



# Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index

Kyoung-jae Kim<sup>\*</sup>, Ingoo Han<sup>1</sup>

*Graduate School of Management, Korea Advanced Institute of Science and Technology, 207-43 Cheongryangri-Dong, Dongdaemun-Gu, Seoul 130-012, South Korea*

## Abstract

This paper proposes genetic algorithms (GAs) approach to feature discretization and the determination of connection weights for artificial neural networks (ANNs) to predict the stock price index. Previous research proposed many hybrid models of ANN and GA for the method of training the network, feature subset selection, and topology optimization. In most of these studies, however, GA is only used to improve the learning algorithm itself. In this study, GA is employed not only to improve the learning algorithm, but also to reduce the complexity in feature space. GA optimizes simultaneously the connection weights between layers and the thresholds for feature discretization. The genetically evolved weights mitigate the well-known limitations of the gradient descent algorithm. In addition, globally searched feature discretization reduces the dimensionality of the feature space and eliminates irrelevant factors. Experimental results show that GA approach to the feature discretization model outperforms the other two conventional models. © 2000 Published by Elsevier Science Ltd.

**Keywords:** Feature discretization; Connection weight optimization; Genetic algorithms; Artificial neural networks; The prediction of stock price index

## 1. Introduction

For a long time, there has been much research interest in predicting the stock price index. Among them, there are many studies using data mining techniques including artificial neural networks (ANNs). However, most studies showed that ANN had some limitations in learning the patterns because stock market data has tremendous noise and complex dimensionality. ANN has preeminent learning ability while it is often confronted with inconsistent and unpredictable performance for noisy data. In addition, sometimes the amount of data is so large that the learning of patterns may not work well. In particular, the existence of continuous data and large amount of data may pose a challenging task to explicit concepts extraction from the raw data due to the huge amount of data space determined by continuous features (Liu & Setiono, 1996). Many researchers in the society of data mining are interested in the reduction of dimensionality. The reduction and transformation of the irrelevant or redundant features may shorten the running time and yield more generalized results (Dash & Liu, 1997).

This paper proposes a new hybrid model of ANN and

genetic algorithms (GAs) for feature discretization to mitigate the above limitations. Feature discretization is to transform continuous values into discrete ones in accordance with certain thresholds. Feature discretization is closely related to the dimensionality reduction (Liu & Motoda, 1998a). Properly discretized data can simplify the process of learning and may improve the generalizability of the learned results. This study uses GA to search the optimal or near-optimal thresholds for feature discretization. In addition, this study simultaneously searches the connection weights between layers in ANN. The genetically evolved connection weights mitigate the well-known limitations of the gradient descent algorithm.

The rest of the paper is organized as follows. Section 2 reviews prior research. Section 3 proposes feature discretization using GA and describes the benefits of the proposed approach. Section 4 describes the research design and experiments. In Section 5, the empirical results are summarized and discussed. In Section 6, conclusions and the limitations of this study are presented.

## 2. Research background

### 2.1. Prior research on stock market prediction using ANN

Many studies on stock market prediction using artificial

<sup>\*</sup> Corresponding author. Tel.: +82-2-958-3131; fax: +82-2-958-3604.

E-mail addresses: kkj@kgsm.kaist.ac.kr (K.-J. Kim), ighan@kgsm.kaist.ac.kr (I. Han).

<sup>1</sup> Tel.: +82-2-958-3613; fax: +82-2-958-3604.

intelligence (AI) techniques were performed during the past decade. These studies used various types of ANN to predict accurately the stock index and the direction of its change.

One of the earliest studies, Kimoto, Asakawa, Yoda and Takeoka (1990) used several learning algorithms and prediction methods for developing the Tokyo stock exchange prices index (TOPIX) prediction system. They used the modular neural network to learn the relationships among various market factors. Kamijo and Tanigawa (1990) used the recurrent neural network and Ahmadi (1990) employed the backpropagation neural network with the generalized delta rule to predict the stock market. Yoon and Swales (1991) also performed predictions using qualitative and quantitative data.

Some researchers investigated the issue of predicting the stock index futures market. Choi, Lee and Rhee (1995) and Trippi and DeSieno (1992) predicted the daily direction of change in the S&P 500 index futures using ANN. Duke and Long (1993) executed daily predictions of the German government bond futures using the backpropagation neural network. The above studies are mainly focused on applications of ANN to the stock market prediction.

Recent research tends to hybridize several AI techniques. Hiemstra (1995) proposed fuzzy expert systems to predict stock market returns. He suggested that ANN and fuzzy logic could capture the complexities of functional mapping because they do not require the specification of the function to approximate. A more recent study of Tsaih, Hsu and Lai (1998) integrated the rule-based technique and ANN to predict the direction of change of the S&P 500 stock index futures on a daily basis. Some researchers tend to include novel factors to the learning process. Kohara, Ishikawa, Fukuhara and Nakamura (1997) incorporated prior knowledge to improve the performance of stock market prediction.

Some of them, however, did not produce outstanding prediction accuracy partly because of the tremendous noise and non-stationary characteristics in stock market data. Lawrence, Tsoi and Giles (1996) pointed out that when the training of ANN tends to be difficult for high noisy data then the networks fall into a naive solution such as always predicting the most common output. Another reason may be the local convergence of the gradient descent algorithms. Most ANN used in prior research employed the gradient descent algorithm to train the network.

## 2.2. The connection weights in the ANN solutions

Many ANN studies relied on the gradient descent algorithm to get the connection weights of the model. Sexton, Alidaee, Dorsey and Johnson (1998a) pointed out that the gradient descent algorithm may perform poorly even on simple problems when predicting the holdout data. Their indication stems from the fact that backpropagation is a local search algorithm and may tend to fall into a local minimum.

Sexton, Dorsey and Johnson (1998b) indicated that the use of the momentum, restarting training at many random points, restructuring the network architecture, and applying significant constraints to the permissible forms can fix it. They also suggested that one of the most promising directions is using global search algorithms to search the weight vector of the network instead of local search algorithms including the gradient descent algorithm. They employed GA to search the weight vector of ANN. The results showed that the GA-derived solution was superior to the corresponding backpropagation solution. Gupta and Sexton (1999) and Ignizio and Soltys (1996) also suggested that the GA-derived solutions are better than the gradient descent algorithm derived solutions.

Some ANN research advocated that other global search algorithms can improve performance. Sexton et al. (1998a) used tabu search to optimize the network and tabu search-derived solutions were significantly superior to those of backpropagation solutions for all test data in the resulting comparison. In another paper, Sexton, Dorsey and Johnson (1999) again incorporated simulated annealing, one of global search algorithms, to optimize the network. They compared GA to the simulated annealing. They concluded that GA outperformed simulated annealing. On the other hand, Shin, Shin and Han (1998) concluded that backpropagation with the gradient descent algorithm outperform ANN with GA in their application on bankruptcy prediction. They concluded that GA solution cannot always guarantee better performance than ANN trained with the gradient descent algorithm.

## 2.3. The feature discretization method in data mining

The data mining society has been interested in feature transformation and subset selection because data preprocessing is an essential step for knowledge discovery. One of the most popular preprocessing methods is feature transformation. Feature transformation is the process of creating a new set of features (Liu & Motoda, 1998b). It may be split into three categories including feature extraction, feature construction, and feature discretization. Among them, feature discretization is closely related to dimensionality reduction (Liu & Motoda, 1998a). The methods of feature discretization are classified as endogenous (unsupervised) versus exogenous (supervised), local versus global, parameterized versus non-parameterized, and hard versus fuzzy (Dougherty, Kohavi & Sahami, 1995; Scott, Williams & Ho, 1997; Susmaga, 1997).

Endogenous (unsupervised) methods do not take into consideration the value of the dependent feature while exogenous (supervised) 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** (Buhlmann, 1998; Scott et al., 1997), and the **clustering method** (Kontkanen, Myllymaki, Silander & Tirri, 1997; Scott et al., 1997). Basak, De and Pal (1998) proposed the neuro-fuzzy approach using the feature evaluation index and Piramuthu, Ragavan and Shaw (1998) suggested the 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).

On the other hand, 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 & Irani, 1993; Martens, Wets, Vanthienen & Mues, 1998; Ting, 1997). The exogenous method also includes feature transformation using GA for C4.5 (Vafaie & De Jong, 1998). These methods discretize an independent feature to maximize its association with the values of dependent and other independent features.

#### 2.4. GA and the design of ANN architecture

GA is a search algorithm based on survival of the fittest among string structures (Goldberg, 1989). GA has been investigated recently and shown to be effective in exploring a complex space in an adaptive way, guided by the biological evolution mechanisms of reproduction, crossover, and mutation (Adeli & Hung, 1995).

The following describes the basic steps of GA.

The first step is problem representation. The problems must be represented as a suitable form to be handled by GA. 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 is gradually evolved by biological operations. Larger populations ensure greater diversity but require more computational burden. Once the population size is chosen, the initial population is randomly generated (Bauer, 1994). After the initialization step, each chromosome is evaluated by the 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 with unfit individuals (Davis, 1994).

Crossover 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. In addition, mutation arbitrarily alters one or

more components of a selected chromosome. It provides the means for introducing new information into the population. Finally, GA tends to converge on optimal or near-optimal solutions (Wong & Tan, 1994).

GA is usually employed to improve the performance of AI techniques. For ANN, GA is popularly used to select neural network topology such as optimizing relevant feature subset, determining the optimal number of hidden layers and processing elements. Feature subset, the number of hidden layers, the number of processing elements in hidden layers, the activation functions, and the connection weights between layers are the architectural factors of ANN to be determined in advance. Most previous studies were focused on the improvement of the learning algorithm itself. There are few studies on the dimensionality reduction and the elimination of irrelevant patterns for ANN.

This study proposes GA approach to feature discretization (GAFD) for ANN. In this study, GA supports the simultaneous optimization of connection weights and feature discretization. GAFD takes into consideration the dependent feature by fitness function in GA. GA iterates the evolution of the populations to maximize the fitness function. GAFD simultaneously discretizes all features into the intervals at the exact thresholds. In addition, GAFD determines the maximal number of thresholds automatically. GAFD is classified as an exogenous, global, hard, and non-parameterized discretization method.

### 3. GA approach to feature discretization for ANN

Many fund managers and investors in the stock market generally accept and use certain criteria for technical indicators as the signal of future market trends. Even if a feature represents a continuous measure, the experts usually interpret the values in qualitative terms such as low, medium, and high (Slowinski & Zopounidis, 1995). For 'Stochastic %K', one of the most popular technical indicators, the value of 75 is basically accepted by stock market analysts as a strong signal if the value exceeds 75, the market is regarded as an overbought situation or a bullish market. On the other hand, if it drops below 25, it is considered as an oversold situation or the signal of a bearish market. When the value of 'Stochastic %K' is placed between 25 and 75, it is regarded as the signal of a neutral market (Edwards & Magee, 1997).

The reasoning process of ANN may be like that of human experts. Properly discretized data can simplify the process of learning and may improve the generalizability of the learned results because it may effectively reduce the noisy and redundant data. Feature discretization needs relevant and rational discretizing thresholds. However, the thresholds may vary depending on the securities being analyzed and the overall market condition (Achelis, 1995). There are no general guidelines to discretize for this reason. We may search the thresholds for discretizing a continuous measure

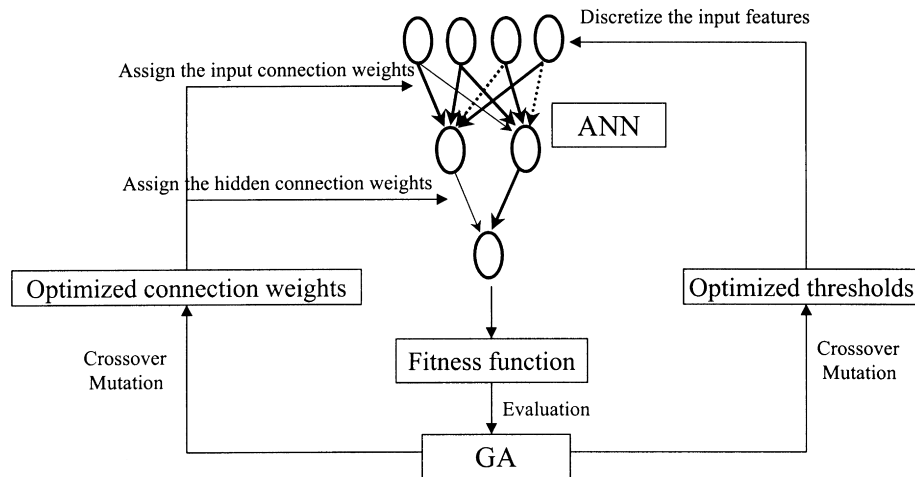


Fig. 1. Overall framework of GAFD.

into a qualitative norm to capture the domain-specific knowledge. As mentioned earlier, although some studies suggested various methods for discretizing features, this paper proposes the optimization of discretizing thresholds based on GA. GAFD may find optimal or near-optimal thresholds of discretization for maximum predictive performance because GA searches the optimal or near-optimal parameters to maximize the fitness function.

The overall framework of GAFD is shown in Fig. 1.

The algorithms of GAFD consist of three phases.

**Phase 1.** In the first phase, GA searches optimal or near-optimal connection weights and thresholds for feature discretization. The populations, the connection weights and the thresholds for feature discretization, are initialized into random values before the search process. The parameters for searching must be encoded on chromosomes. This study needs three sets of parameters. The first set is the set of connection weights between the input layer and the hidden layer of the network. The second set is the set of connection weights between the hidden layer and the output layer. As mentioned earlier, the above two sets may mitigate the limitation of the gradient descent algorithm. These sets were incorporated in the studies of Dorsey and Sexton (1998), Gupta and Sexton (1999), Ignizio and Soltys (1996), Sexton et al. (1998b, 1999) and Shin et al. (1998). The third set represents the thresholds for feature discretization.

The strings used in this study have the following encoding. This study uses 12 input features and employs 12 processing elements in the hidden layer. Each processing element in the hidden layer receives 12 signals from the input layer. The first 144 bits represent the connection weights between the input layer and the hidden layer. These bits are searched from  $-5$  to  $5$ . Each processing element in the output layer receives a signal from the hidden layer. The next 12 bits indicate the connection weights between the hidden layer and the output layer. These bits also varied between  $-5$  and  $5$ . The following 48 bits are the

thresholds for feature discretization. Each feature is discretized into at most five categories and needs four thresholds for discretization. In addition, GA also 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 five and the lower limit is one. This number is automatically determined by the searching process of GA.

The encoded chromosomes are searched to maximize the fitness function. The fitness function is specific to applications. In this study, the objectives of the model are to approximate connection weights and the thresholds of discretization for the correct solutions. These objectives can be represented by the average prediction accuracy of the training data. This study applies the average prediction accuracy of the training data to the fitness function.

The parameters to be searched use only the information about training data. In this phase, GA operates the process of crossover and mutation on initial chromosomes and iterates until the stopping conditions are satisfied. For the controlling parameters of the 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. As the stopping condition, only 5000 trials are permitted.

**Phase 2.** The second phase is the process of feedforward computation in ANN. In this phase, the sigmoid function is used as the activation function. This function is a popular activation function for the backpropagation neural network because it can easily be differentiated. The linear function is used as a combination function for the feedforward computation with derived connection weights from the first phase.

**Phase 3.** The derived connection weights and thresholds for feature discretization are applied to the holdout data. This phase is indispensable to validate the generalizability because ANN has the eminent ability of learning the known

Table 1  
The algorithms of GAFD

Step 0.	Initialize the populations (the connection weights between layers and the thresholds for feature discretization). (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.	Each processing element in the input layer receives an input signal and forwards this signal to all processing elements in the hidden layer.
Step 4.	Each processing element in the hidden layer sums its weighted input signals and applies the sigmoid activation function to compute its output signal of the hidden processing element and forwards it to all processing elements in the output layer.
Step 5.	Each processing element in the output layer sums its weighted signals from the hidden layer and applies the sigmoid activation function to compute its output signal of the output processing element and computes the difference between the output signal and the target value.
Step 6.	Calculate fitness. (Fitness function: average predictive accuracy on training set)
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 stopping condition.

data. If this phase is not carried out the model may fall into the problem of overfitting with the training data. Table 1 summarizes the algorithms of GAFD.

#### 4. Research data and experiments

The research data used in this study is technical indicators and the direction of change in the daily Korea stock price index (KOSPI). The total number of samples is 2928 trading days, from January 1989 to December 1998. Table 2 gives selected features and their formulas (Achelis, 1995; Chang, Jung, Yeon, Jun, Shin & Kim, 1996; Choi, 1995; Edwards & Magee, 1997; Gifford, 1995).

The direction of daily change in the stock price index are categorized as “0” or “1”. “0” means that the next day’s index is lower than the today’s index, and “1” means that the next day’s index is higher than today’s index. We select 12 technical indicators as feature subsets by the review of domain experts and prior research. Table 3 presents the summary statistics for each feature.

This study compares GAFD to the linear transformation with the backpropagation neural network (BPLT) and to the linear transformation with ANN trained by GA (GALT). In this study, linear transformation means the linear scaling of data into the range of 0.0–1.0. Linear transformation is generally used to enhance the performance of ANN because most ANN models accept numeric data only in the range from 0.0 to 1.0 or –1.0 to +1.0 (Bigus, 1996). BPLT uses

Table 2

Selected features and their formulas ( $C$  is the closing price,  $L$  the low price,  $H$  the high price,  $MA$  the moving average of price,  $M_t$ :  $(H_t + L_t + C_t)/3$ ,  $SM_t$ :  $(\sum_{i=1}^n M_{t-i+1})/n$ ,  $D_t$ :  $(\sum_{i=1}^n |M_{t-i+1} - SM_t|)/n$ ,  $Up$  the upward price change,  $Dw$  the downward price change)

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

the gradient descent algorithm to train the network. This is the conventional approach of previous studies. GALT employs GA only to determine the connection weights of ANN. GALT is similar to the model of Gupta and Sexton (1999), Sexton et al. (1998b) and Shin et al. (1998). The number of processing elements in the hidden layer is fixed at 12. This is like the number of feature subsets.

About 20% of the data is used for holdout and 80% for training. The training data is used to search the optimal or near-optimal parameters and is employed to evaluate the fitness function. The holdout data is used to test the results with the data that is not utilized to develop the model. The number of cases for each set is shown in Table 4.

#### 5. Experimental results

Three models are compared according to the methods of determining the connection weights and the methods of feature transformation. Table 5 describes the average prediction accuracy of each model.

In Table 5, GAFD has higher prediction accuracy than BPLT and GALT by 10 ~ 11% for the holdout data. It is worth giving attention to the fact that there is a shade of difference of prediction accuracy between the training data and the holdout data for GAFD. There is, however, a wide difference between the training data and the holdout data for the other two models. These results may be caused by the fact that the globally searched discretization simplifies the learning process and eliminates the irrelevant patterns. This prevents the network from falling into the problem of overfitting and may enhance the generalizability.

From Table 5, we also find that the average prediction performances of BPLT and GALT are similar. Although Sexton et al. (1998b) concluded that ANN with the

Table 3  
Summary statistics

Name of feature	Max	Min	Mean	Standard Deviation
Stochastic %K	100.007	0.000	45.407	33.637
Stochastic %D	100.000	0.000	45.409	28.518
Stochastic slow %D	99.370	0.423	45.397	26.505
Momentum	102.900	−108.780	−0.458	21.317
ROC	119.337	81.992	99.994	3.449
LW %R	100.000	−0.107	54.593	33.637
A/D Oscillator	3.730	−0.157	0.447	0.334
Disparity 5 days	110.003	90.077	99.974	1.866
Disparity 10 days	115.682	87.959	99.949	2.682
OSCP	5.975	−7.461	−0.052	1.330
CCI	226.273	−221.448	−5.945	80.731
RSI	100.000	0.000	47.598	29.531

Table 4  
Number of cases

Set	Year										Total
	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	
Training	232	233	234	236	237	237	235	235	234	234	2,347
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

genetically evolved connection weights outperforms conventional backpropagation neural networks with the gradient descent algorithm, this study does not find evidence to support their conclusions. The reasons for this result may be summarized in two points. First is that there is a generic limitation of global search algorithms. Sexton et al. (1999) reported that performance with the connection weights searched by simulated annealing, one of global search algorithms, was lower than the performance of backpropagation with the gradient descent algorithm. Their result was also supported by Shin et al. (1998). They concluded that the reason of the results came from the fact that GA may be less competent in a local search. Although a global search is more desirable than a local search for learning ANN, sometimes a local search is also needed. The other factor may be a complex dimensionality in data. GA is a global search algorithm, however, financial data including the stock market data is too complex to be searched easily. It is necessary to reduce the dimensionality of data and irrelevant factors before searching.

The McNemar tests are used to examine whether GAFD significantly outperforms the other two models. This test is a nonparametric test for two related samples. This test may be used with nominal data and is particularly useful with before–after measurement of the same subjects (Cooper & Emory, 1995). Table 6 shows the results of the McNemar test to compare the performance for the holdout data.

As shown in Table 6, GAFD performs better than the other two models at a 1% statistical significance level. In

addition, Table 6 shows that BPLT and GALT do not significantly outperform each other.

## 6. Concluding remarks

As mentioned earlier, previous studies tried to optimize the controlling parameters of ANN using global search algorithms. Some of them only focused on the optimization of the connection weights of ANN. Others had an interest in the optimization of the learning algorithms itself, but most studies had little interest in the dimensionality reduction and the elimination of irrelevant patterns. This paper has

Table 5  
Average predictive performance (hit ratio: %)

Year	BPLT		GALT		GAFD	
	Training	Holdout	Training	Holdout	Training	Holdout
1989	59.05	48.28	57.33	49.12	68.10	59.65
1990	62.23	49.15	59.23	56.90	66.95	60.34
1991	58.97	53.45	53.42	50.00	63.25	56.90
1992	61.02	51.72	60.17	44.83	66.95	58.62
1993	54.01	44.07	54.43	44.07	67.09	61.02
1994	62.45	64.41	61.18	59.32	63.29	62.71
1995	63.83	44.83	63.83	53.45	69.36	65.52
1996	61.28	60.35	61.70	50.00	64.26	67.24
1997	46.15	50.00	50.43	50.00	64.10	62.07
1998	55.98	51.72	56.84	48.28	64.53	62.07
Total	58.50%	51.81%	57.86%	50.60%	65.79%	61.70%

Table 6

McNemar values (*P* values) for the pairwise comparison of performance between models

	GALT	GAFD
BPLT	0.227 (0.634)	15.593* (0.000)
GALT		19.267* (0.000)

\* Significant at the 1% level.

proposed a new hybrid GA and ANN to mitigate the above limitations. In this paper, GA not only searches for the optimal or near-optimal solutions of connection weights in the learning algorithm but also looks for the optimal or near-optimal thresholds of feature discretization for the dimensionality reduction. GAFD discretizes the original continuous data according to the GA-derived thresholds and simultaneously assigns the genetically evolved connection weights. We conclude that GAFD reduces the dimensionality of the feature space then enhances the generalizability of the classifier from the empirical results.

This study has some limitations. First, the number of processing elements in the hidden layer is fixed at 12. This is the same as the number of input features. However, the performance of the model may vary with the number of processing elements in the hidden layer. The second limitation is that the objects for optimization are focused on only two factors of the learning process of ANN. GAFD produces valid results in this study. However, GA can potentially be used to simultaneously optimize several factors of the learning process including feature subset selection, network structure optimization, and learning parameter optimization. We also believe that there is great potential for further research with feature discretization using GA for other AI techniques including case-based reasoning and decision trees.

## Acknowledgements

The authors would like to thank Korea Science and Engineering Foundation for supporting this work under Grant No. 98-0102-08-01-3.

## 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, CA.
- 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.
- Bigus, J. P. (1996). *Data mining with neural networks*, New York: McGraw-Hill.
- Buhlmann, P. (1998). Extreme events from the return-volume process: a discretization approach for complexity reduction. *Applied Financial Economics*, 8 (3), 267–278.
- 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, IL: Irwin.
- Dash, M., & Liu, H. (1997). Feature selection methods for classifications. *Intelligent Data Analysis—An International Journal*, 1 (3), 131–156.
- Davis, L. (1994). Genetic algorithms and financial applications. In G. J. Deboeck, *Trading on the edge* (pp. 133–147). New York: Wiley.
- Dorsey, R., & Sexton, R. (1998). The use of parsimonious neural networks for forecasting financial time series. *Journal of Computational Intelligence in Finance*, 6 (1), 24–31.
- Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. *Proceedings of the 12th International Conference on Machine Learning* (pp. 194–202). San Francisco, CA.
- Duke, L. S., & Long, J. A. (1993). *Neural network futures trading—a feasibility study*, Society for Worldwide Interbank Financial Telecommunications, *Adaptive intelligent systems*. (pp. 121–132). Amsterdam: Elsevier.
- Edwards, R. D., & Magee, J. (1997). *Technical analysis of stock trends*, Chicago, IL: 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). Chambéry, France.
- Gifford, E. (1995). *Investor's guide to technical analysis: predicting price action in the markets*, London: Pitman.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*, Reading, MA: Addison-Wesley.
- Gupta, J. N. D., & Sexton, R. S. (1999). Comparing backpropagation with a genetic algorithm for neural network training. *Omega*, 27 (6), 679–684.
- Hiemstra, Y. (1995). Modeling structured nonlinear knowledge to predict stock market returns. In R. R. Trippi, *Chaos and nonlinear dynamics in the financial markets: theory, evidence and applications* (pp. 163–175). Chicago, IL: Irwin.
- Ignizio, J. P., & Soltys, R. (1996). Simultaneous design and training of ontogenic neural network classifiers. *Computers and Operations Research*, 23 (6), 535–546.
- 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, CA.
- 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.
- 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). Bari, Italy.
- Lawrence, S., Tsoi, A. C., & Giles, C. L. (1996). Noisy time series prediction using symbolic representation and recurrent neural network grammatical inference. Institute for Advanced computer Studies,

- Technical report UMIACS-TR-96-27 and CS-TR-3625. University of Maryland.
- Liu, H., & Motoda, H. (1998a). *Feature selection for knowledge discovery and data mining*, Norwell, MA: Kluwer Academic.
- Liu, H., & Motoda, H. (1998b). Feature transformation and subset selection. *IEEE Intelligent Systems and Their Applications*, 13 (2), 26–28.
- 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 (3/4), 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.
- 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.
- Sexton, R. S., Alidaee, B., Dorsey, R. E., & Johnson, J. D. (1998a). Global optimization for artificial neural networks: a tabu search application. *European Journal of Operational Research*, 106 (2/3), 570–584.
- Sexton, R. S., Dorsey, R. E., & Johnson, J. D. (1998b). Toward global optimization of neural networks: a comparison of the genetic algorithm and backpropagation. *Decision Support Systems*, 22 (2), 171–185.
- Sexton, R. S., Dorsey, R. E., & Johnson, J. D. (1999). Optimization of neural networks: a comparative analysis of the genetic algorithm and simulated annealing. *European Journal of Operational Research*, 114 (3), 589–601.
- Shin, K., Shin, T., & Han, I. (1998). Neuro-genetic approach for bankruptcy prediction: a comparison to back-propagation algorithms. *Proceedings of the International Conference of the Korea Society of Management Information Systems* (pp. 585–597). Seoul, South Korea.
- Slowinski, R., & Zopounidis, C. (1995). Application of the rough set approach to evaluation of bankruptcy risk. *International Journal of Intelligent Systems in Accounting, Finance and Management*, 4 (1), 27–41.
- 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 (1–5), 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.
- Vafaie, H., & De Jong, K. (1998). Feature space transformation using genetic algorithms. *IEEE Intelligent Systems and Their Applications*, 13 (2), 57–65.
- Wong, F., & 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). Hawaii.