Toward global optimization of case-based reasoning for the prediction of stock price index[*]

Kyoung-jae Kim and Ingoo Han[**]

Abstract

This paper presents a simultaneous optimization approach of case-based reasoning (CBR) using a genetic algorithm (GA) for the prediction of stock price index. Prior research suggested many hybrid models of CBR and the GA for selecting a relevant feature subset or optimizing feature weights. Most studies, however, used the GA for improving only a part of architectural factors of the CBR system. However, the performance of CBR may be enhanced when these factors are simultaneously considered. In this study, the GA simultaneously optimizes multiple factors of the CBR system. Experimental results show that a GA approach to simultaneous optimization of CBR outperforms other conventional approaches for the prediction of stock price index.

Key words: Simultaneous optimization; Case-based reasoning; Genetic algorithms; Stock market prediction

1. Introduction

Stock market prediction is the long-cherished desire of investors, speculators, and industries. Although many studies investigated the prediction of price movements in the stock market, financial time series is too complex and noisy to forecast. Many researchers attempted to predict price movements in the stock market using artificial intelligence (AI) techniques during past decades.

The earliest studies of this area are mainly focused on applications of artificial neural networks (ANNs) to stock market prediction (Kimoto et al., 1990; Kamijo & Tanigawa, 1990; Ahmadi, 1990; Yoon & Swales, 1991; Trippi & DeSieno, 1992; Choi et al., 1995). Recent research tends to hybridize several AI techniques (Hiemstra, 1995; Tsaih et al., 1998). Some researchers tend to include novel factors in the learning process. Kohara et al. (1997) incorporated prior knowledge to improve the performance of stock market prediction. Lee & Jo (1999) developed a candlestick chart analysis expert system for predicting the best stock market timing.

Case-based reasoning (CBR) is a reasoning technique that reuses past cases to find a solution to the new problem. This technique is popularly applied to many applications because it seems to overcome the drawbacks of existing rule-base systems. Previous research suggested that the integration of domain knowledge into case indexing and retrieval process is important in building a useful CBR system (Shin and Han, 1999). However, this task is very difficult because domain knowledge cannot easily be captured. In addition, the existence of continuous data and large amounts of records may pose a challenging task to explicit concepts extraction from the raw data due to the huge data space determined by continuous features (Liu and Setiono, 1996). The reduction and transformation of the irrelevant and redundant features may shorten the running time of reasoning and yield more generalized results (Dash and Liu, 1997). Prior research tried to solve these two problems but they did not considered these problems simultaneously. If these factors are considered separately, optimization is achieved in part, but may lead locally optimized solution as a whole. However, if these factors are simultaneously considered, the performance may be enhanced because the optimization of factors in a synergistic way may lead global optimization as a whole.

This paper proposes simultaneous optimization approach using genetic algorithms (GAs) for case representation process and indexing and retrieval processes in CBR system. This approach simultaneously selects the relevant feature subset and optimizes the thresholds for feature discretization. Feature discretization,

[*] This work is supported by the Korea Science and Engineering Foundation, under Grants No. 98-0102-08-01-3.

[**] Graduate School of Management,
Korea Advanced Institute of Science and Technology

the process of converting data sets with continuous attributes into input data sets with discrete attributes, filters the noisy data, then we get an enhanced prediction result. This paper applies the proposed approach to stock market analysis. Experimental results on applications will be presented.

The rest of this paper is organized as follows: The next section reviews prior research. Section 3 proposes the GA approach to simultaneous optimization of CBR system. In this section, the benefits of the proposed approach are presented. Section 4 describes the research design and experiments of applications. In the fifth section, the empirical results are summarized. In the final section, conclusions and the limitations of this study are presented.

2. Prior research

CBR is composed of the steps of case representation, indexing, retrieval, and adaptation. This paper simultaneously optimizes case representation step and indexing and retrieval step. In this section, we review basic concepts and prior studies on these steps in CBR system.

2.1. Case representation step

The step of case representation clearly, concisely, and truthfully represents cases to reflect specific knowledge in the case-base. The appropriate case representation relies on the characteristics of the problem domain (Brown and Gupta, 1994). If the problem domain is changed according to time, case representation must have sufficient detail to be able to judge the applicability of the case in the new situation (Kolodner, 1993).

On the other hand, the reduction and transformation of the irrelevant and redundant features may shorten the running time of reasoning and yield more generalized results (Dash and Liu, 1997). The conventional CBR system that simply combines different metrics for continuous and discrete attributes can lead to poor performance. The existence of continuous data and large amounts of records may pose a challenging task to explicit concepts extraction from the raw data due to the huge data space determined by continuous features (Liu and Setiono, 1996). In this aspect, Ting (1997) proposed discretization method for the continuous features in lazy learning algorithms including k-nearest neighbor. He used the entropy minimization strategy (Fayyad and Irani, 1993) to discretize the continuous features. He showed that discretization can improve the performance both in data sets with mixed continuous and discrete attribute types and data sets with only continuous attributes.

Feature discretization has been studied in many papers. The methods of feature discretization are classified as endogenous versus exogenous, local versus global, parameterized versus non-parameterized, and hard versus fuzzy (Dougherty et al., 1995; Scott et al., 1997; Susmaga, 1997).

Endogenous methods do not take into consideration the value of the dependent feature while exogenous methods do. Local methods discretize one attribute at once while the global ones discretize all features simultaneously. Parameterized methods specify the maximal number of intervals generated in advance while non-parameterized methods determine it automatically. Hard methods discretize the intervals at the cutting point exactly while fuzzy methods discretize it by overlapping bounds (Susmaga, 1997).

The endogenous methods include discretizing by the self-organizing map (Lawrence et al., 1996), the percentile method (Scott et al., 1997; Buhlmann, 1998), and the clustering method (Scott et al., 1997; Kontkanen et al., 1997). Basak et al. (1998) proposed a neuro-fuzzy approach using the feature evaluation index and Piramuthu et al. (1998) suggested a decision-tree based approach as an endogenous discretization method. These methods have the advantage of simplicity in the discretization process. However, they do not consider the association among each independent and dependent feature. The prediction performance is enhanced by the ability of discrimination not only by a single feature but also by the association among features. For this limitation, the endogenous method does not provide an effective way of forming categories (Scott et al., 1997).

Exogenous methods include maximizing the statistical significance of Cramer's V between other dichotomized variables (Scott et al., 1997), entropy minimization heuristic in inductive learning and the k-nearest neighbor method (Fayyad and Irani, 1993; Ting, 1997; Martens et al., 1998). The exogenous method also includes feature discretization using the GA for ANN (Kim and Han, 2000). These methods discretize an independent feature to maximize its association with the values of dependent and other independent features.

2.2. Case indexing and retrieval step

Case indexing is the task of assigning labels to cases to ensure that they can be retrieved at appropriate times (Kolodner, 1993). There are few guidelines for selecting good indexes. In most research, indexes are selected by the domain experts. The better the experts understand the domain, the better the index tends to be. But, human experts may not always select good indexes. If the process of index selection is automated, the consistency and maintainability of the index are enhanced.

One of the most popular indexing methods is feature weighting. Feature weighting is assigning a weight to each feature according to the relative importance of each one. Wettschereck et al. (1997) presented various feature weighting methods based on distance metrics in the machine learning literature. Kelly and Davis (1991) proposed the GA-based weighting method for k nearest neighbor. Shin and Han (1999) applied the GA-based feature weighting to the prediction of corporate bond rating. Kim and Shin (2000) presented feature weighting methods based on ANN and the GA. In addition, Liao et al. (2000) used this method for failure-mechanism identification.

Feature subset selection also be considered as a popular case indexing method. Siedlecki and Sklansky (1989) proposed a feature selection algorithm based on genetic search and Cardie (1993) presented a decision tree approach to feature subset selection. Skalak (1994) and Domingos (1997) also proposed a hill climbing algorithm and a clustering method as methods of feature subset selection. Most their approaches are classified as endogenous methods for feature subset selection, and they have the limitations of endogenous one. In addition, Cardie and Howe (1997) used a mixed feature weighting and feature subset selection method. They first selected relevant features using decision tree, then they assigned weights to the remained features using the value of information gain for each feature. Jarmulak et al. (2000a, 2000b) selected relevant features using decision tree algorithm including C4.5 (Quinlan, 1993) and assigned feature weights using the GA. Table 1 presents prior research on feature analysis for CBR.

Table 1
Prior research on feature analysis for CBR

Feature analysis	Description	References
Feature weighting	Using the GA	Kelly and Davis (1991); Shin and Han (1999); Kim and Shin (2000); Liao et al. (2000);
Feature subset selection	Using the GA	Siedlecki and Sklanski (1989)
	Using the decision tree	Cardie (1993)
	Using the random mutation hill climbing algorithm	Skalak (1994)
	Using the clustering technique (local selection)	Domingos (1997)
Feature selection and weighting	Feature selection using the decision tree and weighting the remaining features using information gain	Cardie and Howe (1997)
	Feature selection using the decision tree and weighting the remaining features using the GA	Jarmulak et al. (2000a); Jarmulak et al. (2000b)

Once cases are represented and indexed, the retrieval process is initiated. The indexing and retrieval processes consist of two phases. The first phase is to determine the relative importance of case attributes for the current problem. In the second phase, the case must be matched in the case library using these attributes and their specified importance (Buta, 1994). Nearest-neighbor matching techniques are popularly employed in this step.

2.3. Genetic algorithms

The GA has been investigated recently and shown to be effective in exploring a complex space in an adaptive way, guided by the biological evolution of *selection, crossover,* and *mutation* (Adeli and Hung, 1995). This algorithm uses natural selection, *survival of the fittest,* to solve optimization problems.

The first step of the GA is problem representation. The problem must be represented in a suitable form to be handled by the GA. Thus, the problem is described in terms of genetic code, like DNA chromosomes. The GA often works with a form of binary coding. If the problems are coded as chromosomes, the populations are initialized. Each chromosome within the population gradually evolves through biological operations. There are no general rules for determining the population size. But, population sizes of 100-200 are commonly used in GA research. Once the population size is chosen, the initial population is randomly generated (Bauer, 1994). After the initialization step, each chromosome is evaluated by a fitness function. According to the value of the fitness function, the chromosomes associated with the fittest individuals will be reproduced more often than those associated unfit individuals (Davis, 1994).

The GA works with three operators that are iteratively used. The *selection* operator determines which individuals may survive (Hertz et al., 1991). The *crossover* operator allows the search to fan out in diverse directions looking for attractive solutions and permits chromosomal material from different parents to be combined in a single child. There are three popular crossover methods: single-point, two-point, and uniform. The single-point crossover makes only one cut in each chromosome and selects two adjacent genes on the chromosome of a parent. The two-point crossover involves two cuts in each chromosome. The uniform crossover allows two parent strings to produce two children. It permits great flexibility in the way strings are combined. In addition, the *mutation* operator arbitrarily alters one or more components of a selected chromosome. Mutation randomly changes a gene on a chromosome. It provides the means for introducing new information into the population. Finally, the GA tends to converge on an optimal or near-optimal solution through these operators (Wong and Tan, 1994).

3. Simultaneous optimization of case-based reasoning using genetic algorithms

As mentioned in Section 2, prior studies suggested that feature weighting and feature subset selection are very important to enhance the prediction accuracy of the CBR system. In addition, Ting (1997) suggested that feature discretization using entropy minimization strategy can also enhance the classification accuracy. They, however, still did not simultaneously considered the architectural factors for designing CBR system. The followings are the architectural factors of designing CBR system.

The first architectural factor is relevant feature subset. Irrelevant and redundant features may cause distortion in relationship between the input and the output. The second factor, a rather novel one, is the thresholds for feature discretization. Feature discretization filters noisy data, then enhances prediction performance and generalizability.

This paper proposes the GA as the method of case representation and indexing for CBR system. For the case representation, the GA transforms the representation of each case. If a fitness function is specified, the GA searches for the (near-)optimal form of representation through the process of discretization. Properly discretized features can simplify the reasoning process and may improve the generalizability because it may effectively reduce the noisy and redundant data. Feature discretization needs relevant and rational discretizing thresholds. The thresholds, however, may vary according to the environments being analyzed. For this reason, there are no general guidelines of discretization. We may search the thresholds for discretizing a continuous measure into a qualitative norm to grasp the domain-specific knowledge.

In this paper, the GA is also engaged as the selection method of relevant feature subset for case indexing. Feature subset selection reduces the reasoning time and produces generalized results. The consistency and maintainability of the index may be enhanced through the automated case indexing method.

To verify the effectiveness of the proposed model, we compare the results of four different models. The first model, labeled *COCBR* (conventional CBR), uses the conventional approach for the reasoning process of CBR. This model considers all available features. In addition, the relative importance of each feature is not considered because most conventional CBR systems do not have general feature selection algorithm. Linear scaling is used as a feature transformation method. Linear scaling means linear scaling to unit variance in this study. It transforms a feature component x to a random variable with zero mean and unit variance (Jain and Dubes, 1988). It is usually employed to enhance the performance of CBR system because it ensures the larger value input features do not overwhelm smaller value input features.

In the second model, the GA assigns relevant feature weights. This study names this model *FWCBR* (feature weighting using the GA for CBR). This model uses linear scaling for transforming the original data. Similar models to it were previously suggested by Kelly and Davis (1991), Shin and Han (1999), Kim and Shin (2000), and Liao et al. (2000).

The third model uses the GA to select a relevant feature subset. This study names this model *FSCBR* (feature selection using the GA for CBR). This model also uses linear scaling for feature transformation. Siedlecki and Sklansky (1989) suggested similar model to it.

The fourth model, the proposed model in this study, uses the GA to select a relevant feature subset and to optimize the thresholds for feature discretization simultaneously. This study names this model *SOCBR* (simultaneous optimization using the GA for CBR). The framework of SOCBR is shown in Figure 1.

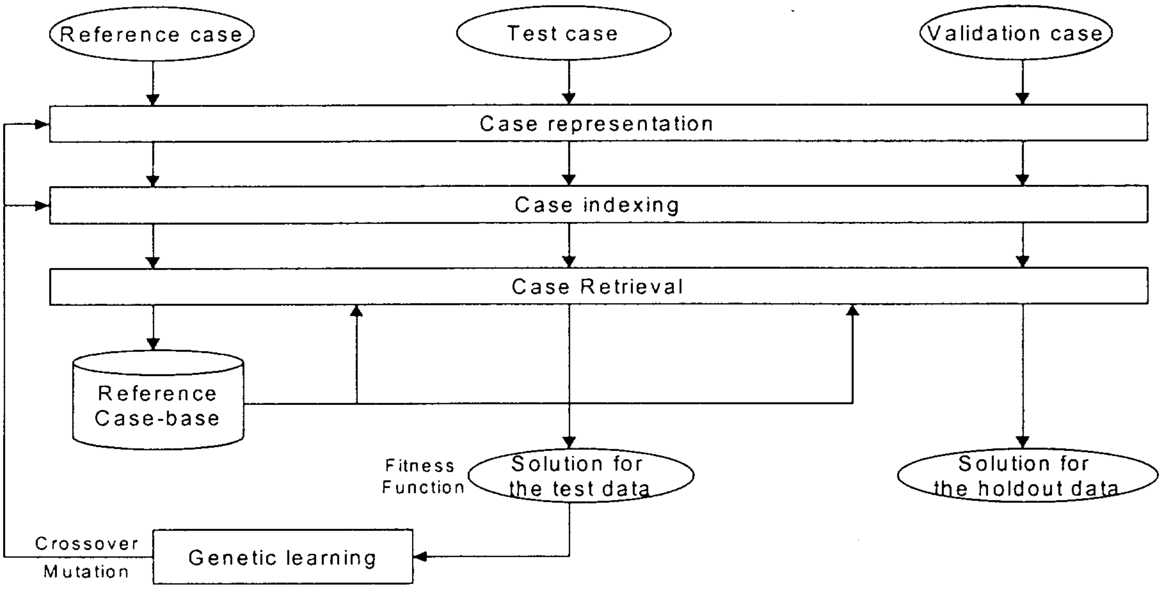

Figure 1.
Framework of SOCBR

Table 2 shows summary of these four models.

Table 2.
Summary of four models

	Feature weighting	Feature subset selection	Feature discretization
COCBR	-	-	LS**
FWCBR	GA*	-	LS
FSCBR	-	GA	LS
SOCBR	-	GA	GA

* Genetic algorithm, ** Linear scaling

Among three hybrid models, this study describes the optimization process of SOCBR. The optimization processes of FWCBR and FSCBR are like SOCBR except for the process of feature discretization. The process of SOCBR consists of the following three stages:

Stage 1. For the first stage, we search the search space to find optimal or near-optimal feature subset and the thresholds of feature discretization. The population – the codes for features subset and the thresholds for feature discretization – are initialized into random values before the search process. The parameters for searching must be encoded on chromosomes. The encoded chromosomes are searched to maximize the specific fitness function. The objectives of this paper are to select relevant feature subset and to approximate rational thresholds of feature discretization for the correct solutions. These objectives can be represented by the average prediction accuracy of the test data. Thus, this paper applies it to the fitness function. In this stage, the GA operates the process of crossover and mutation on initial chromosomes and iterates until the stopping conditions are satisfied.

Stage 2. The second stage is the process of retrieval and matching for new problem in CBR system. In this stage, nearest-neighbor method is used as the method of case retrieval. This method is a popular retrieval method because it can easily applied for numeric data such as financial data.

Stage 3. In the third stage, selected feature subset and the thresholds for feature discretization are applied to the holdout data. This stage is required because the GA optimizes the parameters to maximize the average predictive accuracy of the test data, but sometimes the optimized parameters are not generalized to deal with the unknown data. Table 3 summarizes the algorithm of SOCBR.

Table 3.
Step of SOCBR algorithm

Step 0.	Initialize the populations. (the feature subset and the thresholds for feature transformation) (Set to small random values between 0.0 and 1.0)
Step 1.	While stopping condition is false, do Steps 2-9.
Step 2.	Do Steps 3-8.
Step 3.	Computes the distance d_{ab} between new case in the test case-base x_b and each case in the reference case-base x_a. $d_{ab} = \sqrt{\sum_{i=1}^{n} W_i (x_{ai} - x_{bi})^2}$
Step 4.	Seek the best neighboring case x_b in the past which are closest to x_a according to the distance function.
Step 5.	Calculate the output for x_b from the output of x_a.
Step 6.	Calculate fitness. (Fitness function: Average predictive accuracy on the test case-base)
Step 7.	Select individuals to become parents of the next generation.
Step 8.	Create a second generation from the parent pool. (Perform crossover and mutation.)
Step 9.	Test the stop condition.

4. Research design and experiments

The research data used in this study comes from the daily Korean Composite Stock Price Index (KOSPI) from January 1989 to December 1998. The total number of samples includes 2,928 trading days. Initial features are 12 technical indicators are presented in Table 4. These features are selected by the review of domain experts and prior research (Achelis, 1995; Choi, 1995; Chang et al., 1996; Edwards and Magee, 1997).

Table 4.
Selected features and their formulas

Names of feature	Formulas
Stochastic %K	$\frac{C_t - L_n}{H_n - L_n} \times 100$
Stochastic %D	$\frac{\sum_{i=0}^{n-1} \% K_{t-i}}{n}$
Stochastic slow %D	$\frac{\sum_{i=0}^{n-1} \% D_{t-i}}{n}$
Momentum	$C_t - C_{t-4}$
ROC (rate of change)	$\frac{C_t}{C_{t-n}} \times 100$
LW %R (Larry William's %R)	$\frac{H_n - C_t}{H_n - L_n} \times 100$
A/D Oscillator (accumulation/distribution oscillator)	$\frac{H_t - C_{t-1}}{H_t - L_t}$
Disparity 5 days	$\frac{C_t}{MA_5} \times 100$
Disparity 10 days	$\frac{C_t}{MA_{10}} \times 100$
OSCP (price oscillator)	$\frac{MA_5 - MA_{10}}{MA_5}$
CCI (commodity channel index)	$\frac{(M_t - SM_t)}{(0.015 \times D_t)}$
RSI (relative strength index)	$100 - \frac{100}{1 + \frac{\sum_{i=0}^{n-1} Up_{t-i} / n}{\sum_{i=0}^{n-1} Dw_{t-i} / n}}$

C: Closing price, L: Low price, H: High price, MA: Moving average of price, M$_t$: $\frac{(H_t + L_t + C_t)}{3}$, SM$_t$: $\frac{\sum_{i=1}^{n} M_{t-i+1}}{n}$, D$_t$: $\frac{\sum_{i=1}^{n} |M_{t-i+1} - SM_t|}{n}$, Up / Dw: Upward / Upward price change

The data used in this study are split into three case-bases. The first case-base is the reference case-base. This case-base is used to develop the system. The test case-base is the second one. This case-base measures how well the system interpolates using the selected feature subset and the derived thresholds for feature discretization through the evolutionary search process from the reference case-base. The holdout case-base is the third one and this case-base is used to validate the generalizability of model for the unseen data. The number of cases in each case-base is shown in Table 5.

Table 5.
Number of each case-base

Case-base	Year										Total
	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	
Reference	162	163	163	165	165	165	164	164	163	163	1,637
Test	70	70	71	71	72	72	71	71	71	71	710
Holdout	57	58	58	58	59	59	58	58	58	58	581
Total	289	291	292	294	296	296	293	293	292	292	2,928

This study needs two sets of parameters. The first set represents the thresholds for feature discretization. Each case is discretized into at most 5 categories using these thresholds. In addition, the GA searches the number of categories to be discretized using these thresholds. The third set is the selection codes for relevant feature subset.

The strings used in this study have the following encoding:

The first 48 bits represent the thresholds for feature discretization. These bits varied between -3 and 2. Each feature is discretized into at most five categories and needs four thresholds for discretizaion. The next 36 bits indicate the distances between the discretizing thresholds. These bits are searched from 0 to 100. As mentioned earlier, the GA searches the number of categories to be discretized using these bits. The thresholds are not used if the searched thresholds are more than the maximum value of each feature. The upper limit of the number of categories is 5 and the lower limit is 1. This number is automatically determined by the searching process.

The next 12 bits represent the selection codes for relevant feature subset. These bits are searched to categorize them as 0 and 1. Each bit indicates whether the associated features are included or excluded for the reasoning process. The feature with "0" is excluded and that with "1" is included in the step of case retrieval.

For the controlling parameters of GA search, the population size is set to 100 organisms and the crossover and mutation rates are varied to prevent ANN from falling into a local minimum. The range of the crossover rate is set between 0.5 and 0.7 while the mutation rate ranges from 0.05 to 0.1. This study performs the crossover using a uniform crossover routine. The uniform crossover method is considered better at preserving the schema, and can generate any schema from the two parents, while single-point and two-point crossover methods may bias the search with the irrelevant position of the features. For the mutation method, this study generates a random number between 0 and 1 for each of the features in the organism. If a feature gets a number that is less than or equal to the mutation rate, then that feature is mutated. As the stopping condition, only 5000 trials are permitted. The parameters to be searched use only the information about the reference and the test case-base.

5. Experimental results

In this section, the prediction performances of four models are compared. Table 6 describes the average prediction accuracy of each model for the holdout data.

Table 6.
Average prediction accuracy for the holdout data

Year	COCBR	FWCBR	FSCBR	SOCBR
1989	56.1	56.1	56.1	58.8
1990	50.0	58.6	58.6	62.1
1991	51.7	55.2	56.9	62.1
1992	44.0	51.7	44.8	63.8
1993	49.2	52.5	54.2	61.0
1994	52.5	54.2	59.3	54.2
1995	58.6	55.2	55.2	60.3
1996	62.1	56.9	60.3	56.9
1997	51.7	53.4	53.4	62.1
1998	44.8	51.7	51.7	63.8
Total	52.06%	54.54%	55.05%	60.50%

In Table 6, SOCBR achieves higher prediction accuracy than COCBR, FWCBR and FSCBR by 8.95 %, 5.96 % and 5.45 % for the holdout data. FSCBR outperforms COCBR and FWCBR by 0.51 % and 2.99 % for the holdout data. In addition, FWCBR outperforms COCBR by 2.48 %. These results may be caused by the benefits of the global search techniques.

The McNemar tests are used to examine whether SOCBR significantly outperforms the other three models. This test is used with nominal data and is particularly useful with before-after measurement of the same subjects (Cooper and Emory, 1995). Table 7 shows the results of the McNemar test to compare the performances of four models for the holdout data.

Table 7.
McNemar values for the holdout data

	FWCBR	FSCBR	SOCBR
COCBR	10.370**	4.156*	16.247**
FWCBR		2.704	0.253
FSCBR			4.033*

* significant at the 5 % level, ** significant at the 1 % level

As shown in Table 7, SOCBR better than COCBR at the 1 % and outperforms FSCBR with the 5 % statistical significance level. In addition, Table 7 shows that FSCBR and FWCBR outperform COCBR with the 5 % and 1 % statistical significance level.

In addition, the two-sample test for proportions is executed. This test is designed to distinguish between two proportions when the prediction accuracy of the left-vertical methods is compared with those of the right-horizontal methods (Harnett and Soni, 1991). Table 8 shows p values for the pairwise comparison of performance between models.

Table 8.
p values for the holdout data

	FWCBR	FSCBR	SOCBR
COCBR	0.1985	0.1535	0.0019
FWCBR		0.4307	0.0200
FSCBR			0.0301

As shown in Table 8, SOCBR better than COCBR at the 1 % and outperforms FSCBR and FWCBR with the 5% statistical significance level.

6. Conclusions

In this paper, we use the GA for CBR system in two ways. We first adopt feature discretization based on the GA. Second, we use the GA to select relevant feature subset for CBR system. From the results of the experiment, it is apparent that for stock market prediction, the hybrid model of GA and CBR offers a viable alternative approach. Empirical results show that SOCBR offers better predictive performance than FSCBR, FWCBR and COCBR.

This study has some limitations. There will be other steps which enhance the prediction performance of CBR if they are incorporated with the simultaneous optimization model. SOCBR produces valid results, however, the GA can potentially be used to optimize another step of the reasoning process of CBR system including case deletion and case adaptation. The prediction performance may be enhanced if the GA is employed for relevant instance selection and this remains an interesting topic for further study. In addition, further research will extend the method of feature discretization using other global search algorithms including a tabu search algorithm. Of course, the generalizability of SOCBR should be tested further by applying it to other problem domains.

References

Achelis, S.B. (1995). *Technical analysis from A to Z*. Chicago: Probus Publishing.

Adeli, H., & Hung, S. (1995). *Machine learning: Neural networks, genetic algorithms, and fuzzy systems*, New York: Wiley.

Ahmadi, H. (1990). Testability of the arbitrage pricing theory by neural networks. *Proceedings of the*

International Conference on Neural Networks (pp. 385-393). San Diego, California.

Basak, J., De, R.K., & Pal, S.K. (1998). Unsupervised feature selection using a neuro-fuzzy approach. *Pattern Recognition Letters*, *19*(11), 997-1006.

Bauer, R.J. (1994). *Genetic algorithms and investment strategies*. New York: Wiley.

Brown, C.E., & Gupta, U.G. (1994). Applying case-based reasoning to the accounting domain. *International Journal of Intelligent Systems in Accounting, Finance and Management*, *3*, 205-221.

Buhlmann, P. (1998). Extreme events from the return-volume process: a discretization approach for complexity reduction. *Applied Financial Economics*, *8*, 267-278.

Buta, P. (1994). Mining for financial knowledge with CBR. *AI Expert*, *9*(2), 34-41.

Cardie, C. (1993). Using decision trees to improve case-based learning. *Proceedings of the Tenth International Conference on Machine Learning* (pp. 25-32). San Francisco, CA: Morgan Kaufmann.

Cardie, C., & Howe, N. (1997). Improving minority class prediction using case-specific feature weights. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 57-65). San Francisco, CA: Morgan Kaufmann.

Chang, J., Jung, Y., Yeon, K., Jun, J., Shin, D., & Kim, H. (1996). *Technical indicators and analysis methods*. Seoul: Jinritamgu Publishing.

Choi, J. (1995). *Technical indicators*. Seoul: Jinritamgu Publishing.

Choi, J.H., Lee, M.K., & Rhee, M.W. (1995). Trading S&P 500 stock index futures using a neural network. *Proceedings of the Third Annual International Conference on Artificial Intelligence Applications on Wall Street* (pp. 63-72). New York.

Cooper, D.R., & Emory, C.W. (1995). *Business research methods*. Chicago, Illinois: Irwin.

Dash, M., & Liu, H. (1997). Feature selection methods for classifications. *Intelligent Data Analysis-An International Journal*, *1*(3), 131-156.

Davis, L. (1994). *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold.

Domingos, P. (1997). Context-sensitive feature selection for lazy learners. *Artificial Intelligence Review*, *11*, 227-253

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 194-202). San Francisco, California.

Edwards, R.D., & Magee, J. (1997). *Technical analysis of stock trends*. Chicago, Illinois: John Magee.

Fayyad, U.M., & Irani, K.B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the 13th International Joint Conference on Artificial Intelligence* (pp. 1022-1027).

Harnett, D.L., & Soni, A.K. (1991). *Statistical Methods for Business. and Economics*. Reading, Massachusetts: Addison-Wesley Publishing.

Hertz, J., Krogh, A., & Palmer, R.G. (1991). *Introduction to the theory neural computation*. Reading, MA: Addison-Wesley.

Hiemstra, Y. (1995). Modeling structured nonlinear knowledge to predict stock market returns, In R. R. Trippi, *Chaos & nonlinear dynamics in the financial markets: theory, evidence and applications* (pp. 163-175). Chicago, Illinois: Irwin.

Jain, A.K., & Dubes, R.C. (1988). *Algorithms for clustering data*. NJ: Prentice Hall.

Jarmulak, J., Craw, S., & Rowe, R. (2000a). Genetic algorithms to optimise CBR retrieval, In E. Blanzieri & L. Portinale, *Advances in case-based reasoning: Proceedings of EWCBR-2K* (pp. 136-147). Trento, Italypp.

Jarmulak, J., Craw, S., & Rowe, R. (2000b). Self-optimising CBR retrieval, In *ICTAI-2000 Proceedings* (pp.376-383). Vancouver, Canada.

Kelly, J.D.J., & Davis, L. (1991). Hybridizing the genetic algorithm and the k nearest neighbors classification algorithm. *Proceedings of the Fourth International Conference on Genetic Algorithms* (pp. 377-383). San Diego, CA: Morgan Kaufmann.

Kamijo, K., & Tanigawa, T. (1990). Stock price pattern recognition: A recurrent neural network approach. *Proceedings of the International Joint Conference on Neural Networks* (pp. 215-221). San Diego, California.

Kim, K., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, *19*(2), 125-132.

Kim, S.H., & Shin, S.W. (2000). Identifying the impact of decision variables for nonlinear classification tasks. *Expert Systems with Applications*, *18*, 201-214.

Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural network. *Proceedings of the International Joint*

Conference on Neural Networks (pp. 1-6). San Diego, California.

Kohara, K., Ishikawa, T., Fukuhara, Y., & Nakamura, Y. (1997). Stock price prediction using prior knowledge and neural networks. *International Journal of Intelligent Systems in Accounting, Finance and Management*, *6*(1), 11-22.

Kolodner, J. (1993). *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann.

Kontkanen, P., Myllymaki, P., Silander, T., & Tirri, H. (1997). A Bayesian approach to discretization. *Proceedings of the European Symposium on Intelligent Techniques* (pp. 265-268).

Lawrence, S., Tsoi, A.C., & Giles, C.L. (1996). Noisy time series prediction using symbolic representation and recurrent neural network grammatical inference. In Institute for Advanced computer Studies, *Technical report UMIACS-TR-96-27 and CS-TR-3625*. University of Maryland.

Lee, K.H., & Jo, G.S. (1999). Expert system for predicting stock market timing using a candlestick chart. *Expert Systems with Applications, 16*(4), 357-364.

Liao, T.W., Zhang, Z.M., & Mount, C.R. (2000). A case-based reasoning system for identifying failure mechanisms. *Engineering Applications of Artificial Intelligence*, *13*(2), 199-213.

Liu, H., & Setiono, R. (1996). Dimensionality reduction via discretization. *Knowledge-Based Systems*, *9*(1), 67-72.

Martens, J., Wets, G., Vanthienen, J., & Mues, C. (1998). An initial comparison of a fuzzy neural classifier and a decision tree based classifier. *Expert Systems with Applications*, *15*, 375-381.

Piramuthu, S., Ragavan, H., & Shaw, M.J. (1998). Using feature construction to improve the performance of neural networks. *Management Science*, *44*(3), 416-430.

Quinlan, J.R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Scott, P.D., Williams, K.M., & Ho, K.M. (1997). Forming categories in exploratory data analysis and data mining. In X. Liu, P. Cohen, & M. Berthold, *Advances in intelligent data analysis* (pp. 235-246). Berlin: Springer-Verlag.

Shin, K., & Han, I. (1999). Case-based reasoning supported by genetic algorithms for corporate bond rating. *Expert Systems with Applications*, *16*, 85-95.

Siedlecki, W., & Sklanski, J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, *10*, 335-347.

Skalak, D.B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 293-301). New Brunswick, New Jersey.

Susmaga, R. (1997). Analyzing discretizations of continuous attributes given a monotonic discrimination function. *Intelligent Data Analysis-An International Journal*, *1*(3), 157-179.

Ting, K.A. (1997). Discretization in lazy learning algorithms. *Artificial Intelligence Review*, *11*, 157-174.

Trippi, R.R., & DeSieno, D. (1992). Trading equity index futures with a neural network. *The Journal of Portfolio Management*, *19*, 27-33.

Tsaih, R., Hsu, Y., & Lai, C.C. (1998). Forecasting S&P 500 stock index futures with a hybrid AI system. *Decision Support Systems*, *23*(2), 161-174.

Wettschereck, D., Aha, D.W., & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, *11*, 273-314.

Wong, F., and Tan, C. (1994). Hybrid neural, genetic and fuzzy systems. In G.J. Deboeck, *Trading On The Edge* (pp. 243-261). New York: Wiley.

Yoon, Y., & Swales, G. (1991). Predicting stock price performance: A neural network approach. *Proceedings of the 24th Annual Hawaii International Conference on System Sciences* (pp.156-162).