The output is a range of classifications that provides to the user the option to choose the best trade-off between miss-classification and false alarm costs according to the user's needs. Moreover, it presents various beneficial features as comprehensible rules.

3 Performance metrics

The 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 and some learning systems. Given an instance and a classifier, there are four possible results:

- *True positive* (TP): the instance is positive and it is classified as positive.

DRAFT October 12 2007, 15:03

Table 1. Confusion matrix to classify two classes.

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

- *False positive* (FP): the instance is negative and it is classified as positive.
- *False negative* (FN): the instance is positive and it is classified as negative.
- *True negative* (TN): the instance is negative and it is predicted as negative.

Table 1 shows a confusion matrix for two classes. For detailed analysis, we have used other metrics taken the confusion matrix as a basis. These are shown in Table 2.

Table 2. Metrics used for a detail analysis of the results found by our approach.

Accuracy	is the proportion of the total number of predictions that were done correctly. This is determined by the equation: $Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$
Recall	also called *sensitivity* or *true positive rate*, it is the proportion of positive cases that was correctly identified. This is given by: $Recall = \frac{TP}{TP+FN}$
Precision	also called *positive predictive value*, is the proportion of positive cases that were correctly predicted. It is calculated as follows: $Precision = \frac{TP}{TP+FP}$
False positive rate	also known as *false alarm rate*, it is the proportion of negative cases that were wrongly predicted as positive. It is calculated as follows: $False\ positive\ rate = \frac{FP}{FP+TN}$

DRAFT October 12, 2017, 15:06

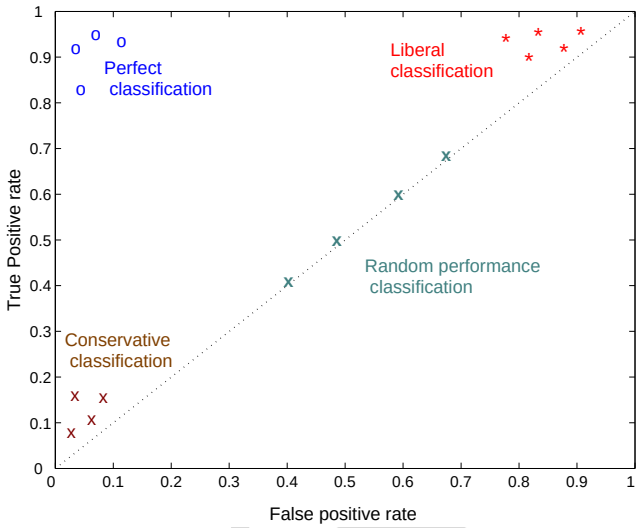

Fig. 1. Receiver Operating Characteristic (ROC) space.

3.1 ROC space

Receiver Operating Characteristic (ROC) is a technique that plots the performance of a classifier and is able to select the best trade-off between successful cases and false alarms based on benefits and costs Fawcett (2004).

The ROC graph is constructed by plotting the *true positive rate* (recall) on the Y-axis and the *false positive rate* (false alarms) on the X-axis (see for instance Hanley and McNeil (1998),Fawcett (2004)). Figure 1 depicts the ROC space. As can be seen, ROC graph is plotted in the space of (0,0) and (1,1). The performance of the classifier is plotted in (0,0) when it does not find any positive case and it does not report any false alarm. Thus, it gets all the negative cases right but it gets all the positive wrong. The opposite case is at position (1,1), where the totality of the cases are classified as positive.

The performance of a classifier is better than other if it is plotted in the upper left area of the ROC space Fawcett (2004). The classifiers whose performance is plotted in the left hand side of the ROC space close to the X-axis, are denominated *conservative*. This is because these make a positive classification just when these have strong indications or evidences; as a consequence, these have few false alarms. On the other hand, classifiers on the upper right hand side of the ROC space are called *liberal* because these make positive classifications with unsubstantial evidence. Finally, the diagonal line that goes from position (0,0) to (1,1) describes classifiers whose performance is no different from those made by random predictions.

DRAFT October 12, 2007, 16:06

3.2 Area Under the ROC Curve (AUC)

One of the most used ROC metrics is the Area Under the Curve (AUC) which indicates the quality of the classifier (Vanagas (2004)). Huang and Ling (2005) showed theoretically and empirically that AUC is a better measure than accuracy.

When $AUC = 1$, the classification is perfect. In general, the closer AUC is to 1 the better performance of the classifier is. When a classifier's AUC is close to 0.5, it represents a random classifier performance.

3.3 Choosing the Best Operating Point

ROC can be used to estimate the best threshold of the classifier by calculating the best balance between the cost of misclassifying positive and negative cases. To calculate the best trade-off, let us define the following variables:

μ The cost of false positive or false alarm
β The cost of false negative
ρ the percentage of positive cases

Thus, the slope is calculated by the following formula:

$$Slope = \mu \cdot (1 - \rho)/(\beta \cdot \rho)$$

The point where the line is tangent to the curve indicates the threshold of the best trade-off between misclassifications and false alarms costs.

We have selected ROC as a performance measure for the following reasons. Firstly, ROC is suitable for measuring the performance of classifiers for imbalanced data sets. Secondly, it is able to measure the performance of a classifier that is base on thresholds. Thirdly, ROC offers the possibility to the user (investor) to tune the parameters in a way to choose the best trade-off between miss-classification and false alarms.

4 Evolving Comprehensible Rules

To detect important movements in financial stock prices, our approach is inspired by two previous works: EDDIE Tsang et al. (1998), Li (2001), Tsang et al. (2004), Tsang et al. (2005) and Repository Method Almanza and Tsang (2006b), Almanza and Tsang (2006a). EDDIE is a financial forecasting tool that trains a GP using a set of examples. Every instance in the data set is composed by a set of attributes or independent variables and a *signal* or *desired output*. The independent variables are indicators derived from financial technical analysis. These indicators have been used to identify patterns that can suggest future activity Sharpe et al. (1995). The signal is calculated looking ahead in a future horizon of n units of time, trying to detect an increase or

DRAFT October 12, 2007-16:06

decrease of at least $r\%$. However, when the value of r is very high, which implies an important movement in the stock price, the number of positive cases is extremely small and it becomes very difficult to detect these events. To deal with these special cases, we have proposed a method to discover patterns in financial data sets. This method was designed to detect cases in extreme imbalanced environments.

Our approach, that we have called *Evolving Comprehensible Rules* (ECR), evolves a set of decision rules by using GP and receives feedback from a key element that we called repository. The idea behind using GP in our approach is to being able to represent rules as tree-like structures and so, the GP will create comprehensible rules that the final user could analyse. The resulting rules can be used to create a range of classifications that allows the user to choose the best trade-off between the misclassifications and the false alarms cost. So, our approach has the novelty of offering a range of classifications that best suits the user risk guidelines.

Our approach is composed by the following steps:

1. *Initialisation of population.* Initialise the population using the grow method (Koza 1992b, pages 91-94) (i.e., individuals could have any shape). The objective of this procedure is to create a collection of candidate solutions. We propose to create a population of decision trees using the Discriminator Grammar (DG) Almanza and Tsang (2006b) (Figure 3 depicts the Grammar used in our approach). This grammar[1] produces decision trees that classify or not a single class. By using *DG*, it is simple to get the rules that are embedded in the individuals. Moreover, it is in charge of producing valid individuals. In other words, assuring a valid structure. Figure 2 shows a typical individual created with *DG*.
2. *Extraction of rules.* Once the initial population has been created, before the algorithm starts evaluating, the system decomposes each individual in its corresponding rules and these are evaluated. Given that the system will deal with many rules, we need to define a precision threshold to keep only those rules that have good performance. Let us define a rule $R_k \in T$ as the minimal set of conditions which intersection satisfies the decision tree T. In other words, the tree could contain one or more rules. A decision tree is satisfied when at least one of its rules is satisfied. A rule is satisfied when all its conditions are satisfied. Figure 2 shows a decision tree that holds three rules and as can be seen each of them is able to satisfy the decision tree. To recognise every rule it is necessary to identify the minimal sets of conditions that satisfy the tree T.
3. *Rule simplification.* The aim of rule simplification is to remove noisy conditions (which refers to rules that regardless its values are always satisfied) and redundant conditions. Redundant conditions are those which are repeated or report the same event e.g. $R_1 = \{Var_1 > 0.5 \text{ and } Var_1 > 0.7\}$

[1] The term grammar refers to a representation concerned with the syntactic components and the regulation that specify it Chomsky (1965).

DRAFT October 17, 2006, 7:33pm

where the first condition is redundant. None of them (noisy and redundant conditions) affect the decision of the rule. The simplification of rules is an important process because it allows to recognise the real variables and interactions involved in that rule. Furthermore, it allows to identify the duplication of rules in the repository. This is one of the important elements of our approach because it assures to collect different rules.

4. *Adding new rules in the repository.* The process detects new rules by comparing the new ones with those rules that are stored in the repository. If the rule is totally new, then it is added in the repository. If there is a similar rule in the repository but the new one offers better performance then the old rule is replaced by the new rule. In any other case the rule will be discarded.
5. Once the evolutionary search starts, we create the next population as follows:
   - *Evolving population.* The resulting rules that are stored in the repository are taken as parents and then we apply mutation and use hill-climbing [2] to create the new offsprings.
   - In case there are no enough rules in the repository to create the population, then the remaining individuals are created at random (this is determined by the parameter called "repository random"). This also allows to prevent losing variety. The process is repeated from step 2 until the algorithm has reached the maximum number of generations.
6. Once the evolutionary process has finished, *ECR* is tested by using the corresponding testing data set. It is evaluated by using the rules that are stored in the repository. Those rules are grouped according to their precision in order to classify the cases (i.e., $Precision = \{1, .95, \cdots, .05, 0\}$).

5 Results and Discussion

To conduct our experiments we have used 3 examples based on 2 data sets. These are explained as follows:

Barclays complete - 1,718 cases. The data sets to train and test our approach came from the London stock market from Barclays' stock (from March, 1998 to January, 2005). The attributes of each record are composed by indicators derived from financial technical analysis. Technical analysis is used in financial markets to analyze the stock price behaviour. This is mainly based on historical prices and volume trends Sharpe et al. (1995). The indicators were calculated on the basis of the daily *closing price*[3], volume and some

[2] Hill-climbing is a stochastic technique to find the global optimum where the search is given by trying one step in each direction and then choosing the steepest one (Langdon and Poli (2002)).

[3] The settled price at which a traded instrument is last traded at on a particular trading day.

DRAFT October 12, 2007 - 19:06

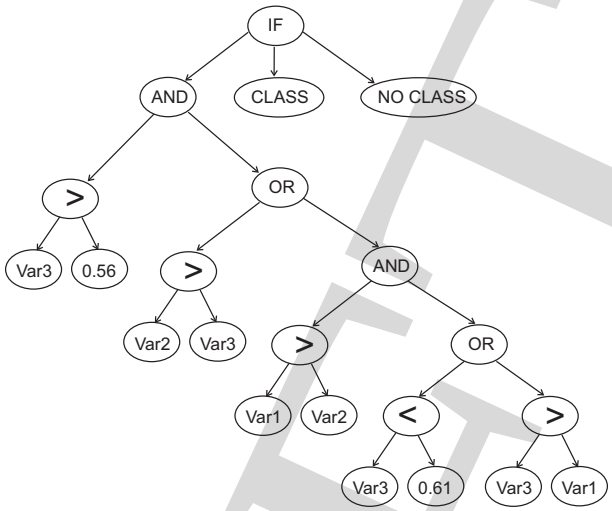

Fig. 2. A typical individual created with Discriminator Grammar. The rules contained within the tree are: $R_1 = \{Var_3 > 0.56 \wedge Var_2 > Var_3\}$, $R_2 = \{Var_3 > 0.56 \wedge Var_1 > Var_2 \wedge Var_3 < 0.61\}$, $R_3 = \{Var_3 > 0.56 \wedge Var_1 > Var_2 \wedge Var_3 > Var_1\}$.

Fig. 3. Discriminator Grammar.

G	→ <Root>
<Root>	→ "If-then-else",<Conjunction> | <Condition>,"Class","No Class"
<Condition>	→ <Operation>, <Variable>, <Threshold> | <Variable>
<Conjunction>	→ "AND"|"OR",<Conjunction>|<Conditional>, <Conjunction>|<Conditional>
<Operation>	→ "<" | ">"
<Variable>	→ var_1 | var_2 | ... var_n
<Threshold>	→ Real number

financial indices as the FTSE[4]. We looked for an increase of 15% in the stock price in an horizon of 10 days. The training data set contains 887 cases (where 39 cases are positive which represents 4.3% of this data set) and the testing data set contains 831 instances (where 24 cases are positive which represents 2.8% of this data set).

Arbitrage - 1,641 cases. This data set consists of 1,000 cases (where 242 cases are positive which represents 24.2% of this data set) for the training data set and the testing data set consists of 641 cases (where 159 cases are positive which represents 24.80% of this data set).

[4] An index of 100 large capitalization companies stock on the London Stock Exchange, also known as "Footsie".

DRAFT October 12 2007 - 16:06

Table 3. Parameters used for the data sets called Barclays complete and variation of Barclays.

Parameters	Values
Population size	1000
Maximum number of generations	25
Hill-Climbing Probability	.14
Maximum number of rules in the repository	2500
Precision threshold	.08
Repository random	.80

Table 4. Parameters used for the data set called Arbitrage.

Parameters	Values
Population size	1000
Maximum number of generations	30
Hill-Climbing Probability	.05
Maximum number of rules in the repository	2000
Precision threshold	.10
Repository random	.80

Variation of Barclays' data set - 400 cases. This data set is a variation of the first data set. It is a sample of 400 cases for each data set (training and testing). In this case, we have 15 positive cases for the training data set (which correspond to 3.7%) and there are 13 positive cases for the testing data set (which correspond to 3.2%).

5.1 Performance analysis

Let us start analysing the results found by our approach, ECR, in the first example (i.e., using the complete Barclays data set). In Figure 4, we show the plot produce by the results found by ECR and in Table 5 we show the parameters used by our approach. Notice how the results produced by our approach are fine balance between the conservative and liberal space. The area under the curve (AUC) plotted when using the results found by our approach is equal to 0.80. Now, let us discuss the points that form the curve. For instance, when the precision threshold is equal 0.50, the recall has a high recall given that the system was able to detect 79% of the positive examples. This result has been found without sacrifying a good accuracy (i.e., 77%).

When the investor's risk-guidelines are conservative, the system is able to offer him/her a suitable classification. For instance, take a look when the precision threshold is 0.70. The system is able to classify up to a quarter of positive cases with a very high accuracy (92%). For values where the precision threshold is greater or equal to 0.40, then the classifier's performance tends to decrease because the number of new positive cases that are detected are based on a serious decrease of accuracy.

DRAFT October 17, 2007, 16:46

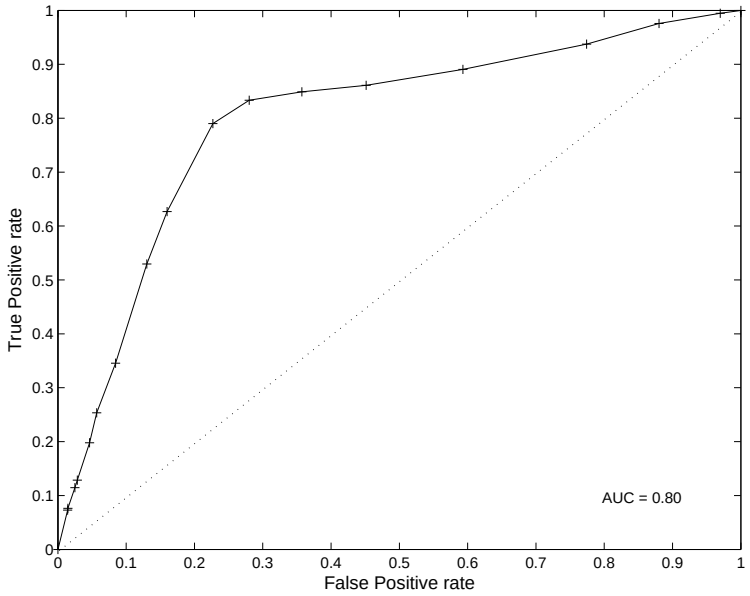

Fig. 4. ROC curve for the Barclays complete data set.

Table 5. GP Summary of Parameters

Parameter	Value
Population size	1,000
Initialisation method	Growth
Generations	100
Crossover Rate	0.8
Mutation Rate	0.05
Selection	Tournament (size 2)
Control bloat growing	50% of trees whose largest branch exceed 6 nodes are penalized with 20% in its fitness
Fitness Function	$\sqrt{Recall \cdot Precision}$

Moreover, we tested the same data set using a standard GP. To obtain meaningful results, we performed a series of 20 runs. Table 5 displays the parameters used by the standard GP. The results found by the GP are: TP=3, FP=68, FN=21, TN=739. Thus, the false positive rate = 0.084 and true positive rate =0.12 which is under the ROC curve produced by the results found by our classifier (see Figure 4). As it can be observed the standard GP classification is conservative despite the fact that its fitness function is the $\sqrt{Recall \cdot Precision}$ which has been used in ML to deal with imbalanced data sets Kubat et al. (1998) and it is able to encourage the classification of positive cases. On the other hand, ECR not just has better performance but it also offers a range of classification to suit the user's risk guidelines.

DRAFT October 26 2007 - 16:06

Fig. 5. Results found by our approach. Precision, recall and accuracy are shown in labels (a), (b) and (c), respectively for the 3 sets of examples.

Precision	Barclays			Arbitrage			Barclays 400		
Threshold	(a)	(b)	(c)	(a)	(b)	(c)	(a)	(b)	(c)
1.00	0.13	0.07	0.96	0.97	0.62	0.90	0.08	0.55	0.78
0.95	0.13	0.07	0.96	0.92	0.70	0.91	0.08	0.55	0.78
0.90	0.13	0.08	0.96	0.84	0.76	0.90	0.08	0.56	0.78
0.85	0.12	0.11	0.95	0.75	0.77	0.88	0.08	0.63	0.76
0.80	0.12	0.13	0.95	0.67	0.82	0.85	0.07	0.71	0.67
0.75	0.11	0.20	0.93	0.58	0.87	0.81	0.06	0.81	0.58
0.70	0.12	0.25	0.92	0.53	0.89	0.77	0.06	0.84	0.54
0.65	0.11	0.35	0.90	0.46	0.95	0.71	0.05	0.89	0.47
0.60	0.11	0.53	0.86	0.42	0.96	0.65	0.04	0.96	0.32
0.55	0.10	0.63	0.83	0.39	0.99	0.61	0.04	0.99	0.22
0.50	0.09	0.79	0.77	0.35	1.00	0.54	0.04	1.00	0.15
0.45	0.08	0.83	0.72	0.32	1.00	0.47	0.04	1.00	0.13
0.40	0.07	0.85	0.65	0.29	1.00	0.39	0.03	1.00	0.10
0.35	0.05	0.86	0.56	0.26	1.00	0.29	0.03	1.00	0.07
0.30	0.04	0.89	0.42	0.25	1.00	0.25	0.03	1.00	0.04
0.25	0.03	0.94	0.25	0.25	1.00	0.25	0.03	1.00	0.03
0.20	0.03	0.98	0.14	0.25	1.00	0.25	0.03	1.00	0.03
0.15	0.03	0.99	0.08	0.25	1.00	0.25	0.03	1.00	0.03
0.10	0.03	0.99	0.06	0.25	1.00	0.25	0.03	1.00	0.03
0.05	0.03	0.99	0.06	0.25	1.00	0.25	0.03	1.00	0.03
0.00	0.03	0.99	0.06	0.25	1.00	0.25	0.03	1.00	0.03

Now, let us focus our attention in the second example (using the Arbitrage data sets - which is an imbalanced data set but less imbalanced than the one used in our previous example). When the precision threshold = 0.95, the system is able to detect 70% of the positive examples with an impressive precision of 92% and an accuracy of 91%. As can be seen from Table 5, almost the totality (96%) of the positive cases is achieved using a precision threshold of 0.60 with an accuracy of 65%. At this point, the performance of the classifier becomes liberal.

However, it is fair to say that the approach proposed by Tsang et al. (2005), called EDDIE-ARB, achieved a precision of 100% and a recall of 42%. Nevertheless, our approach (ECR) achieved a precision of 96% with the same recall reported by Tsang and co-workers. The advantage of our approach is that is able to produce a range of classifications that allows the user to detect more positive cases according to his requirements. This comes at a cost that the user could deal with more false alarms (i.e., losing precision).

Finally, let us discuss the third example used in this work. For analysis purposes we have used C.5 Quinlan (1993) to compare our proposed approach (ECR).

DRAFT October 24, 2007, 10:06

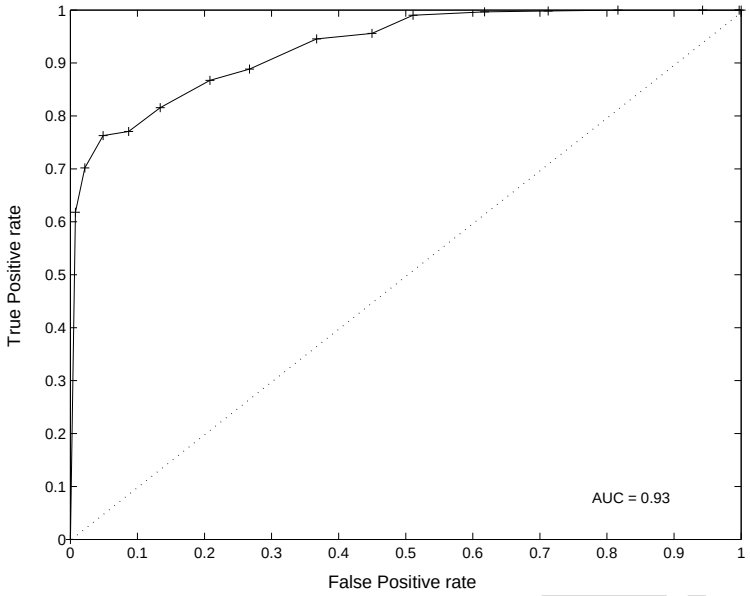

Fig. 6. ROC curve for the Arbitrage data set.

To perform the experiment we used the trial C.5 version. This demonstration version, however, cannot process more than 400 training or testing cases. For such reason, the size of the data sets had to be adjusted to meet this requirement. The new data sets hold 400 records each of them and these are conformed as follows: training data contains 385 negative cases and 15 positive examples and the testing data holds 387 negative cases and 13 positive cases. We performed a series of 20 runs using the new data sets. On the other hand, Quinlan's algorithm was tested using ten-fold cross-validation and standard parameters settings.

The result obtained by C.5 is the following: *True positive* = 0, *false positive*=0, *false negatives* = 13, *true negative* = 187. As it can be seen C.5 has an excellent accuracy (96.7%), however, it fails to detect the positive cases. In contrast ECR was able to detect about 63% of positive cases using a precision threshold = 0.85 with an accuracy of 76%. The remaining results obtained by ECR are plotted in Figure 7, which shows that AUC = 0.75.

6 Conclusions

In this work, we have presented a new approach called *Evolving Comprehensible Rules* (ECR) to classify imbalanced classes. For analysis purposes, we have

DRAFT October 12, 2007-16:06

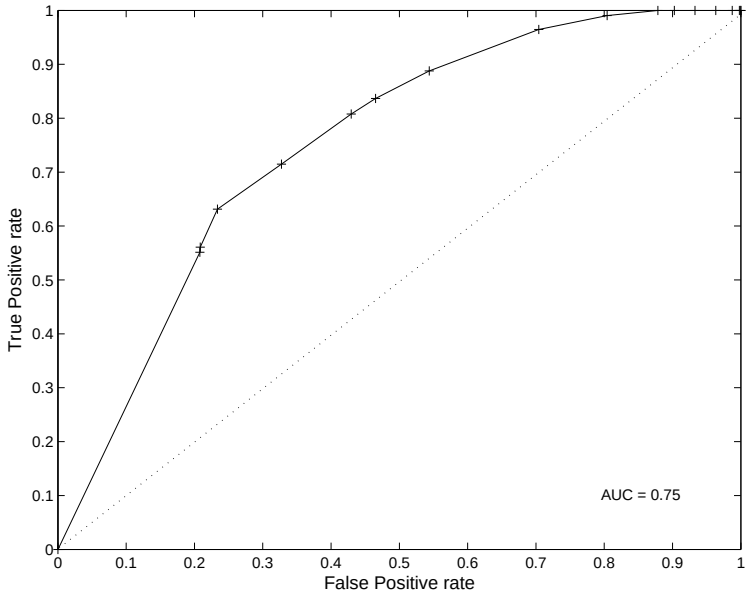

Fig. 7. ROC curve for the Barclays data set - 400 cases.

used 2 data sets and conducted 3 different experiments. Our approach was applied to the data sets in order to discover patterns to classify rare events. The system's output is a set of decision rules which, by varying a threshold, is able to produce a range of classifications to suit different investor's risk preferences.

The core of our approach is based on GP, which is aided by a repository of rules. The aim of this repository is to collect useful patterns that could be used to produce the following population in the evolutionary process. The main operator of this approach is based on subtree mutation, which produces instances of the collected patterns.

From experimental results, it has been showed that our approach is able to improve the recall in the classification. In other words, it has been able to pick up more positive cases in imbalanced environments. For a detailed analysis, we have plotted the range of classifications in Receiver Operating Characteristic (ROC). This helped us to visualise how the points in the curve are well-distributed. The applicability of such tool can be translated that the investor could choose among different risk strategies.

Following the same idea, we have used Area Under the ROC Curve (AUC) to measure the general performance of our approach (ECR). Finally, to complement our analysis, we have used the standard GP for the first example, a specialized GP proposed by Tsang et al. (2005) for the second example and C.5 for the remaining example to compare with the results found by ECR. For the first example (which is an extremely imbalanced data set), our approach outperformed the traditional GP. For the second example, which is less im-

Draft October 4 2007-16:06

balanced than the former data set, our approach did not outperform Tsang's approach. However, our approach was able to provide a full-range of classifications to meet different users' requirements. Finally, for the last example, we have compared the performance of C.5 with the performance of our approach. ECR was able to classify more positive cases than C.5 in extreme imbalanced data sets. These results support the claim that ECR is effective and practical for classification in imbalanced data sets.

Acknowledgments

The first and the third authors thank to CONACyT for support to pursue graduate studies at University of Essex. The authors would like to thank the anonymous reviewers for their valuable comments.

References

Almanza, A. L. G. and Tsang, E. P.: 2006a, Forecasting stock prices using genetic programming and chance discovery, *12th International Conference On Computing In Economics And Finance*.

Almanza, A. L. G. and Tsang, E. P.: 2006b, The repository method for chance discovery in financial forecasting, *in* Springer-Verlag (ed.), *To appear in KES2006 10th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*.

Batista, G. E. A. P. A., Prati, R. C. and Monard, M. C.: 2004, A study of the behavior of several methods for balancing machine learning training data, *SIGKDD Explor. Newsl.* **6**(1), 20–29.

Bobbin, J. and Yao, X.: 1999, Automatic Discovery of Relational Information in Comprehensible Control Rules by Evolutionary Algorithms, *Proceedings of the Third Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, Canberra, Australia, pp. 117–123.

Bojarczuk, C. C., Lopes, H. S. and Freitas, A. A.: 1999, Discovering comprehensible classification rules by using genetic programming: a case study in a medical domain, *in* W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela and R. E. Smith (eds), *Proceedings of the Genetic and Evolutionary Computation Conference*, Vol. 2, Morgan Kaufmann, Orlando, Florida, USA, pp. 953–958.

Chen, S.-H. (ed.): 2002, *Genetic Algorithms and Genetic Programming in Computational Finance*, Kluwer Academic.

Chen, S.-H. and Wang, P. P. (eds): 2004, *Computational intelligence in economics and finance*, Springer.

Chomsky, N.: 1965, *Aspects of the theory of syntax*, Cambridge M.I.T. Press.

DRAFT October 12 2007 - 16:06

Corcoran, A. L. and Sen, S.: 1994, Using Real-Valued Genetic Algorithms to Evolve Rule Sets for Classification, *International Conference on Evolutionary Computation*, pp. 120–124.

Falco, I. D., Cioppa, A. D. and Tarantino, E.: 2001, Discovering interesting classification rules with genetic programming, *Applied Soft Computing* **1**(4), 257–269.

Fawcett, T.: 2004, Roc graphs: Notes and practical considerations for researchers, *Introductory paper*.

Fidelis, M. V., Lopes, H. S. and Freitas, A. A.: 2000, Discovering comprehensible classification rules a genetic algorithm, *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, IEEE Press, La Jolla Marriott Hotel La Jolla, California, USA, pp. 805–810.

Hanley, J. A. and McNeil, B. J.: 1998, The meaning and use of the area under a reciever operating characteristic roc curve, *Radiology*, Vol. 143, W. Madison, pp. 29–36.

Huang, J. and Ling, C. X.: 2005, Using auc and accuracy in evaluating learning algorithms, *IEEE Transactions on Knowledge and Data Engineering* **17**(3), 299–310.

Janikow, C. Z.: 1993, A Knowledge-Intensive Genetic Algorithm for Supervised Learning, *Machine Learning* **13**(1-3), 189–228.

Japkowicz, N.: 2000, The class imbalance problem: Significance and strategies, *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000)*, Vol. 1, pp. 111–117.

Jong, K. A. D. and Spears, W. M.: 1991, Learning Concept Classification Rules using Genetic Algorithms, *Proceedings of the Twelfth International Conference on Artificial Intelligence IJCAI-91*, Vol. 2.

Kohavi, R. and Provost, F.: 1998, Glossary of terms, *Edited for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, Vol. 30.

Koza, J.: 1992a, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, Cambridge, Massachusetts.

Koza, J. R.: 1992b, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, Cambridge, Massachusetts.

Kubat, M., Holte, R. C. and Matwin, S.: 1998, Machine learning for the detection of oil spills in satellite radar images, *Machine Learning*, Vol. 30, 195-215.

Kwedlo, W. and Kretowski, M.: 1999, An Evolutionary Algorithm Using Multivariate Discretisation for Decision Rule Induction, *Principles of Data Mining and Knowledge Discovery*, pp. 392–397.

Langdon, W. B. and Poli, R.: 2002, *Foundations of Genetic Programming*, Springer Verlag, Berlin, Germany.

Li, J.: 2001, *A genetic programming based tool for financial forecasting*, PhD Thesis, University of Essex, Colchester CO4 3SQ, UK.

McCarthy, K., Zabar, B. and Weiss, G.: 2005, Does cost-sensitive learning beat sampling for classifying rare classes?, *UBDM '05: Proceedings of the*

DRAFT October 19, 2016

1st international workshop on Utility-based data mining, ACM Press, New York, NY, USA, pp. 69–77.

Mitchell, T. M.: 1997, *Machine Learning*, McGraw-Hill, Boston, Mass.

Niimi, A. and Tazaki, E.: 2000, Rule discovery technique using genetic programming combined with apriori algorithm, *in* S. Arikawa and S. Morishita (eds), *Proceedings of the Third International Conference on Discovery Science*, Vol. 1967 of *Lecture Notes in Computer Science*, Springer.

Quinlan, J. R.: 1993, *C.45 Programs for Machine Learning*, Morgan Kaufmann, San Mateo California.

Sharpe, W. F., Alexander, G. J. and Bailey, J. V.: 1995, *Investments*, Prentice-Hall International, Inc, Upper Saddle River, New Jersey 07458.

Tsang, E. P., Li, J. and Butler, J.: 1998, Eddie beats the bookies, *International Journal of Software, Practice and Experience*, Vol. 28 of *10*, Wiley, pp. 1033–1043.

Tsang, E. P., Markose, S. and Er, H.: 2005, Chance discovery in stock index option and future arbitrage, *New Mathematics and Natural Computation, World Scientific*, Vol. 1 of *3*, pp. 435–447.

Tsang, E. P. and Martinez-Jaramillo, S.: 2004, Computational finance, *IEEE Computational Intelligence Society Newsletter*, IEEE, pp. 3–8.

Tsang, E. P., Yung, P. and Li, J.: 2004, Eddie-automation, a decision support tool for financial forecasting, *Journal of Decision Support Systems, Special Issue on Data Mining for Financial Decision Making*, Vol. 37 of *4*.

Vanagas, G.: 2004, Receiver operating characteristic curves and comparison of cardiac surgery risk stratification systems, *Interact CardioVasc Thorac Surg* **3**(2), 319–322.

Weiss, G. M.: 2004, Mining with rarity: A unifying framework, *Special issue on learning from imbalanced datasets*, Vol. 6, pp. 7–19.

DRAFT
October 12, 2007 - 16:16