International Journal of Management Science
Vol 15, No 2, November 2009

An Evolutionary Approach to Inferring Decision Rules from Stock Price Index Predictions of Experts

Myoung-Jong Kim*

Division of Business, Dongseo University,
San 69-1, Churye-2Dong, Sasang-Gu, Busan, 617-716, Korea,

(Received: July 15, 2009 / Revised: September 8, 2009 / Accepted: September 18, 2009)

ABSTRACT

In quantitative contexts, data mining is widely applied to the prediction of stock prices from financial time-series. However, few studies have examined the potential of data mining for shedding light on the qualitative problem-solving knowledge of experts who make stock price predictions. This paper presents a GA-based data mining approach to characterizing the qualitative knowledge of such experts, based on their observed predictions. This study is the first of its kind in the GA literature. The results indicate that this approach generates rules with higher accuracy and greater coverage than inductive learning methods or neural networks. They also indicate considerable agreement between the GA method and expert problem-solving approaches. Therefore, the proposed method offers a suitable tool for eliciting and representing expert decision rules, and thus constitutes an effective means of predicting the stock price index.

Keywords: Data Mining, Genetic Algorithms, Qualitative Decision Rules, Stock Price Index

1. Introduction

Data mining is commonly applied in various business domains, such as marketing, finance, banking, manufacturing, and telecommunications (Brachman *et al.* [5]. In relation to this, the prediction of the stock price index is a particularly important problem in the realm of business. Numerous studies on the prediction of the stock price index employ one of three main approaches, two of which are quantitative and one of which is qualitative. First, numerous studies develop appropriate prediction models

* Corresponding author, E- mail: mjongkim@gdsu.dongseo.ac.kr

by applying various data mining techniques, such as ARIMA, neural networks, and genetic algorithms (GAs) (Barr and Mani [3]; Kim and Han [14]; Pankratz [19]; Tsaih *et al.* [23]; Walczak [24]). Second, some researchers extract useful predictive rules automatically from a huge amount of financial time-series data. In this instance also, data mining techniques, such as inductive learning methods, neural networks, and GAs, are applied (Bauer [4]; Giles [7]). The third approach involves the construction of qualitative prediction models from the problem-solving knowledge of experts. Interactive techniques such as interviews are sometimes used to investigate the expert knowledge frameworks associated with their prediction of the stock price index (Kuo *et al.* [15]). However, the knowledge acquisition and verification processes in this respect are difficult and time-consuming. Although data mining offers an alternative approach with which to deal with such problems, few studies have proposed data mining as a means of discovering subjective knowledge frameworks from expert qualitative predictions. The shortage of existing research in this area may be due to the difficulties inherent in collecting qualitative information and in resolving the inconsistencies present in the subjective knowledge of various experts.

This paper proposes a GA-based data mining method that allows the extraction of decision rules from qualitative expert predictions of the Korean stock price index (KOSPI). The use of GAs in this study is novel in that existing studies do not make use of these approaches for the purpose of discovering the expert problem-solving knowledge associated with stock price index predictions. We derive results using two alternative data mining techniques, namely, neural networks and inductive learning methods, and compared these results with those from the GA method as a means of assessing its performance. The results of the experiment reveal that the GA method performs significantly better than neural networks and inductive learning methods in terms of predictive accuracy and coverage. They also indicate reasonable agreement between the GA and expert knowledge. Therefore, the proposed GA method offers a suitable tool for eliciting and representing the problem-solving knowledge of experts.

The remainder of this paper is organized as follows. Section 2 presents a review and a comparison of the three kinds of data mining technique used in this paper. Section 3 proposes an GA-based data mining method for extracting subjective expert knowledge, while section 4 discusses the experimental design and the results of the experiments. Concluding remarks and further research issues are described in section 5.

2. Data Mining Techniques for Extracting Expert Decision Rules

The complexity of a decision making process depends, to a large extent, on the nature of the underlying decision problem. Decision making processes employed in the context of well-structured decision problems tend to be minimally subjective, and such processes can be elicited and represented with relative ease. On the other hand, decision making processes associated with ill-structured or semi-structured problems tend to be highly subjective, and decision-makers generally employ their intuition and experience when making decisions in such situations.

Stock experts use their subjective knowledge for predicting the stock market in order to deal with a variety of factors such as macroeconomics, stock market, substitutive goods, politics, sociological and psychotic factors. From the interview with stock experts, the factors are identified into the four categories of the qualitative information including economic prospects (EP), the levels of stock supply and demand (SSD), the amount of currency that can be used to buy stocks (AOC), and any conditions that are favorable or unfavorable for the stock market trend (CFU). The EP refers to the economic forecast, which is determined by the composite effects of exports, GNP, inflation, and so on. Those factors potentially affecting the SSD include the capital-increases of listed firms, new stock supplies, and the investment activities of institutions. The AOC is determined by the bond yield, the call rate of interest, the amount of cash deposited with stockbrokers, and the monetary policy of the government. The concept of CFU is relatively broad; for example, the CFU can include any political situation, domestic or international, that may affect the stock market trend. Moreover, news coverage of the stock market could influence investor decisions to buy or sell the corresponding stocks. Therefore, CFU covers a wide range of macro and micro factors. Experts analyze these qualitative factors subjectively and then predict the next trend of the stock market.

Data mining techniques used in the extraction of quantitative rules can also be applied to infer the subjective decision rules of experts from their predictions. Such techniques include inductive learning methods, neural networks, and GAs. Inductive learning methods are typical rule extraction techniques that operate via a successive partitioning of cases until all subsets belong to a single class (Quinlan [21, 22]). Several studies have employed inductive learning methods in predicting the stock price index (Tsaih *et al.* [23]).

Neural networks are also sometimes employed to extract rules for solving crisp and fuzzy classification problems (Giles *et al.* [7], Hayashi and Imura [9], Lin and Lee [16]). Evidence indicates that neural networks are suitable for building a logic system with a relatively small number of numerical variables. However, neural networks lack analytical guidance in determining the network configuration. They may also be trapped at local optima during the learning process. These problems place limitations on the quality of rules that can be generated by neural networks.

A technique more recently applied to prediction problems comprises GAs, which are heuristic search techniques based on the theory of natural selection and evolution (Holland [10]). GAs are used in the task of rule extraction under propositional and first-order logic (Anglano *et al.* [1], Augier *et al.* [2], Giordana *et al.* [8], Noda *et al.* [18]). GA-based methods are also used for choosing appropriate sets of fuzzy "if-then" rules for classification problems (Ishibuchi *et al.* [11], Peña and Sipper [20]). Hybrid classification learning systems involve a combination of GAs and neural networks (Yao and Liu [25]), or a combination of GAs and linear discrimination models. GAs are also successfully used to obtain rules for predicting stock prices from financial time-series data (Bauer [4]).

3. Methodology

In this section, we propose an GA-based data mining approach to extracting fuzzy rules from qualitative expert decisions.

3.1 Initialization of the Population

Traditional GAs use an initial population consisting of chromosomes randomly distributed by the system, while the initial population of the proposed GA method consists of fuzzy rules that are converted directly from expert prediction cases. This method provides a better starting point for reproduction. We use a binary string to represent each rule or case. A rule is coded as one chromosome, which consists of several segments. Each segment corresponds either to an attribute in the condition part of the rule or to a class in the conclusion part of the rule. Each segment consists of a string of genes that take a binary value of either 0 or 1. Each gene corresponds to

one discrete linguistic term of the attribute or class.

Assume that we have a prediction (i.e., case) made by expert u, as shown in Figure 1. Each black circle in the qualitative factor table indicates the level of each factor assigned by the expert, while a black circle in the class table indicates the actual class for the same case.

Qualitative Factor	Negative	Average	Positive
EP	○	○	•
SSD	○	•	○
AOC	○	○	•
CFU	○	•	○

Genetic Coding
(001)
(010)
(001)
(010)

Class	Upward	Downward
Actual Class	•	○

;
(10)

Figure 1. An example of rule conversion from a case

We use the following binary string to represent case u by converting each black circle into the binary value 1 and each white circle into the binary value 0:

$$u = [\ (001)\ (010)\ (001)\ (010);\ (10)\ ],$$

where the parentheses separate segments and the semicolon separates the IF part of the rule from the THEN part of the rule. This binary string can be read as follows: "IF EP is positive, SSD is average, AOC is positive, and CFU is average, THEN Upward."

This genetic coding scheme can also represent cases having an OR relation. As the initial population evolves from one generation to the next, new children are born. These new children, such as 110 and 111, represent the OR relation for an attribute. A binary string 110 represents the OR relation between two linguistic terms and can be interpreted as the following condition for an attribute: "negative or average." The all-unitary string 111 represents the OR relation among all of the linguistic terms in an attribute. This string includes all possible cases for the attribute; that is, it is unconditional, such that the corresponding attribute is not involved in the condition part of the rule. However, the all-zero string 000 is not allowed in our coding scheme, because each attribute must take at least one term.

3.2 Fitness Evaluation of Rules

The role of the fitness function is to characterize numerically the performance of the rule considered. In this study, we aimed to use the GA method to find those rules that are most accurate and general among all of the rules of a population. Thus, our GA method employs a composite fitness function for optimal accuracy and coverage. To measure the accuracy and coverage of each rule, we use the following definitions, which are adapted from the pattern matching definitions of Yuan and Zhuang [26]:

Definition 1: The condition match of rule r and case u is defined as

$$mA(r,\ u) = \begin{cases} 1 \text{ - matched} & \text{if } r(A_{kj}) \geq u(A_{kj}) \text{ for all } k \text{ and } j \\ 0 \text{ - mismatched} & \text{otherwise,} \end{cases} \tag{1}$$

where A_{kj} denotes the jth linguistic term of the kth attribute. In this study, j represents one of three levels, namely Positive, Average, and Negative, while k corresponds to one of the four qualitative factors in Table 1. This definition specifies that rule r is applicable to case u and that $mA(r,\ u) = 1$, if all linguistic values in the IF part of rule r are greater than or equal to those of case u. Otherwise, the conditions of the rule and the case are mismatched, and $mA(r,\ u) = 0$.

Definition 2: The conclusion match of rule r and case u is defined as

$$mC(r,\ u) = \begin{cases} 1 \text{ - matched} & \text{if } r(C_i) = u(C_i) \text{ for all } i, \\ 0 \text{ - mismatched} & \text{otherwise.} \end{cases} \tag{2}$$

where i denotes the i-th class. The conclusions of the rule and the case are matched and $mC(r,\ u) = 1$, if the rule and the case have the same class. Otherwise, $mC(r,\ u) = 0$.

Definition 3: The rule match between rule r and case u is defined as

$$mR(r,\ u) = \begin{cases} 1 \text{ - matched} & \text{if } mA(r,\ u) = mC(r,\ u) = 1, \\ 0 \text{ - mismatched} & \text{otherwise.} \end{cases} \tag{3}$$

This definition specifies that the case is accurately predicted by the rule and that $mR(r, u) = 1$, if the IF part of rule r is applicable to case u and has the same conclusion as the case. Otherwise, $mR(r, u) = 0$.

For example, assume that we obtain a rule $r1$ and a case $u1$, as follows:

$$r1 = [\ (011)\ (100)\ (111)\ (001)\ ;\ (10)\ ]$$
and
$$u1 = [\ (001)\ (100)\ (001)\ (001)\ ;\ (10)\ ].$$

For rule $r1$ and case $u1$, the condition match is $mA(r1, u1) = 1$, because rule $r1$ contains an OR operation that covers case $u1$; that is, all of the linguistic values of rule $r1$ are greater than or equal to those of case $u1$. Similarly, the conclusion match is $mC(r1, u1) = 1$, because rule $r1$ and case $u1$ have the same conclusion; thus, the rule match is $mR(r1, u1) = 1$, because the condition and conclusion matches each take on the value 1.

Definition 4: The coverage of a rule indicates the extent to which its condition part is applicable to all cases. Therefore, the coverage is the proportion of cases used in learning to which a rule r can be applied. It can be defined as

$$COV(r) = \frac{\sum(mA(r, u))}{n} \tag{4}$$

where n is the total number of cases. The larger a rule's coverage, the more general is that rule.

Definition 5: The predictive accuracy of rule r, which corresponds to the quality of that rule, is defined as

$$PA(r) = \frac{\sum mR(r, u)}{\sum mA(r, u)}$$

$PA(r)$ is the ratio of the accurately predicted cases to the cases to which rule r is applicable.

Definition 6: The fitness function of rule r is defined as

$$\text{Objective} : Maximize\,(PA(r)+Cov(r))$$
$$\text{Subject to}: PA(r) \geq \alpha \text{ and } Cov(r) \geq \beta$$

The objective of the fitness function is defined as a composite measure of accuracy and coverage. The constraints of the fitness function control the generation of useless rules that have low quality or small coverage, and α and β are predefined by the user. This fitness function provides an effective selection criterion that balances the accuracy and generality of the rules selected.

3.3 Genetic Operators

Selection is a process by which rules with high fitness value are chosen as parents for the purpose of reproduction. The mating selection of the proposed GA method is restricted to a single species; that is, the parents selected for reproduction are selected from among rules having the same class, because the genetic operation between two rules with different classes tends to generate low-performance offspring (Holland 1975).

Crossover is a GA process by which two parent chromosomes exchange information to generate two child chromosomes, and mutation is an optimization process that occurs via occasional alternation. These processes are performed at the bit level in traditional GAs, while those of the proposed GA method are performed segment by segment, rather than bit by bit, because each segment of a rule has a special meaning. This choice reduces the possibility of generating useless rules. Figures 2 and 3 illustrate examples of crossover and mutation. Two parent chromosomes can generate two child chromosomes through a crossover of their second and fourth segments, as shown in Figure 2. The mutation on the second segment of the chromosome may generate one of six possible new chromosomes, as shown in Figure 3.

The semantic meanings of the GA strings are as follows:

Parent 1: IF EP is negative, SD is average, AOC is average, and CFU is average, THEN Upward.

Parent 2: IF EP is negative, SSD is negative, AOC is negative, and CFU is positive, THEN Upward.

Child 1: IF EP is negative, SSD is negative, AOC is average, and CFU is positive, THEN Upward.

Child 2: IF EP is negative, SSD is average, AOC is negative, and CFU is average, THEN Upward.

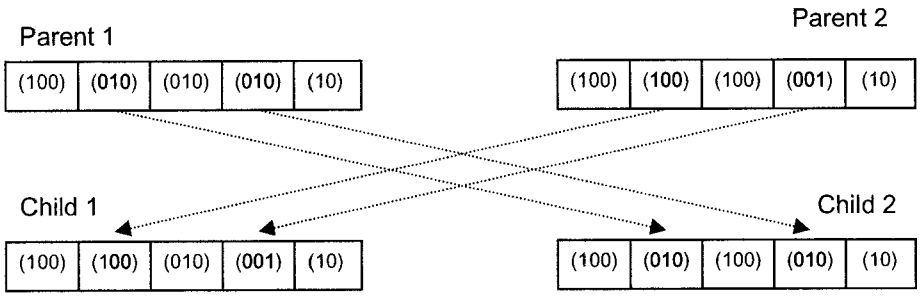

Figure 2. An example of crossover

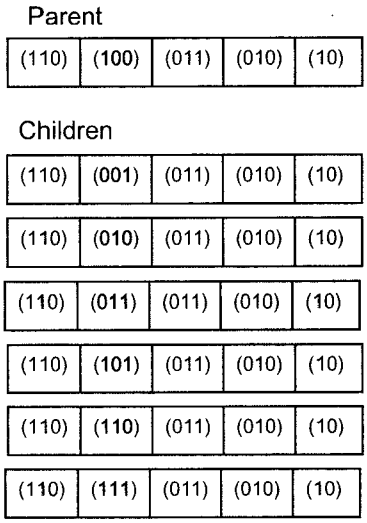

Figure 3. An example of mutation

Replacement is performed between the rules with the same conclusion, because the replacement between two rules with different classes tends to generate low-quality rules. After replacement, the new population is evaluated based on the fitness function. This process continues iteratively until a stopping condition is satisfied.

4. Experimental Evaluation

4.1 Data and Experimental Design

The sample was collected during the period 2001~2006 and consists of a set of 312 weekly KOSPI observations. The experts who evaluated four qualitative factors and predicted future trends in the stock price index are currently experienced stock analysts working for one of the largest securities companies in Korea. We use professional experiences as the standards of selecting the experts. Comparing the average of professional experience of stock experts in Korea which is about 5~6 years, the experts involved in this study have above 6 years' experiences and 9 years' experiences on average in which the result of their evaluation can be regarded highly reliable. They evaluated four qualitative factors and assigned an appropriate level to each of these four factors, as shown in Figure 1.

The data set for the GA may be split into two subsets, a training set and a validation set, each of which comprises 50% of the data. The control constraint parameters, α and β, are to be determined according to the characteristics of decision problems. Having several experiments by varying α and β, we ended up with the result whose parameters, α and β, are set to 0.7 and 0.2, respectively. The crossover rate ranges from 0.5 to 0.7 in the sample, and the mutation rate similarly ranges from 0.06 to 0.12. The set of individuals has evolved over 1000 generations.

To apply inductive learning methods, we assign values to three levels of each factor: negative (1), average (2), and positive (3). The CHAID algorithm (Kass [12]), which facilitates a multi-way split based on a chi-square test, is applied to generate the induction rules from the same training data set as that used for the GA.

The two stages of the learning process generate rules from neural networks. In the first stage, 3-layered backpropagation neural networks learn classification functions from the training cases. The data set for the backpropagation neural networks is split into three subsets: a training set, test set, and validation set that comprise 30%, 20% and 50% of the data, respectively. In the second stage, the CHAID algorithm generates rules from the trained neural networks. The software package SAS Enterprise Miner 4.0 is used in the context of both the inductive learning methods and the neural networks.

4.2 Experimental Results

Finally, the genetic evolution process extracts eight prediction rules, five of which are upward rules. The rest are downward rules. These rules and their corresponding descriptions are illustrated in Tables 1 and 2. Moreover, their performance is stable for both the training and test data, in terms of both coverage and accuracy, as shown in Table 1.

Table 1. Rules generated by the genetic evolutionary process

No.	Rule					Training (156 cases)			Validation (156 Cases)		
	EP	SSD	AOC	CFU	Class	Coverage	Accuracy	Fitness	Coverage	Accuracy	Fitness
1	(011)	(001)	(011)	(111)	(10)	0.250	0.897	1.147	0.244	0.895	1.139
2	(100)	(110)	(100)	(110)	(01)	0.256	0.825	1.081	0.250	0.821	1.071
3	(111)	(011)	(001)	(111)	(10)	0.263	0.780	1.043	0.269	0.762	1.031
4	(001)	(111)	(001)	(011)	(10)	0.212	0.758	0.970	0.205	0.750	0.955
5	(111)	(011)	(011)	(001)	(10)	0.237	0.730	0.967	0.231	0.722	0.953
6	(001)	(011)	(111)	(011)	(10)	0.231	0.730	0.961	0.224	0.714	0.938
7	(111)	(100)	(110)	(100)	(01)	0.244	0.711	0.955	0.237	0.703	0.940
8	(111)	(100)	(100)	(110)	(01)	0.218	0.706	0.924	0.212	0.697	0.909
Average						0.239	0.767	1.006	0.234	0.758	0.992

Table 2. Descriptions of the rules generated by the GA

Rule	Descriptions
Rule 1	IF EP is average or positive, SSD is positive, and AOC is average or positive, THEN Upward.
Rule 2	IF EP is negative, SSD is negative or average, AOC is negative, and CFU is negative or average, THEN Downward.
Rule 3	IF SSD is average or positive and AOC is positive, THEN Upward.
Rule 4	IF EP is positive, AOC is positive, and CFU is average or positive, THEN Upward.
Rule 5	IF SSD is average or positive, AOC is average or positive, and CFU is positive, THEN Upward.
Rule 6	IF EP is positive, SSD is average or positive, and CFU is average or positive, THEN Upward.
Rule 7	IF SSD is negative, AOC is negative or average, and CFU is negative, THEN Downward.
Rule 8	IF SSD is negative, AOC is negative, and CFU is negative or average, THEN Downward.

The rules generated from the inductive learning methods and the neural networks are illustrated in Tables 3 and 4, respectively. Twelve rules are generated via inductive learning methods, seven of which are upward rules, as shown in Table 3. Nine rules are generated by the neural networks; these are listed in Table 4. Five of these rules are upward rules, while the rest are downward rules.

Table 3. Descriptions of the rules generated by the inductive learning methods

Rule	Descriptions
Rule 1	IF SSD is positive and AOC is average or positive, THEN Upward.
Rule 2	IF SSD is positive, AOC is negative, and CFU is average or positive, THEN Upward.
Rule 3	IF SSD is positive, AOC is negative, and CFU is negative, THEN Downward.
Rule 4	IF SSD is average and AOC is positive, THEN Upward.
Rule 5	IF SSD is average, AOC is average, and EP is average or positive, THEN Upward.
Rule 6	IF SSD is average, AOC is average, EP is negative, and CFU is average or positive, THEN Upward.
Rule 7	IF SSD is average, AOC is average, EP is negative, and CFU is negative, THEN Downward.
Rule 8	IF SSD is average, AOC is negative, and EP is average or positive, THEN Upward.
Rule 9	IF SSD is average, AOC is negative, and EP is negative, THEN Downward.
Rule 10	IF SSD is negative, AOC is average, and CFU is average or positive, THEN Upward.
Rule 11	IF SSD is negative, AOC is average, and CFU is negative, THEN Downward.
Rule 12	IF SSD is negative and AOC is negative, THEN Downward.

Table 4. Descriptions of the rules generated by the neural networks

Rule	Descriptions
Rule 1	IF SSD is positive and AOC is average or positive, THEN Upward.
Rule 2	IF SSD is positive, AOC is negative, and CFU is average or positive, THEN Upward.
Rule 3	IF SSD is positive, AOC is negative, and CFU is negative, THEN Downward.
Rule 4	IF SSD is average and EP is average or positive, THEN Upward.
Rule 5	IF SSD is average, EP is negative, and AOC is average or positive, THEN Upward.
Rule 6	IF SSD is average, EP is negative and AOC is negative, THEN Down Downward.
Rule 7	IF SSD is negative, AOC is average, and CFU is average or positive, THEN Upward.
Rule 8	IF SSD is negative, AOC is average, and CFU is negative, THEN Downward.
Rule 9	IF SSD is negative and AOC is negative, THEN Downward.

When the rules are applied to the cases, two or more rules can be applied to predict the same case. We use the following steps to deal with such a situation. First, the case is assigned the class of its corresponding rule only if a single rule is applied to the case. Second, the case is assigned the class of the rule with the highest accuracy if two or more rules are applied to the case at the same time. Finally, the case is assigned the class of the rule with the largest coverage if two or more rules with the same accuracy are applied to classify the case at the same time.

Table 5 compares the performance of the three extraction methods with regard to the validation data set. Table 5 indicates that the GA method generates rules with

higher accuracy and greater coverage than do the other techniques, and that the structure of those rules is more compact. The concept of overall predictive accuracy refers to the accuracy level when rules are applied to cases according to the application steps. The GA method also exhibits better predictive accuracy than do neural networks and inductive learning methods.

Table 5. Performance of the data mining techniques (156 cases)

Data mining Techniques	The number of rules extracted	Average Coverage	Average Accuracy	Overall Accuracy
GA	8	23.9%	75.8%	79.5%
Inductive learning	13	19.8%	72.7%	75.6%
Neural networks	9	22.7%	73.4%	76.3%

We use the Wilcoxon matched-pairs, signed-ranks test to examine whether the predictive accuracy of the rules generated by each of the three data mining techniques differs significantly across methods. Table 6 presents the results of the test. The results indicate that the rules derived using the GA method are significantly better than those obtained using the other data mining techniques. However, the predictive accuracy of the rules generated via inductive learning methods and neural networks does differ significantly.

Table 6. Results of Wilcoxon's matched-pairs, signed-ranks test

	GA	Inductive learning	Neural Networks
GA	-		
Inductive learning	3.312*	-	
Neural networks	3.112*	1.346	

Note: * significant at the 1% level.

It is important to measure the agreement between the expert predictions and each predictor, because such a measurement indicates the degree to which the subjectivity of the experts has been incorporated into the model. We adopt Cohen's *kappa* (Cohen [6]) as the measure of agreement. Cohen's *kappa* measures the agreement between two predictors (e.g., an expert and each of the data mining techniques) in classifying the same set of cases. Cohen's *kappa* is defined as the ratio of the percentage of

agreement minus the chance agreement to the largest possible non-chance agreement. Thus, this measure takes into account the fact that predictions may coincide merely by chance. The likelihood of such an agreement actually depends upon the percentage of matches in each class, and it decreases as the number of classes increases. Using the above definition, a *kappa* value of 1 indicates perfect agreement, while a *kappa* value of 0 indicates that the agreement is no better than what would have been obtained by chance.

Table 7 presents the upward (UW) and downward (DW) trends predicted by the experts, paired with each of the data mining techniques. The value in each cell indicates the number of cases that result from each combination, and the value in parentheses indicates the fraction of total cases that each combination represents. For example, 70 cases are predicted to be upward by both the experts and the GA method, and cases in this cell account for 44.9% (70/156) of the total. From Table 7, we can compute the chance of agreement between the experts and the GA method as 49.98% (48.7% × 50.6% + 51.3% × 49.4%). Therefore, the value of *kappa* is (79.5 – 49.98)/(100 - 50.0) = 0.5803. Thus, the GA method is more in agreement with expert problem-solving knowledge than are the inductive learning methods or neural networks. Therefore, the GA method is an effective means of inferring the subjective knowledge of experts from their qualitative predictions.

Table 7. Results of the kappa tests of the expert predictions and the data mining techniques

Experts	GA			Inductive learning			Neural networks		
	UW	DW	Total	UW	DW	Total	UW	DW	Total
UW	70	6	76	65	13	78	62	14	76
	(44.9)	(3.8)	(48.7)	(41.7)	(8.3)	(50.0)	(39.7)	(9.0)	(48.7)
DW	9	71	80	14	64	78	16	64	80
	(5.8)	(45.5)	(51.3)	(9.0)	(41.0)	(50.0)	(10.3)	(41.0)	(51.3)
Total	79	77	156	79	81	156	78	78	156
	(50.6)	(49.4)	(100.0)	(50.6)	(49.4)	(100.0)	(50.0)	(50.0)	(100.0)
Chance agreement		49.98			50.00			50.00	
Kappa		0.5803			0.5260			0.5140	

5. Conclusion

In a quantitative context, data mining is widely applied in predicting stock prices

from financial time-series. However, few studies have discussed the potential of data mining for eliciting the qualitative problem-solving knowledge of experts from their predictions. This paper demonstrates a GA-based data mining approach to discovering such decision rules. This study is the first to employ GAs for this purpose. The fitness function of the GA is a composite measure that elicits decision rules satisfying two conditions relating to accuracy and coverage. This composite fitness function provides an efficient environment for reproduction. Effective learning strategies are implemented by means of genetic operators, including selection, crossover, mutation, and replacement, to generate useful rules.

Two data mining techniques, neural networks and inductive learning methods, have been applied as points of comparison with the GA method. The results of the experiments show that the performance of the GA method is significantly better than that of the neural networks or inductive learning methods in terms of predictive accuracy and coverage. They also show that considerable agreement is achieved between the GA method and expert problem-solving knowledge.

The proposed GA-based method has the potential to be useful in the prediction of stock prices. First, formulating models for the purpose of stock price prediction is an important yet difficult task; it requires access to subjective expert knowledge, given the complexity of the problem. This study provides effective support for such formulation, as it incorporates subjective knowledge into the models. It thus facilitates the efficient development of models of stock price prediction.

Second, further improvements in stock price prediction can be achieved via hybrid models that blend both quantitative and qualitative approaches. Numerous decision support studies have suggested that such integration can improve reasoning performance (Kuo *et al.* [15]). Several studies have proposed a variety of techniques for combining quantitative and qualitative models (Kim *et al.* [13]). The results of this study can be used in the development of hybrid models.

However, several issues remain to be addressed by further research. First, the structure of current rules is redundant and overlapping, because these rules have been generated from a data set with a small number of inputs. This structure could be refined considerably through the introduction of additional inputs. Such a refinement would facilitate both more efficient learning and more effective decision support.

Second, this study uses a small data set, as it is difficult to collect qualitative information. Therefore, further research should be conducted using a larger and more

general data set in order to confirm the efficiency of the proposed GA-based method.

Third, this study uses several learning strategies to improve the efficiency of the GAs. Unfortunately, we have no record of their impact on learning efficiency. Although it seems that these strategies should significantly reduce the learning time required to obtain rules by preventing the generation of useless rules, more advanced research is needed to further improve the algorithm. For example, further improvements could be achieved through an application of the niching method, which causes the population eventually to converge around a single point in the solution space (Mahfoud and Mani [7]). A GA that uses the niching method converges around multiple solutions or niches.

References

[1] Anglano C., G. Giordana, Bello Lo, and L. Saitta, "A network genetic algorithm for concept learning," Proceedings of the ICGA'97, Morgan Kaufmann, San Francisco, (1997), 434-441.

[2] Augier, S., G. Venturini, and Y. Kodratoff, Learning first order logic rules with a genetic algorithm, Proceedings of the First International Conference on Knowledge Discovery and Data Mining, AAAI Press, Menlo Park, CA, (1995), 21-26.

[3] Barr, D. S. and G. Mani, "Using neural nets to manage investments," AI Expert: (1994), 16-21.

[4] Bauer, R. J., Genetic algorithms and investment strategies, Wiley, NewYork, 1994.

[5] Brachman, R. J., T. Khabaza *et al.*, "Mining business databases," *Communication of the ACM* 39, 11 (1996), 42-48.

[6] Cohen, J., "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement* 20 (1960), 37-46.

[7] Giles, C., C. Lee, S. Lawrence, and A. C. Tsoi, "Rule inference for financial prediction using recurrent neural networks," *Proceedings of IEEE/IAFE Conference on Computational Intelligence for Financial Engineering*, Piscataway, NJ, (1997), 253-259.

[8] Giordana, A., L. Saitta, and F. Zini, "Learning disjunctive concepts by means of

genetic algorithms," *Proceedings of the 11th International Conference on Machine Learning*, (1994), 96-104.

[9] Hayashi, Y. and A. Imura, "Fuzzy neural expert system with automated extraction of fuzzy if-then rules from a trained neural network," *The first International Symposium on Uncertainty Modeling and Analysis*, (1990), 489-494.

[10] Holland, J. H., "Adaptation in Natural and Artificial Systems," Ann Arbor: The University of Michigan Press, 1975.

[11] Ishibuchi, H., K. Kozaki, N. Yamamoto, and H. Tanaka, "Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms," *Fuzzy Sets and Systems* 65 (1994), 237-253.

[12] Kass, G., "An exploratory technique for investigating large quantities of categorical data," *Applied Statistics* 29, 2 (1980), 119-127.

[13] Kim, E., W. Kim, and Y. Lee, "Combination of multiple classifiers for the customer's purchase behavior prediction," *Decision Support Systems* 34 (2002), 167-175.

[14] Kim, K. and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert systems with applications* 19, 2 (2000), 125-132.

[15] Kuo, R. J., C. H. Chen, and Y. C. Hwang, "An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network," *Fuzzy sets and Systems* 118 (2001), 21-45.

[16] Lin, C. T. and C. S. G. Lee, "Neural network-based fuzzy logic control and decision system," *IEEE Transactions on Computer* 12 (1991), 1320-1336.

[17] Mahfoud, S. and G. Mani, "Genetic algorithm for predicting individual stock performance," Proceedings of the Third International Conference on Artificial Intelligence Applications on Wall Street, (1995), 174-181.

[18] Noda, E., A. A. Freitas, and H. S. Lopes, "Discovering interesting prediction rules with a genetic algorithm." Proceedings of the Congress on Evolutionary Computation, IEEE Press, Piscataway, NJ, 1999.

[19] Pankratz, A., Forecasting with univariate Box-Jenkins model. Wiley and Sons, 1983.

[20] Peña, C. A. and M. Sipper, "Designing breast cancer diagnosis system via a hybrid fuzzy-genetic methodology," *Proceedings of IEEE International Fuzzy Systems Conferenc, IEEE Press, Piscataway*, NJ, 1 (1999), 135-139.

[21] Quinlan, J. R., "Induction of decision trees," *Machine Learning* 1 (1986), 81-106.

[22] Quinlan, J. R., C4.5-Programs for Machine learning, Palo Alto: Morgan Kaufmann, 1993.

[23] Tsaih, R., Y. Hsu, and CC. Lai, "Forecasting S&P 500 stock index futures with a hybrid AI system," *Decision support Systems* 23 (1998), 161-174.

[24] Walczak, S., "Gaining competitive advantage for trading in emerging capital markets with neural networks," *Journal of Management Information Systems* 16, 2 (1999), 177-192.

[25] Yao, X. and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Transactions on Neural Networks* 8, 3 (1997), 694-713.

[26] Yuan, Y. and H. Zhuang, "A genetic algorithm for generating fuzzy classification rules," *Fuzzy Sets and Systems* 84 (1996), 1-19.