Neural Comput & Applic (2004) 13: 255–260
DOI 10.1007/s00521-004-0428-x

ORIGINAL ARTICLE

Kyoung-jae Kim · Won Boo Lee

Stock market prediction using artificial neural networks with optimal feature transformation

Received: 5 June 2001 / Accepted: 13 May 2004 / Published online: 25 June 2004

Abstract This paper compares a feature transformation method using a genetic algorithm (GA) with two conventional methods for artificial neural networks (ANNs). In this study, the GA is incorporated to improve the learning and generalizability of ANNs for stock market prediction. Daily predictions are conducted and prediction accuracy is measured. In this study, three feature transformation methods for ANNs are compared. Comparison of the results achieved by a feature transformation method using the GA to the other two feature transformation methods shows that the performance of the proposed model is better. Experimental results show that the proposed approach reduces the dimensionality of the feature space and decreases irrelevant factors for stock market prediction.

Keywords Feature transformation · Genetic algorithms · Fuzzification · Artificial neural networks · Stock market prediction

1 Introduction

Recently, artificial neural networks (ANNs) have been regularly applied to the research area of finance, such as stock market prediction, bankruptcy prediction, and corporate bond rating. In these applications, ANNs try to learn the pattern of financial data. When data are loaded into the ANN, they must be preprocessed from their numeric range into the numeric range that the ANN deals with efficiently. In this stage, proper transformation of data simplifies the process of learning and may improve the generalizability of the learned results.

K. Kim (✉) · W. B. Lee
Department of Information Systems,
Dongguk University, 3-26 Pil-Dong,
Chung-Gu, Seoul, 100-715, Korea
E-mail: kjaekim@hanmail.net
Tel.: +82-2-22603324
Fax: +82-2-22603684

Prior research often used linear scaling as a data preprocessing method for ANNs. The goal of linear scaling is to independently normalize each feature component to the [0, 1] range. It is usually employed to enhance the performance of the ANN because most ANNs accepts numeric data only in the range of [0.0, 1.0] or [−1.0, +1.0] [1]. It also ensures that the larger value input features do not overwhelm smaller value input features, which then helps to reduce prediction errors. It is a simplistic method of data preprocessing and does not consider the association among independent and dependent features. The prediction performance, however, is enhanced through the ability of discrimination not only by a single feature, but also by the association among features.

On the other hand, one of popular preprocessing methods is feature transformation. Feature transformation is the process of creating a new set of features [2]. Although various methods of feature transformation had been proposed, there were few comparative studies on their performance. This paper proposes a hybrid model of ANNs and genetic algorithms (GAs) for optimal feature transformation. The proposed model is used to predict the future direction of change in the Korea composite stock price index (KOSPI). In addition, this study compares this model with conventional feature transformation methods.

The rest of the paper is organized as follows. The next section reviews prior research on stock market prediction using ANNs. Section 3 describes basic concept of the GA. Section 4 proposes a feature transformation method using the GA and describes the benefits of the proposed model. Section 5 describes the research data and experiments. In Sect. 6, the experimental results are summarized and discussed. In Sect. 7, conclusions and future research issues are presented.

2 Research background

Many studies on stock market prediction using ANNs were performed during the past decade. One of the earliest studies, Kimoto et al. [3] used modular neural networks for the prediction of the Tokyo stock exchange prices index (TOPIX). Kamijo and Tanigawa [4] used recurrent neural networks and Ahmadi [5] employed backpropagation neural networks with the generalized delta rule to predict the stock market. Yoon and Swales [6] also performed predictions using qualitative and quantitative data. Trippi and DeSieno [7] and Choi et al. [8] predicted daily direction of change in the S&P 500 index futures using ANNs. These studies are mainly focused on the application of ANNs to stock market prediction.

Recent research tends to hybridize several artificial intelligence (AI) techniques to improve the prediction performance. Tsaih et al. [9] integrated the rule-based technique and ANNs 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 in the learning process. Kohara et al. [10] incorporated prior knowledge to improve the learning process of the conventional ANN.

However, some of the above studies ignored the tremendous noise and non-stationary characteristics in stock market data. Lawrence et al. [11] pointed out that, when the training of a ANN tends to be difficult due to the noise of data, then the networks fall into a naive solution such as always predicting the most common output. The data preparation process is needed to reduce noise and complex dimensionality in data.

Most of the above studies relied on the gradient descent algorithm to get weights of connections in ANNs. However, this may force the solutions of the ANN to the local minimum. Sexton et al. [12] pointed out the fact that the gradient descent algorithm may perform poorly even on simple problems when predicting the holdout data. Sexton et al. [13] also proposed that the use of 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 for the weight vector of network. In this study, the GA will be used for the step of assigning connection weights between the layers to mitigate the above limitation.

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 mechanisms of *selection*, *crossover*, and *mutation* [14]. The first step of the GA is problem representation. The problems must be represented as a suitable form to be handled by the GA. If the problems are coded as chromosomes, the populations are initialized. Each chromosome within the population is gradually evolved by biological operations. Once the population size is chosen, the initial population is randomly generated [15]. 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 with unfit individuals [16].

The GA works with three operators that are iteratively used. The *selection* operator determines which individuals may survive [17]. 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. In addition, the *mutation* operator arbitrarily alters one or more components of a selected chromosome. It provides the means for introducing new information into the population. Finally, the GA tends to converge on optimal or near-optimal solutions [18].

The GA is usually employed to improve the performance of AI techniques. For ANNs, the GA was applied to the selection of neural network topology, including optimizing relevant feature subset, determining the optimal number of hidden layers, and processing elements.

4 Feature transformation for ANNs

Artificial neural networks offer pre-eminent learning ability, but it often results in inconsistent and unpredictable performance for the prediction of noisy financial data. Especially, 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 [19]. Thus, reduction and transformation of the irrelevant or redundant features shortens the running time and yields more generalized results [20]. One of the most popular preprocessing methods is feature transformation. The methods of transformation are classified as endogenous vs. exogenous [21–23]. Endogenous methods do not take into consideration the value of the decision attribute while exogenous methods do [23].

Endogenous transformation has the advantage of simplicity in the transformation process, but it does not consider the association among each independent and dependent feature. The prediction performance, however, 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 transformation [22]. This study compares an exogenous transformation method with two endogenous transformation methods.

4.1 Linear transformation

Many classical feature transformation methods perform a linear transformation of the original feature vectors [24]. In general, linear transformation means the linear scaling of data into the specified numeric range. Linear transformation is often used to enhance the performance of ANNs because most ANN models accept numeric data only in the ranges [0.0, 1.0] or [−1.0, +1.0] [1]. In this study, linear transformation denotes the linear scaling of data into the range [0.0, 1.0]. This study names ANNs with linear transformation as the linear transformation model (LTM). The LTM is classified as an endogenous method because this method does not take into consideration the value of the decision attribute.

4.2 Fuzzification

It may be necessary to transform numerical data into symbolic data to reduce the dimensionality. Fuzzification is an ideal approach because fuzzy membership functions map numerical data to symbolic data with fuzzy degree [25]. Although there are many methods for the construction of fuzzy membership functions, this study employs the clustering algorithm proposed by Klimasauskas [26].

The Klimasauskas [26] method is composed of two steps. The first step uses the k means clustering algorithm to group numerical data into k clusters, and then transfers the clusters to membership functions. The k means algorithm returns a crisp set of size k. The second step is the fuzzification step. The fuzzy set is returned by transformation of this crisp set. This study assumes that the fuzzy degree of the boundaries for each cluster is 0.5, and the maximum fuzzy degree, which is 1, is at the mean of each cluster. The process of fuzzification step in this study is presented as follows:

$$C = 0.5 \times (S + E)$$
$$K = 2 \times (1 - M)/(E - S)$$
$$F_i = \max\,(0.1 - K \times |X_i - C|)$$

where M is the value of the fuzzy membership function at the boundaries, S is the leftmost boundary, E is the rightmost boundary, C is the center between the boundary, K is the scale factor, X_i is the feature value, and F_i is the fuzzy membership value.

This approximation allows two adjacent membership functions to intersect at a 0.5 degree of membership. Fuzzification in this study is classified as an endogenous method. This study names ANNs with fuzzification as the fuzzy transformation model (FTM).

4.3 The genetic algorithms approach

Feature transformation may be split into three categories; feature extraction, feature construction, and feature discretization. Among them, feature discretization is closely related to dimensionality reduction [27]. Discretization needs relevant and rational discretizing thresholds. However, the thresholds may vary, depending on the securities being analyzed and the overall market condition [28]. For this reason, there may be no general guidelines with which to discretize.

However, we may search the thresholds for discretizing continuous measures into qualitative norms to capture the domain-specific knowledge. Although some studies suggested various methods of discretizing features, this paper proposes the optimization of discretizing thresholds based on the GA. It may find optimal or near-optimal thresholds of discretization for maximum predictive performance because the GA searches for the parameter to maximize a specified fitness function. This study names ANNs with discretization using the GA approach as the GA-based feature transformation model (GTM). The GTM is classified as an exogenous transformation method.

5 Research data and experiments

This study compares GTM with LTM and FTM. In addition, this study tests the statistical significance of the difference. The research data used in this study consists of technical indicators and the direction of change in the daily KOSPI. The total number of samples come from 2,348 trading days, from January 1991 to December 1998. For the selection of a relevant feature subset, this study performs the pairwise t-test ($P < 0.05$). Then, this study uses five domain experts to review the final feature subset. Finally, selected features are stochastic %K, stochastic %D, stochastic slow %D, momentum, rate of change (ROC), Larry William's %R (LW %R), accumulation/distribution oscillator (A/D oscillator), disparity 5 days, disparity 10 days, price oscillator, commodity channel index (CCI), and relative strength index (RSI). Among the data, about 20% are used for holdout and 80% for training. The number of cases in each set is shown in Table 1.

Searching the thresholds for feature transformation proceeds in the following ways.

First, the GA searches for optimal or near-optimal connection weights and thresholds for feature transformation. The parameters for searching must be encoded on chromosomes as decision variables. The encoded connection weights and discretizing thresholds are searched to maximize the fitness function of the GA. The number of discretizing thresholds for each feature is fixed to 2 because the experts in the stock market usually interpret continuous values in qualitative terms, such as low, medium, and high. The fitness function is specific to applications. In this study, the objective of the model is to train the network and to approximate the connection weights and the thresholds for transformation for

Table 1 Number of cases

Set	1991	1992	1993	1994	1995	1996	1997	1998	Total
Training	234	236	237	237	235	235	234	234	1,882
Holdout	58	58	59	59	58	58	58	58	466

Table 2 Average predictive performance (hit ratio: %)

Year	LTM (%)		FTM (%)		GTM (%)	
	Training	Holdout	Training	Holdout	Training	Holdout
1991	53.42	50.00	58.97	58.62	60.26	63.79
1992	60.17	44.83	60.17	55.17	62.29	56.90
1993	54.43	44.07	56.54	61.02	59.07	62.71
1994	61.18	59.32	50.63	61.02	64.98	66.10
1995	63.83	53.45	66.81	53.45	65.96	67.24
1996	61.70	50.00	59.15	62.07	65.11	68.97
1997	50.43	50.00	61.11	55.17	62.82	63.79
1998	56.84	48.28	60.68	58.62	60.68	63.79
Total	57.76	50.00	59.24	58.15	62.65	64.16

correct solutions. This objective can be represented by the average prediction accuracy of the training data. The parameters are searched using only the information about training data.

The GA operates the process of crossover and mutation on initial chromosomes and iterates until the stopping conditions are satisfied at this stage. For the controlling parameters of the GA search, the population size is set to 100 organisms and the crossover and mutation rates are changed to prevent the ANN from falling into a local minimum. The range of crossover rate is set between 0.5 and 0.7, while the mutation rate ranges from 0.05 to 0.1 in this study. 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. 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 5,000 generations are permitted. The range of thresholds for feature transformation is permitted between the maximum and minimum value of each feature.

The second stage is the process of feed-forward computation in ANNs. In this process, the sigmoid function is used as the activation function. This is a popular function for backpropagation neural networks because it can easily be differentiated. The linear function is used as the combination function for the feed-forward computation with derived connection weight from the first stage. Finally, the derived connection weights and the thresholds of feature transformation are applied to the holdout data.

6 Experimental results

This study compares GTM with LTM and FTM. The number of processing elements in the hidden layer is fixed at 12. This is like the number of features subset. Table 2 describes the average prediction accuracy of each model.

In Table 2, GTM performs better than FTM and LTM by 3.41% and 4.89%, respectively, for the training data. FTM outperforms LTM by 1.48% for the training data. In addition, GTM has a higher prediction accuracy than LTM by 14.16% for the holdout data. GTM also outperforms FTM by 6.01% 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 GTM. In addition, FTM has a higher prediction accuracy than LTM by 8.15% for the holdout data.

Table 3 McNemar values for the pairwise comparison of performance between models

	FTM	GTM
LTM	6.845[a]	30.616[a]
FTM		3.919[b]

[a]Significant at a 1% level
[b]Significant at a 5% level

The McNemar tests are used to examine whether GTM significantly outperforms LTM and FTM. This test is executed for the pairwise comparison of the performance between models. The McNemar values of the holdout data are presented in Table 3.

In Table 3, GTM performs better than LTM at a 1% statistical significance level and outperforms FTM at a 5% level for the holdout data. In addition, FTM significantly outperforms LTM at a 1% significance level.

It appears that GTM and FTM allows the ANN to better learn noisy patterns than LTM. The reason may be that the features in modeling ANNs contribute the value of network outputs not only individually, but also in a synergistic way. The connection weight in the ANN reflects the importance of specific connections. It is computed by taking into consideration the value of dependent features and the association of other independent features. However, it may not reflect the pure embedded knowledge of each feature. The pure embedded knowledge of each feature may be reflected by a qualitative norm.

Table 4 shows the optimized thresholds for the discretization of GTM and the thresholds for interpretation of human experts in the stock market. Each threshold is used as a criteria for discretizing the original continuous data.

From Table 4, we find the fact that the optimized thresholds for the feature transformation is approximate to some domain knowledge in the stock market. The

Table 4 The thresholds for interpretation of experts and average discretization thresholds for GTM

	Murphy [29]	Achelis [28]	Choi [30]	Chang et al. [31]	Edwards and Magee [32]	This study
Stochastic %K	30/70	20/80	20/80	25/75	20–25/75–80	30.84/67.04
Stochastic %D	30/70	20/80	20/80	25/75	20–25/75–80	36.29/66.18
Stochastic slow %D	30/70	20/80	20/80	25/75	20–25/75–80	29.43/65.69
Momentum	0			0		−14.53/16.71
ROC	100		100			98.24/103.86
LW %R	20/80	20/80	20/80	10/90		31.02/81.29
A/D oscillator				0.5 or 0.2/0.8		0.32/0.73
Disparity 5 days			100			98.69/102.88
Disparity 10 days			100			98.20/102.35
CCI	0 or −100/+100	−100/+100	0 or −100/+100	0 or −100/+100		−44.05/111.47
Price oscillator		0	0			−0.80/0.46
RSI	30/70	30/70	30/70	30/70	20–30/70–80	31.19/74.15

majority of thresholds for the feature transformation in Table 4 are similar to the domain knowledge of the stock market, except for “Momentum” and “CCI”.

7 Conclusions

This paper presents the GA-based feature transformation model (GTM) for ANNs to predict the patterns of the stock market. The GTM discretizes the original continuous data according to optimal or near-optimal thresholds. We may conclude that the GTM reduces the dimensionality of the feature space and then enhances the generalizability of the classifier from the empirical results. In addition, the fuzzy transformation model (FTM) also significantly outperforms the linear transformation model (LTM).

This study has some limitations. First, the number of categories for the discretization is limited to three. This number is varied with the nature of each feature. This study limits the categories because the computational burden of unlimited categories is too heavy to be efficiently executed by a personal computer. The second limitation is that the objects for optimization are focused on only two factors of the learning process of the ANN. The GTM produces valid results, however, the GA can potentially be used to 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 the GTM for other AI techniques, including case-based reasoning and decision tree.

Acknowledgements This work is supported by the Dongguk University Research Fund.

References

1. Bigus JP (1996) Data mining with neural networks: solving business problems from application development to decision support. McGraw-Hill, New York
2. Liu H, Motoda H (1998) Feature transformation and subset selection. IEEE Intell Syst Appl 13(2):26–28
3. Kimoto T, Asakawa K, Yoda M, Takeoka M (1990) Stock market prediction system with modular neural network. In: Proceedings of the international joint conference on neural networks (IJCNN), Washington, DC, June 1990, pp 1–6
4. Kamijo K, Tanigawa T (1990) Stock price pattern recognition: a recurrent neural network approach. In: Proceedings of the international joint conference on neural networks (IJCNN), Washington, DC, June 1990, pp 215–221
5. Ahmadi H (1990) Testability of the arbitrage pricing theory by neural networks. In: Proceedings of the international joint conference on neural networks (IJCNN), Washington, DC, June 1990, pp 385–393
6. Yoon Y, Swales G (1991) Predicting stock price performance: a neural network approach. In: Proceedings of the 24th annual Hawaii international conference on system sciences (HICSS-24), Kauai, Hawaii, January 1991, pp 156–162
7. Trippi RR, DeSieno D (1992) Trading equity index futures with a neural network. J Portfolio Manag 19:27–33
8. Choi JH, Lee MK, Rhee MW (1995) Trading S&P 500 stock index futures using a neural network. In: Freedman RS (ed) Proceedings of the 3rd annual international conference on artificial intelligence applications on Wall Street, New York, June 1995, pp 63–72
9. Tsaih R, Hsu Y, Lai CC (1998) Forecasting S&P 500 stock index futures with a hybrid AI system. Decis Support Syst 23(2):161–174
10. Kohara K, Ishikawa T, Fukuhara Y, Nakamura Y (1997) Stock price prediction using prior knowledge and neural networks. Int J Intell Syst Accounting Finance Manage 6(1):11–22
11. Lawrence S, Tsoi AC, Giles CL (1996) Noisy time series prediction using symbolic representation and recurrent neural network grammatical inference. Technical report UMIACS-TR-96–27 and CS-TR-3625, Institute for Advanced Computer Studies, University of Maryland
12. Sexton RS, Alidaee B, Dorsey RE, Johnson JD (1998) Global optimization for artificial neural networks: a tabu search application. Eur J Operational Res 106(2–3):570–584
13. Sexton RS, Dorsey RE, Johnson JD (1998) Toward global optimization of neural networks: a comparison of the genetic algorithm and backpropagation. Decis Support Syst 22(2):171–185
14. Adeli H, Hung S (1995) Machine learning: neural networks, genetic algorithms, and fuzzy systems. Wiley, New York
15. Bauer RJ (1994) Genetic algorithms and investment strategies. Wiley, New York
16. Davis L (1994) Genetic algorithms and financial applications. In: Deboeck GJ (ed) Trading on the edge. Wiley, New York, pp 133–147
17. Hertz A, Kobler D (2000) A framework for the description of evolutionary algorithms. Eur J Operational Res 126(1):1–12

18. Wong F, Tan C (1994) Hybrid neural, genetic, and fuzzy systems. In: Deboeck GJ (ed) Trading on the edge. Wiley, New York, pp 243–261
19. Liu H, Setiono R (1996) Dimensionality reduction via discretization. Knowl Based Syst 9(1):67–72
20. Dash M, Liu H (1997) Feature selection methods for classifications. Intell Data Anal Int J 1(3):131–156
21. Dougherty J, Kohavi R, Sahami M (1995) Supervised and unsupervised discretization of continuous features. In: Prieditis A, Russell S (eds) Proceedings of the 12th international conference on machine learning (ICML), Tahoe City, California, July 1995. Morgan Kaufmann, San Francisco, pp 194–202
22. Scott PD, Williams KM, Ho KM (1997) Forming categories in exploratory data analysis and data mining. In: Liu X, Cohen PR, Berthold MR (1997) Proceedings of the 2nd international symposium on advances in intelligent data analysis (IDA'97), London, UK, August 1997. Springer, Berlin Heidelberg New York, pp 235–246
23. Susmaga R (1997) Analyzing discretizations of continuous attributes given a monotonic discrimination function. Intell Data Anal Int J 1(3):157–179
24. Raymer ML, Punch WF, Goodman ED, Kuhn LA, Jain AK (2000) Dimensionality reduction using genetic algorithms. IEEE Trans Evol Comput 4(2):164–171
25. Chou ST, Hsu H, Yang C, Lai F (1997) A stock selection DSS combining AI and technical analysis. Ann Operations Res 75:335–353
26. Klimasauskas CC (1992) Hybrid fuzzy encodings for improved backpropagation performance. Advanced technology for developers, September 1992, pp 13–16
27. Liu H, Motoda H (1998) Feature selection for knowledge discovery and data mining. Kluwer, Norwell, Massachusetts
28. Achelis SB (1995) Technical analysis from A to Z. Probus Publishing, Chicago, Illinois
29. Murphy JJ (1986) Technical analysis of the futures markets: a comprehensive guide to trading methods and applications. New York Institute of Finance, Prentice-Hall, New York
30. Choi J (1995) Technical indicator. Jinritamgu Publishing, Seoul, Korea
31. Chang J, Jung Y, Yeon K, Jun J, Shin D, Kim H (1996) Technical indicators and analysis methods. Jinritamgu Publishing, Seoul, Korea
32. Edwards RD, Magee J (1997) Technical analysis of stock trends, 7th edn. Amacom, New York