Data Mining using Instance Selection in Artificial Neural Networks for Bankruptcy Prediction

Kyoung-jae Kim
Department of Information Systems, Dongguk University
(kjkim@dongguk.edu)

Corporate financial distress and bankruptcy prediction is one of the major application areas of artificial neural networks (ANNs) in finance and management. ANNs have showed high prediction performance in this area, but sometimes are confronted with inconsistent and unpredictable performance for noisy data. In addition, it may not be possible to train ANN or the training task cannot be effectively carried out without data reduction when the amount of data is so large because training the large data set needs much processing time and additional costs of collecting data. Instance selection is one of popular methods for dimensionality reduction and is directly related to data reduction. Although some researchers have addressed the need for instance selection in instance-based learning algorithms, there is little research on instance selection for ANN. This study proposes a genetic algorithm (GA) approach to instance selection in ANN for bankruptcy prediction. In this study, we use ANN supported by the GA to optimize the connection weights between layers and select relevant instances. It is expected that the globally evolved weights mitigate the well-known limitations of gradient descent algorithm of backpropagation algorithm. In addition, genetically selected instances will shorten the learning time and enhance prediction performance. This study will compare the proposed model with other major data mining techniques. Experimental results show that the GA approach is a promising method for instance selection in ANN.

Key Words : Bankruptcy prediction, Instance selection, Genetic algorithms, Artificial neural networks

Received: June 2004 Accepted June 2004 Corresponding Author: Kyoung-jae Kim

1. Introduction

Predicting corporate financial distress and bankruptcy is one of the most important problems in finance and management. Thus, there has been substantial research into the prediction of bankruptcies. The early days of these studies focused on application of statistical methods such as discriminant analysis and logistic regression. In addition, artificial intelligence (AI) techniques

* The authors are grateful to Giri Tayi and Seungik Baek for constructive comments.

such as decision tree, k-nearest neighbor, and case-based reasoning also have been applied to this area. Among these AI techniques, artificial neural network (ANN) is one of the most popular techniques for the prediction of corporate bankruptcy.

In general, ANN produces robust performance for the area which has a large amount of data such as corporate bankruptcy prediction. However, ANN often exhibits inconsistent and unpredictable performance on noisy data. In addition, it may not be possible to train ANN or the training task cannot be effectively carried out without data reduction when the amount of data is so large because training the large data set needs much processing time and additional costs of collecting data. Data reduction can be achieved in many ways such as feature selection or feature discretization (Blum & Langley, 1997; Liu & Motoda, 1998; Kim & Han, 2000).

One facet of data mining concerns the selection of relevant instances for this reason. Instances are a collection of training examples in supervised learning and instance selection chooses a part of the data that is representative and relevant to the characteristics of all the data. Instance selection is one of popular methods for dimensionality reduction and is directly related to data reduction. Although instance selection is the most complex form of data reduction because the computationally expensive prediction methods must be invoked more often to determine the effectiveness of instance selection, we can usually remove irrelevant instances as well as noise and redundant data (Weiss & Indurkhya, 1998; Liu & Motoda, 2001).

For some applications, a mining quality is continue to improve with additional instances. However, the number of instances tends to increase the complexity of induced solution. Increase complexity is not desirable, but may be the price to pay for better performance. In addition, increased complexity decreases the interpretability of the result (Weiss & Indurkhya, 1998). In this sense, many researchers have suggested instance selection methods.

Many researchers have suggested instance selection methods such as squashed data, critical points, prototype construction, in addition to many forms of sampling (Liu & Motoda, 2001). The efforts to select relevant instances from an initial data set have stemmed from the need to reduce immense storage requirements and computational loads (Kuncheva, 1995). The other perspective on this subject, as pointed out in Dasarathy (1990), is to achieve enhanced performance from the learning algorithm through instance selection. In addition, training time may be shortened by use of the proper instance selection algorithm.

This paper proposes a new hybrid model of ANN and genetic algorithms (GAs) for instance selection. An evolutionary instance selection algorithm reduces the dimensionality of data and may eliminate noisy and irrelevant instances. In addition, this study simultaneously searches the connection weights between layers in ANN through an evolutionary search. The genetically evolved connection weights mitigate the well-known limitations of gradient descent algorithm.

The rest of this paper is organized as follows: The next section presents the research background. Section 3 proposes the evolutionary instance selection algorithm and describes the benefits of the proposed algorithm. Section 4 describes the application of the proposed algorithm. In the final section, conclusions and the limitations of this study are presented.

2. Research background

2.1 Prior research on bankruptcy prediction using AI techniques

As mentioned earlier, there has been substantial research into the bankruptcy prediction because it is one of the most important problems for companies and financial institutions. Various techniques including ANN, decision tree, logistic regression, and discriminant analysis have been employed to predict corporate bankruptcy prediction.

Early studies by Altman (1968) and Deakin (1972) used discriminant analysis to predict corporate bankruptcies. More recent research by Ohlson (1980) used logit and probit models to predict bankruptcies. In addition, several studies in the past used artificial intelligence techniques to predict financial distress. In one of the earliest studies, Odom and Sharda (1990) and Tam and Kiang (1992) introduced ANNs for predicting corporate bankruptcies. Following these studies, a number of studies further investigated the use of ANNs in financial distress prediction. <Table 1> summarizes literature, methodological issues for using ANNs to predict corporate bankruptcies.

<Table 1> Prior research on the prediction of corporate bankruptcies

Reference	ANN model	Benchmark models
Tam and Kiang, 1992	BPN	DA, LR, k-NN, ID3
Martin-del-Brio and Serrano-Cinca, 1993	SOM	N/A
Serrano-Cinca, 1996	SOM	N/A
Serrano-Cinca, 1997	BPN	DA, LR
Altman et al., 1994	BPN	DA
Wilson and Sharda, 1994	BPN	DA
Boritz and Kennedy, 1995	BPN	DA, Logit ,Probit
Boritz et al., 1995	BPN	DA, k-NN, Logit, Probit
Lee et al., 1996	BPN	No
Jo and Han, 1996	BPN	DA, k-NN
Lee et al., 1996	BPN	LR, DA
Jo et al., 1997	BPN	DA, k-NN
Kiviluoto, 1998	SOM, RBF-SOM, LVQ	DA, k-NN
Yang et al. 1999	PNN, BPN	DA

BPN: Backpropagation neural networks, SOM: Self-organizing map, RBF: Radial basis function, LVQ: Learning vector quantization, DA: Discriminant analysis, LR: Logistic regression, k-NN: k-nearest neighbor

In <Table 1>, the authors of prior research mainly tested the feasibility of ANN to bankruptcy prediction. However, it may not be possible to train ANN or the training task cannot be effectively carried out without data reduction when the amount of data is so large because training the large data set needs much processing time and additional costs of collecting data. In this respect, data reduction may be needed in future research on bankruptcy prediction. This study introduces the concept of instance selection, one of the most popular techniques of data reduction, for corporate bankruptcy prediction.

2.2 Instance selection methods

Instance-based learning algorithms often faced the problem of deciding which instances to store for use during generalization in order to avoid excessive storage and time complexity, and to improve generalizability by avoiding noise and overfitting (Wilson & Martinez, 2000). Many researchers have addressed the problem of training data reduction and have presented algorithms for maintaining an instance base or case base in instance-based learning algorithms.

Instance selection in instance-based learning algorithms may be considered as a method of knowledge refinement and it maintains the instance-base. In this sense, some researchers proposed many instance selection algorithms for maintaining the case-base in case-based reasoning (CBR) systems. Smyth (1998) presented an approach to maintenance which is based on the deletion of harmful and redundant cases from the case-base. In addition, McSherry (2000) suggested an instance selection method in the construction of a case library in which evaluation of the coverage contributions of candidate instances are based on an algorithm called disCover. This algorithm reverses the direction of CBR to discover all cases that can be solved with a given case-base.

Although many different approaches have been used to address the problem of case authoring and data explosion for instance-based algorithms, there is little research on instance selection in ANN. Reeves & Taylor (1998) suggested that a GA is a promising approach to finding 'better' training data set for classification problems in radial basis function (RBF) nets. Reeves & Bush (2001) reported that the GA can also be used effectively to find a smaller subset of a 'good' training set in RBF nets for both classification and regression problems. Although, the GA has been shown to be a promising instance selection method for RBF nets, its performances on other neural network models are untested.

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 mechanisms of selection, crossover, and mutation (Adeli & Hung, 1995). The GA simulates the mechanics of population genetics by maintaining a population of knowledge structure which is made to evolve (Odetayo, 1995).

The problems must be represented in a suitable form to be handled by the GA. The GA often works with a form of binary coding. If the problems are coded as chromosomes, the population is initialized. Each chromosome within the population is gradually evolved by biological operations. 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 unfit individuals (Davis, 1994).

The GA works with three operators that are iteratively used. The selection operator determines which individuals may survive (Hertz & Kobler, 2000). 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 (Wong & Tan, 1994).

The GA is usually employed to improve the performance of artificial intelligence techniques. For ANN, the GA was applied to the selection of neural network topology including optimizing a relevant feature subset, determining the optimal number of hidden layers and processing elements. In addition, some researchers searched the connection weights of ANN using the GA instead of local search algorithms including a gradient descent algorithm. They suggested that global search techniques including the GA might prevent ANN from falling into a local optimum (Sexton et al., 1998; Gupta & Sexton, 1999; Kim & Han, 2000).

3. A GA approach to instance selection for ANN

As mentioned earlier, there are many studies on instance selection for the instance-based learning algorithm. However, there are few studies on instance selection for ANN. Thus, there are few relevant theories concerning instance selection for ANN. This paper proposes the GA approach to instance selection for ANN (GAANNIS). In this study, the GA supports the simultaneous optimization of connection weights and selection of relevant instances.

The procedure of GAANNIS consists of the following three phases: GA search phase, feed-forward computation phase, and validation phase.

GA search phase: In the GA search phase, the GA searches the search space to find optimal or near-optimal connection weights and relevant instances for ANN. The populations, the connection weights and the codes for instance selection, 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. The third set represents the codes for instance selection. The strings have the following encoding: each processing element in the hidden layer receives signals from the input layer. The first set of bits represents the connection weights between the input layer and the hidden layer. Each processing element in the output layer receives signals from the hidden layer. The next set of bits indicates the connection weights between the hidden layer and the output layer. The following bits are instance selection codes for the training data. The parameters to be searched use only the information about the selected instances within the training data. In this phase, the GA operates the process of crossover and mutation on initial chromosomes and iterates until the stopping conditions are satisfied.

Feed-forward computation phase: This phase is the process of feed-forward computation in ANN. Proper activation function is required to facilitate the learning process. However, there are no clear criteria regarding which activation function to use. Some researchers recommended the sigmoid function for classification problems and the hyperbolic tangent function for forecasting problems because of the difference between the sigmoid and the hyperbolic tangent function for the value range of delta weights with the SSE error function (Coakley & Brown, 2000). In addition, the majority of back-propagation applications used the sigmoid activation function (Hansen et al., 1999). There are few comparative studies between the sigmoid function and other activation functions in ANN. This study uses the sigmoid function as the activation function because this study is performed to classify the accurate direction of change in the daily stock price index. The linear function is used as a combination function for the feed-forward computation with the derived connection weights from the first phase.

Validation phase: The derived connection weights are applied to the holdout data. This phase is indispensable to validate the generalizability because ANN has the eminent ability of learning the known data. In addition, this phase is very important in this study because learning with selected instances may cause the problem of overfitting. If this phase is not carried out, the model may fall into the problem of overfitting with the selected instances within the training data.

4. Comparative analysis on corporate bankruptcy prediction

This section compares five competitive models including GAANNIS, hybrid GA & ANN model without instance selection(GAANN), backpropagation neural networks(BPN), case-based reasoning(CBR), and logistic regression(LR).

4.1 Application data

The application data used in this study consists of financial ratios and the status of bankrupt or non-bankrupt for corresponding corporate. The data was collected from one of largest commercial banks in Korea. The sample of bankrupt companies was 1335 companies in heavy industry which filed for bankruptcy between 1996 and 2000. The non-bankrupt companies were 1335 ones in heavy industry which filed between 1999 and 2000. Thus, the total number of samples is 2670 companies.

The financial status for each company is categorized as "0" or "1" and it is used as a dependent variable. "0" means that the corporate is bankrupt, and "1" means that the corporate is solvent. For independent variables, we first generate 164 financial ratios from the financial statement from each company. Finally, we get 15 financial ratios as independent variables through the two independent sample t-test and the forward selection procedure based on logistic regression. <Table 2> gives selected features and some statistics from outputs of descriptive statistics and logistic regression analysis.

4.2 Experiments

The following experiments are carried out:

Whole training data. Four models including logistic regression(LR), case-based reasoning (CBR), backpropagation neural networks(BPN)

<Table 2> Selected features and their statistics

Name of feature	Range	Mean	Standard deviation	Wald	Sig.
Financial expenses to liabilities	17.021	7.082	3.551	37.373	0.000
Cost of sales to net sales	51.000	81.879	8.167	4.876	0.027
Net worth to total assets	107.192	24.183	16.821	58.059	0.000
Financial expenses growth	0.259	-0.007	0.035	4.289	0.038
Payables turnover	191.440	13.373	21.388	13.219	0.000
Solvency ratios	294.118	38.553	36.620	4.087	0.043
Window coefficient	12.454	0.964	1.547	5.666	0.017
Financial expenses & normal profit to total assets	156.723	17.054	23.104	178.444	0.000
Cash flow to total liabilities	3.021	0.094	0.323	16.127	0.000
Total asset change ratios	111.592	20.390	20.739	54.823	0.000
Financial expenses growth rate to assets	16.961	0.262	2.845	12.665	0.000
Non-operating expenses growth rate to assets	24.991	-0.080	3.903	28.431	0.000
Cost of sales X Cost of sales growth ratio	1172.787	142.358	152.350	30.616	0.000
Inventories growth rate to sales	51.987	1.597	7.146	9.660	0.002
Total assets turnover X Sales growth rate	12.352	1.991	1.837	18.751	0.000

and hybrid GA & ANN model without instance selection(GAANN) use whole training data because these models have not the mechanism to select relevant instances.

Hybrid GA & ANN model with Selected instances(GAANNIS). Experiments on corporate bankruptcy data are implemented using GAANNIS. The procedure of the experiment is as follows. The GA searches for optimal or near-optimal connection weights and relevant instances for ANN. As mentioned earlier, this study needs three sets of parameters: The connection weights between the input and the hidden layer, the connection weights between the hidden and the output layer, and the codes for instance selection.

This study uses the following encoding for the strings: 15 input features are used and 15 processing elements in the hidden layer are employed. Each processing element in the hidden layer receives 15 signals from the input layer. The first 225 bits represent the connection weights between the input layer and the hidden layer. These bits are searched from -5 to 5. There is only one output processing element in the output layer. The processing element in the output layer receives signals from the hidden layer. Thus, the next 15 bits indicate the connection weights between the hidden layer and the output layer. These bits also varied between -5 and 5. The following bits are instance selection codes for the training data. The chromosome of these bits consists of n genes (where n is the number of initial training instances), each one with two possible states: 0 or 1. "1" means the associated instance is selected into the analysis and "0" means the associated instance is not chosen.

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 to select relevant instances for the correct solutions. These objectives can be represented by the average prediction accuracy of the selected instances within the training data. Thus, this study applies the average prediction accuracy of the selected instances within the training data to the fitness function. Mathematically, the fitness function is represented as equation (1):

$$Fitness = \frac{1}{n}\sum_{i=1}^{n} CR_i \quad (i = 1, 2, \ldots n) \tag{1}$$

$$\begin{cases} \text{if } PO_i = AO_i & CR_i = 1 \\ \text{otherwise} & CR_i = 0 \end{cases}$$

where CR_i is the prediction result for the i th trading day which is denoted by 0 or 1, PO_i is the predicted output from the model for the i th trading day, and AO_i is the actual output for the i th trading day.

For the controlling parameters of the GA search, the population size is set at 100 organisms and the crossover and mutation rates are varied to prevent ANN from falling into a local minimum. The value of the crossover rate is set at 0.7 while the mutation rate is 0.1. For the crossover method, 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 variables. Thus, this study performs crossover using the uniform crossover routine. For the mutation method, this study generates a random number between 0 and 1 for each of the variables in the organism. If a variable gets a number that is less than or equal to the mutation rate, then that variable is mutated. As the stopping condition, 2000 generations are permitted.

4.3 Experimental results and discussions

This study compares GAANNIS to the conventional ANN with the GA. The conventional ANN with the GA, named GAANN, denotes the ANN model with the connection weights which are determined by the GA. This model does not use the gradient descent algorithm but uses the GA to determine the connection weights between layers. However, this model analyzes all available training data to learn.

On the other hand, GAANNIS also uses the GA to determine the connection weights, but learns the patterns of the stock market data from the selected instances through an evolutionary search. For the GAANN model, about 20% of the data is used for holdout and 60% for training and 20% for testing. The training data is used to search for 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.

In addition, this study compares the performance of GAANNIS with major data mining techniques including case-based reasoning, logistic regression and backpropagation neural net. The number of the training instances in these models and the number of the selected instances within the training instances in GAANNIS are presented in <Table 3>.

<Table 4> describes the average prediction accuracy of each model.

<Table 3> Number of instances in each model

	LR	CBR	BPN	GAANN	GAANNIS
Training	2,135	2,135	1,601	1,601	822
Test			534	534	534
Holdout	535	535	535	535	535

<Table 4> Average predictive performance (hit ratio: %)

	LR	CBR	BPN	GAANN	GAANNIS
Training	80.14		80.70	81.20	83.46
Test	81.27		82.77	87.45	87.45
Holdout	79.44	80.75	81.68	82.06	84.67

In <Table 4>, GAANNIS has higher accuracy than the other models by 2.61-5.23% for the holdout data. GAANNIS also outperforms the other models by 2.26-3.42% for the training data. This result may be caused by the benefits of the instance selection through evolutionary search techniques. In addition, GAANN outperforms the other models except GAANNIS by 0.38-2.61% for the holdout data. The results show that the hybrid GA and ANN models offer viable alternative approaches.

The McNemar tests are used to examine whether GAANNIS significantly outperforms the other models. This test may be used with nominal data and is particularly useful with a before-after measurement of the same subjects (Cooper & Emory, 1995). <Table 5> shows the results of the McNemar test to compare the performances of five models for the holdout data.

As Shown in <Table 5>, GAANNIS produces better performance than LR, CBR, and GAANN with statistical significance. In addition, GAANN outperforms LR with the 1% statistical significance level. However, GAANNIS does not statistically outperform BPN. The other models do not significantly outperform each other.

In addition, the two-sample test for proportions is performed. 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 & Soni, 1991). <Table 6> shows P values for the pairwise comparison of performance between models.

As shown in <Table 6>, GAANNIS outperforms the other models except GAANN with statistical significance. The other models do not significantly outperform each other.

<Table 5> McNemar values for the holdout data

	CBR	BPN	GAANN	GAANNIS
LR	0.356	1.061	2.7926*	9.592***
CBR		0.137	0.396	4.494**
BPN			0.011	2.557
GAANN				3.250*

* significant at the 10% level, ** significant at the 5% level, *** significant at the 1% level

<Table 6> *P* values for the holdout data

	CBR	BPN	GAANN	GAANNIS
LR	0.2810	0.1788	0.1401	0.0129
CBR		0.3483	0.2912	0.0455
BPN			0.4404	0.0967
GAANN				0.1271

5. Concluding remarks

Prior 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 placed little emphasis on the optimization of the learning algorithm itself, but most studies focused little on instance selection for ANN. In this paper, we use the GA for ANN in two ways. We first use the GA to determine the connection weights between layers. This may mitigate the well-known limitations of the gradient descent algorithm. In addition, we adopt the evolutionary instance selection algorithm for ANN. This directly removes irrelevant and redundant instances from the training data. We conclude that GA-based learning and the instance selection algorithm (GAANNIS) significantly outperforms the conventional GA-based learning algorithm (GAANN) and the other typical classification models.

The prediction performance may be more enhanced if the GA is employed not only for instance selection but also for relevant feature selection, and this remains a very interesting topic for further study. Although instance selection is a direct method of noise and dimensionality reduction, feature selection effectively reduces the dimensions of feature space. Of course, there are still many tasks to be done for the proposed model. The generalizability of the model should be further tested by applying it to other problem domains.

References

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

[2] Altman. E.I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy, *The Journal of Finance*, **September**, 589-609.

[3] Altman, E.I., G. Macro and F. Varetto (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks, *Journal of Banking and Finance*, **18**, 505-529.

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

[5] Blum, A. and P. Langley (1997). Selection of relevant features and examples in machine learning, *Artificial Intelligence*, **97** (1-2), 245-271.

[6] Boritz, J.E. and D.B. Kennedy (1995). Effectiveness of neural network types for prediction of business failure. *Expert Systems with Applications*, **9** (4), 503-512.

[7] Boritz, J.E., D.B. Kennedy and A. de Miranda e Albuquerque (1995). Predicting corporate failure using a neural network approach, *Intelligent Systems in Accounting, Finance and Management*, **4**, 95-111.

[8] Coakley, J.R. and C.E. Brown (2000). Artificial neural networks in accounting and finance: Modeling issues, *International Journal of Intelligent Systems in Accounting, Finance and Management*, **9**, 119-144.

[9] Cooper, D.R. and C.W. Emory (1995). *Business research methods*, Chicago: Irwin.

[10] Dasarathy, B.V. (1990). *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, CA: IEEE Computer Society Press.

[11] Davis, L. (1994). Genetic algorithms and

financial applications, in *Trading on the edge*, G.J. Deboeck (ed), New York: Wiley, 133-147.

[12] Deakin, E. (1974). A discriminant analysis of predictors of business failure, *Journal of Accounting Research*, **Spring**, 167-179.

[13] Gupta, J.N.D. and R.S. Sexton (1999). Comparing backpropagation with a genetic algorithm for neural network training, *Omega*, **27** (6), 679-684.

[14] Harnett, D.R. and A.K. Soni (1991). *Statistical methods for business and economics*, Reading: Addison-Wesley.

[15] Hansen, J.V., J.B. McDonald and R.D. Nelson (1999). Time series prediction with genetic-algorithm designed neural networks: An empirical comparison with modern statistical models, *Computational Intelligence*, **15** (3), 171-184.

[16] Hertz, A. and D. Kobler (2000). A framework for the description of evolutionary algorithms, *European Journal of Operational Research*, **126** (1), 1-12.

[17] Jo, H. and I. Han (1996). Integration of case-based forecasting, neural network and discriminant analysis for bankruptcy prediction, *Expert Systems with Applications*, **11** (4), 415-422.

[18] Jo, H., I. Han and H. Lee (1997). Bankruptcy prediction using case-based reasoning, neural network and discriminant analysis, *Expert Systems with Applications*, **13** (2), 97-108.

[19] Kim, K. and I. Han (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.

[20] Kiviluoto, K. (1998). Predicting bankruptcies with the self-organizing map, *Neurocomputing*, **21**(1-3), 203-224.

[21] Kuncheva, L.I. (1995). Editing for the k-nearest neighbors rule by a genetic algorithm, *Pattern Recognition Letters*, **16** (8), 809-814.

[22] Lee, K.C., I. Han and Y. Kwon (1996). Hybrid neural network models for bankruptcy predictions, *Decision Support Systems*, **18**, 63-72.

[23] Liu, H. and H. Motoda (1998). Feature transformation and subset selection, *IEEE Intelligent Systems and Their Applications*, **13** (2), 26-28.

[24] Liu, H. and H. Motoda (2001). Data reduction via instance selection, in *Instance selection and construction for data mining*, H. Liu and H. Motoda (eds), Massachusetts: Kluwer Academic Publishers, 3-20.

[25] Martin-del-Brio, B. and C. Serrano-Cinca (1993). Self-organizing neural networks for the analysis and representation of data: Some financial cases, *Neural Computing & Applications*, **1**, 193-206.

[26] McSherry, D. (2000). Automating case selection in the construction of a case library, *Knowledge-Based Systems*, **13** (2-3), 133-140.

[27] Odetayo, M.O. (1995). Knowledge acquisition and adaptation: a genetic approach, *Expert Systems*, **12** (1), 3-13.

[28] Odom. M. and R. Sharda (1990). A neural network model for bankruptcy prediction, in *Proceedings of the IEEE International Conference on Neural Networks*, 163-168.

[29] Ohlson, J. (1980). Financial ratios and the probabilistic prediction of bankruptcy, *Journal of Accounting Research*, **18** (1), 109-131.

[30] Reeves, C.R. and D.R. Bush (2001). Using genetic algorithms for training data selection in RBF networks, in *Instance selection and construction for data mining*, H. Liu and H. Motoda (eds), Massachusetts: Kluwer Academic Publishers, 339-356.

[31] Reeves, C.R. and S.J. Taylor (1998). Selection

of training sets for neural networks by a genetic algorithm, in *Parallel problem-solving from nature-PPSN V*, A.E. Eiden, T. Bäck, M. Schoenauer and H.-P. Schwefel (eds), Berlin: Springer-Verlag, 633-642.

[32] Serrano-Cinca, C. (1996) Self organizing neural networks for financial diagnosis, *Decision Support Systems*, **17**, 227-238.

[33] Serrano-Cinca, C. (1997). Feedforward neural networks in the classification of financial information, *The European Journal of Finance*, **3**(3), 183-202.

[34] Sexton, R.S., R.E. Dorsey and J.D. Johnson (1998). Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation, *Decision Support Systems*, **22** (2), 171-185.

[35] Smyth, B. (1998). Case-base maintenance, in *Proceedings of the 11th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems*, Castellon, Spain: Springer, 507-516.

[36] Tam, K.Y. and Kiang, M.Y. (1992). Managerial applications of the neural networks: *The case of bank failure predictions, Management Science*, **38**(7), 926-947.

[37] Weiss, S.M. and N. Indurkhya (1998). *Predictive data mining: A practical guide,* CA: Morgan Kaufmann Publishers.

[38] Wilson, D.R. and T.R. Martinez (2000). Reduction techniques for instance-based learning algorithms, *Machine Learning*, **38,** 257-286.

[39] Wilson, R.L. and R. Sharda (1994). Bankruptcy prediction using neural networks, *Decision Support Systems*, **11**, 545-557.

[40] Wong, F. and C. Tan, C. (1994). Hybrid neural, genetic, and fuzzy systems, in *Trading on the edge,* G.J. Deboeck (ed), New York: Wiley, 243-261.

[41] Yang, Z.R., M.B. Platt and H.D. Platt (1999). Probabilistic neural networks in bankruptcy prediction, *Journal of Business Research*, **44**, 67-74.