

Decision tree combined with neural networks for financial forecast

József Bozsik

Received 2012-07-09

Abstract

In this article I would like to introduce a hybrid adaptive method. There is wide range of financial forecasts. This method is focusing on the economical default forecast, but the method can be used generally for other financial forecasts as well, for example for calculating the Value at Risk.

This hybrid method is combined by two classical adaptive methods: the decision trees and the artificial neural networks. In this article I will show the structure of the hybrid method, the problems which occurred during the construction of the model and the solutions for the problems. I will show the results of the model and compare them with the results of another financial default forecast model. I will analyse the results and the reliability of the method and I will show how the parameters can influence the reliability of the results.

Keywords

decision tree · neural network · default forecast · financial default · discriminant analysis

1 Introduction

An important area of decision trees is the forecast. This method of artificial intelligence is used and applied widely. It is used mostly for classification problems, but this method shows really good results also for forecast problems [1]. Nowadays the methods can be improved also by hybrid methods. There are more well-known methods which are tried to be combined in a way of saving the advantages of different methods, multiplying the effects in order to construct a more effective method. In this model which I would like to introduce the ability of decision trees will be improved by artificial neural networks.

One of the well-known problems in the financial world is the default forecast. In the current financial situation this problem is very important, if we think about the financial crisis. For the default forecast in the economics the so called default models are used [2]. One of the most known models is the so called discriminant analysis [3]. In Hungary the default forecast has no long history, because the law for bankruptcy and liquidation process¹ was accepted only in 1991. The first Hungarian default model was developed in 1996 by Miklós Virág and Ottó Hajdú [2] based on financial data of 1990 and 1991 using discriminant analysis.

In the article I will shortly introduce the basics of decision trees and neural networks. These well-known structures can be used very well in case of classification and forecast problems. Now I will construct by the combination of the two methods one new model. Shortly I will introduce the well-known perceptron and the multi layer perceptron models and the ID3 algorithm which is used in decision trees. I will show the special points which are needed for the combination and I will show the combination's classical way. After that I will introduce in details the used hybrid model. I will show the abstract model of the new model and the problems which occurred during the construction of the model and the solutions, among others the treatment of over-teaching and the continuity valued attributes. For the treatment of problems I used on the one hand the classical methods from the literature, on the other hand using the speciality of the

¹1991. XLIX. law about the process of bankruptcy, liquidation and end-liquidation

József Bozsik

Department of Software Technology And Methodology, ELTE, H-1117, Pázmány Péter sétány 1/C, Hungary
e-mail: bozsik@inf.elte.hu

problem own methods and with the help of them tried to bridge the problems.

The new method was built and tested by using data of real companies. During the building of the model I took 17 financial ratios into account [2]. I compared the results with the results of the widely used economics default forecast model. This model is the before mentioned discriminance analysis model. During the introduction of the results I will show the basics of discriminance analysis. For the testing it is necessary to understand the discriminance model. I did the comparison of the models on data of 2009.

At the end of the article I will summarize the advantages and disadvantages of the model and the barriers of the model. I will show the reasons of the classification accuracy of the results and I will explain the barriers. I will introduce the key-steps of development opportunities and the next research opportunities and questions.

2 The model

In order to build the hybrid model I used the 17 financial ratios which were used by Miklós Virág and Ottó Hajdú in the discriminance analysis model. In order to understand the basics it is necessary to introduce shortly the discriminance analysis. Here I will show only the main points of the model.

2.1 Discriminance analysis

"The discriminance analysis with more variables analyses the distribution of more ratio in the same time and sets up a classification rule, which contains more weighted financial ratio (these are the independent variables of the model) and summarize them in only one discriminant value." [3] The most important criteria of choosing the financial ratios which will be used in the model is that the ratios should have low correlation to each other. Otherwise the added value for the classification will be low. It is worth to begin the construction of the model with a significant ratio and in each further step the less correlated but the second significant ratio should be involved. In the discriminance-function which is a linear combination the values of the financial ratios should be substituted which are calculated from the annual report of the individual companies. In order to be able to classify if a company is able to pay or not, we have to compare the values with the discriminance value.

The general formula of the discriminance function is defined in formula 1.

$$Z = w_1X_1 + w_2X_2 + \dots + w_nX_n \quad (1)$$

In the formula used signs are:

- Z - discriminance value
- w_i - discriminance weights
- X_i - independent variables (financial ratios)
- $i = 1, \dots, n$ where n means the number of financial ratios

The analysis of 2009 as well as the analysis from 1996 showed that the companies which are able to pay and which are not are different in the following financial ratios:

- X_1 - quick liquidity ratio²
- X_2 - cash flow / total debt
- X_3 - current assets³ / total assets⁴
- X_4 - cash flow / total assets

The order of the financial ratios reflects the discriminance power of the ratios, it means that the most discriminative ratio is the quick liquidity ratio after that the three other ratios. The discriminance function is made by the involvement of these ratios based on the data of 2009 [4]. The equation is defined in formula 2.

$$Z = 1.3173 X_1 + 1.5942 X_2 + 3.0453 X_3 + 0.5736 X_4 \quad (2)$$

The critical Z value is 2.17658, it means if we substitute the values of a company's financial ratios and the result is higher than 2.17658, the company will be classified by the function as solvent (it is able to pay), otherwise insolvent (it is not able to pay).

2.2 The result of the discriminance analysis for the test data

During the test I chose randomly 400 companies from the open database of the webpage of the Hungarian Ministry of Justice and Law Enforcement⁵. 200 companies were solvent the other 200 insolvent. I applied the data for the introduced discriminance function; the result is shown in the Table 1.

Tab. 1. The results of the discriminance analysis based on data of 2009.

Type of results	The results
Wrong classified solvent companies (pieces)	48
Wrong classified solvent companies (percent)	24%
Wrong classified insolvent companies (pieces)	36
Wrong classified insolvent companies (percent)	18%
Total (pieces)	84
Total (percent)	21%
Classification accuracy (percent)	79%

Our aim is to set up a default forecast model which gives higher accuracy than the model described above.

²This ratio shows if the company is able to pay immediately

³Current assets: inventories, receivables, cash and cash equivalents

⁴Total assets are those assets which are used within one year in the company

⁵<http://www.e-beszamolo.irm.hu/kereses-Default.aspx>

2.3 Decision tree (ID3 and C4.5 algorithms)

I would like to construct a more reliable and faster method for default forecasts than the discriminance analysis. It is obvious from the introduced method that the companies can be classified into two classes depending on the value of the attributes. The decision trees can be applied for the classification and forecast problems. It can be confirmed that the measurements can be inaccurate and may contain failures or there can be also problems by the measurements. In these cases the decision tree can be well applied. It means that I will use a type of inductive teaching, the decision trees. I constructed the first model by using the classical decision tree algorithm.

The ID3 algorithm is used for decision tree construction by using teaching 'samples' [5]. The algorithm is built from bottom to up, from the roots to the leaves. The basic idea of the algorithm is the following: chose the target attribute, which is the attribute which has the binary value. In this case it means the classification of two classes. I would like to determine this value; it will be the target function. In the next step I will determine the attribute which has the highest added value to the output value of the target function based on the samples. After this I can start the construction of the decision tree in the following way: the attribute which was found before will be the root of the tree, and the possible values of the attribute will be the edges. In every step of the construction of the tree we have to determine for the rest attributes which value how much will add to the value of the target function and the algorithm has to choose the best one.

So we need to be able to determine the information gain or the descriptive-power of the attribute for the target-attribute. The ID3 [6] sorts the attributes according to the descriptive power and constructs the tree by this. The descriptive power will be a statistical value, which is called entropy in the information theory.

Let be S a sample set and B a binary target-attribute, and S_+ a positive, S_- a negative target-attributed sample-set:

$$S = S_+ \cup S_- \quad (3)$$

$$\forall a \in S_+ : B(s) = true \quad (4)$$

$$\forall a \in S_- : B(s) = false \quad (5)$$

Furthermore by every calculation of entropy let $\log_0 0$ zero(0) by definition. The value of entropy can be calculated by the following formula 6:

$$Entropy(S) = - \left(\frac{|S_-|}{|S|} \log_2 \frac{|S_-|}{|S|} + \frac{|S_+|}{|S|} \log_2 \frac{|S_+|}{|S|} \right) \quad (6)$$

Here we can define the descriptive-power of an A attribute using the entropy by the following: Let n_A the number of possible values of A attribute, S_i the set sample which have the i . possible values A_i of A attributes:

$$\forall s \in S_i : A(s) = A_i \quad (7)$$

$$DescriptivePower(S, A) =$$

$$Entropy(S) - \sum_{i=1}^{n_A} \frac{|S_i|}{|S|} Entropy(S_i) \quad (8)$$

The first problem with the ID3 algorithm is that it is not able to handle the continuity variable attributes therefore I used a completed and modified version: the C4.5 algorithm. In case of C4.5 algorithm every continuity variable attributes can be substituted by a dynamically composed binary attribute. The A continues value attribute is substituted by an A_c binary attribute, where $A_c = true$, if $A < c$.

So I could compose the first model using the C4.5 algorithm. This decision tree could reach the following classification accuracy showed in the next table (Table 2).

These results are reassuring but unfortunately not enough. The researches and international literature suggest that the classification accuracy can be improved using neural networks. In the following part I will show in details the hybrid model.

Tab. 2. The classification accuracy of decision tree using C4.5 algorithm.

Type of results	Results of C4.5 algorithm
Wrong classified solvent companies (pieces)	64
Wrong classified solvent companies (percent)	32%
Wrong classified insolvent companies (pieces)	62
Wrong classified insolvent companies (percent)	31%
Total wrong classified (pieces)	126
Total wrong classified (percent)	31.5%
Classification accuracy (percent)	68.5%

2.4 Decision tree with neural network

The literature classifies the hybrid systems combined by decision trees and neural networks into two main categories. I used this classification also to the classification of my method.

Those solutions belong to the first group where the decision trees are used for construction the neural network's structure. In this case the key-point is to compose the decision tree and then based on the structure to compose the neural network. This means that the decision tree will be converted into the neural network. Sethi [7] composed a 3-layerd neural network using decision trees, where the neurons of the hidden layer were determined and constructed by the decision tree. Brent [8] did a complexity analysis for this method and suggested some development opportunities which are important for the practice and the sophistication. Park [9] shows further results of other complementary suggestions and results of research by linear decision tree. Golea and Marchand [10] propose a linearly separable neural-network decision tree architecture to learn a given but arbitrary Boolean function.

2.4.1 Brute force model

The model developed by me belongs to the second group according to this classification. The base of the model is the C4.5

algorithm, but I ordered a single layer perceptron model to the inner node of the tree. With this it was possible to substitute the calculation of information gain, it means the task was taken over locally in every node by one neural network. These neural networks have the same structure in case of every node, but they are individual objects in every node.

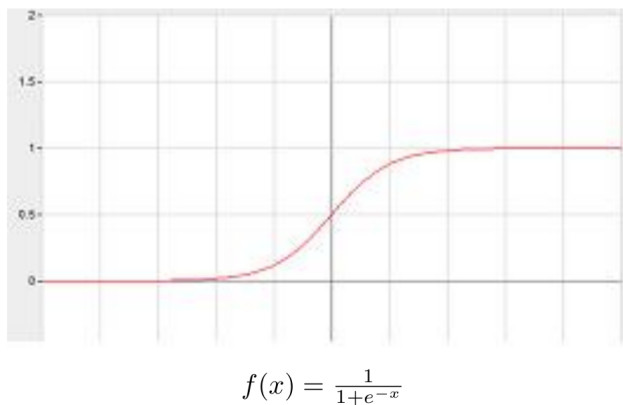


Fig. 1. The logistic activation function

According to the structure I used traditional neuron with logistic activation function (see Fig. 1). The structure of the network is homogeny structured perceptron (HSP) model as it is shown in the following figure (see Fig. 2). The number or attributes which are found in the input layer are in accordance with the layers of the tree in every case, but in every case there are two outputs. The two outputs are in accordance with the two classes, it means with the classes of solvent and insolvent companies.

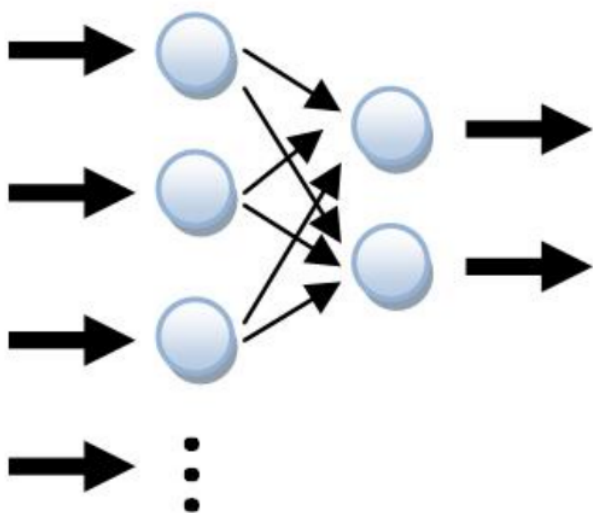


Fig. 2. Homogeny structured perceptron (HSP) model

The 17 financial ratios which are used for the financial forecast will be involved in the system as 17 continues variable attributes. As the financial ratios can be all in different intervals

of the set of real numbers, it is needed to transform all of them into the $[0, 1]$ interval.

$$\varphi : \mathbb{R} \mapsto [0, 1] \quad (9)$$

$$\forall i \in [1, 17] : A_i := \varphi(\tilde{A}_i) \quad (10)$$

The conversion is necessary in order to have the elements of the vector homogeny which are used as input attributes in the base set. Please note, that the conversion can be also implemented into the network as well. To achieve it an entity doing a simple transformation must be connected to the inputs of the neurons. This can be in practise a pre-processor. After the conversion there is no more need for further pre-processors. In the followings I will show in details the structure of the tree. In order to build a tree completed by perceptron model the following steps must be done:

- 1 The most discriminative attribute must be determined by the entropy. By this attribute the node of the tree root will be labelled.
- 2 An object of the introduced special perceptron model must be set up for the node of the root. This object must be initialized by random weights on the traditional way and it must be trained on a small test set. The smaller set must be chosen randomly from the teaching samples. The maximum number of the teaching cycles must be limited, but it is also possible to give an error barrier as a stop-condition.
- 3 If there is no more attributes, the building of the tree will be stopped. If there is left, than it must be continued by the Step4.
- 4 Two edges must be taken from the node and the child nodes must be fixed. The most discriminative attribute must be determined by the entropy. The new node must be labelled by this attribute.
- 5 For each child nodes an SLP model must be set up. The most discriminative attribute which was determined in the last stage of the tree must be deleted from the model. The new model must be initialized by random weights in the traditional way and it must be trained on a small test set. The smaller set must be chosen randomly from the teaching samples, this is how the homogeneity will be ensured. From that point the algorithm must go on from the 3rd step.

A part or this special decision tree will be shown in the next figure (see Fig. 3). In the example three attributes are present. From the three attributes the 2-indexed (A_2) is the most discriminative; it is shown by a darker neuron on the figure. There are two classes as output: solvent and insolvent. In the second step the 2-labelled (A_2) attribute must be deleted from the system. After that on the second stage of the decision tree the 1-indexed (A_1) attribute is the most discriminative. This is shown in the figure by a darker neuron.

I used the Backpropagation model [14] for the teaching. Let me introduce the following indicators:

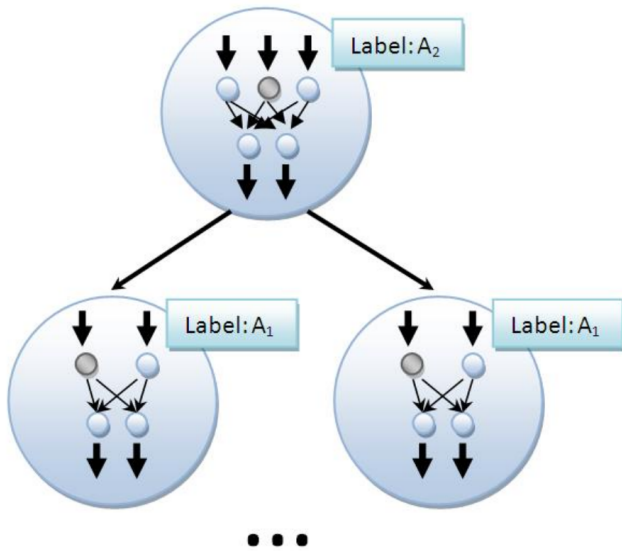


Fig. 3. Decision tree with perceptron model

- Let the input layer be indicated by 0. layer for the easier fitting
- Let me indicate the l -layer's i -ranked neuron's input value as: $a_0^{l-1}, \dots, a_{n_i}^{l-1}$
- Let me indicate the l -layer's i -ranked neuron's input weights as: $w_0^{l-1}, \dots, w_{n_i}^{l-1}$
- Let me indicate the l -layer's i -ranked neuron's output value as: a_i^l

Accordingly the steps of the error backpropagation algorithm which is used for the teaching are following [14]:

- 1 Calculate each neuron's output from the $x = [a_0^0, \dots, a_{n_0}^0]$ input vector by layers: a_i^l , after the calculation we receive the output of the output layers which i indicate by: a_{out} .
- 2 Calculate the errors for the output values:

$$e_{output} = a_{out}(1 - a_{out})(a_{expected} - a_{out}) \quad (11)$$

and the changes of the weights.

- 3 Calculate the error of the neurons by layers from back to the forth:

$$e_i^l = a_i^l(1 - a_i^l) \sum_{k=1}^{n^{l+1}} e_k^{l+1} w_k^{l+1,i} \quad (12)$$

and the changes of the weights where η means the learning-coefficient:

$$\Delta w_i^{l,j} = \eta e_i^l a_i^{l-1} \quad (13)$$

- 4 Modify the weights of the network:

$$w_i^{l,j} = w_i^{l,j} + \Delta w_i^{l,j} \quad (14)$$

Let these steps repeat until the input layer will be reached. With this step the weight-modifications will be validated for the

Tab. 3. The classification accuracy of the brute force model.

Type of results	The result of the new model
Wrong classified solvent companies (pieces)	30.6
Wrong classified solvent companies (percent)	15.3%
Wrong classified insolvent companies (pieces)	32.24
Wrong classified insolvent companies (percent)	16.12%
Total wrong classified (pieces)	62.84
Total wrong classified (percent)	15.71%
Classification accuracy (percent)	84.29%

Brute force Decision Tree Accuracy [%]

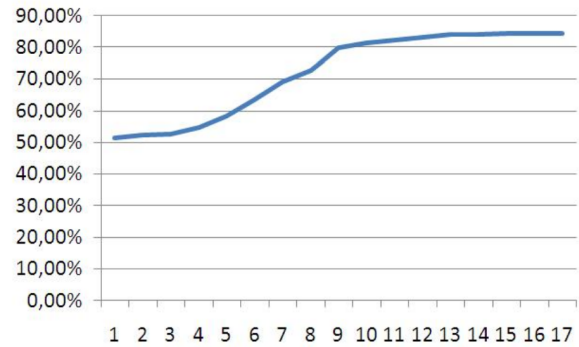


Fig. 4. The classification accuracy of the brute force model

whole network, it means the correction referring to the error is done for every layer and every neuron in the network.

Using this method very good classification accuracy could be achieved, the results are shown in the table. (see Table 3).

The results are better than it was expected, but it is not optimal according to the running time and the memory usage. The problem is obvious, because during the tree building at every stage the nodes will be doubled. This means that in case of 17 attributes the number of the levels of the tree is $2^{17} = 131072$. During the tree building layer by layer the classification accuracy and the teaching time will be measured on the partial tree. These measurements are shown in the following table (see Table 4) and in a graphical way on the Fig. 4.

2.4.2 Sophisticated model

It can be a very obvious simplification, that if in every stage the same structured neural networks take place, that all occurrences can be substituted by a neural network object. By this the running time can be made linear, but the classification accuracy decreases dramatically. By this simplification the classification accuracy showed similar trend like the model without the simplification, but the best classification accuracy was not even 75%.

Tab. 4. Measured data of the brute force model.

Level	Classification accuracy [%]	Runing time [ms]
1	51,46%	16486,11143
2	52,41%	32935,20584
3	52,69%	65813,60957
4	54,81%	131193,912
5	58,45%	263257,881
6	63,56%	525483,1841
7	69,21%	1056286,679
8	72,83%	2102539,921
9	79,96%	4212080,117
10	81,26%	8452559,106
11	82,23%	16907693,81
12	83,26%	33602052,96
13	84,01%	67581873,89
14	84,13%	135844148,7
15	84,24%	271444759,5
16	84,29%	540817535,7

That is why I did not use this simplification. The scheme of the logical structure is shown in the following figure (see Fig. 5).

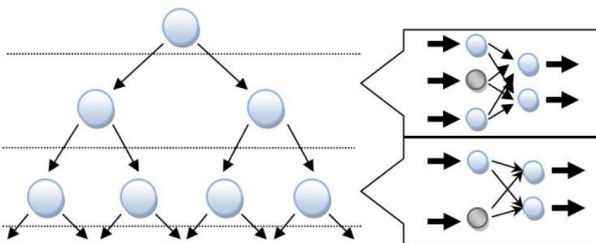


Fig. 5. A neural network object by layers

Unfortunately the model shows the typical signs of over-fit as well. In the next graph (Fig. 6) it is shown that in every stage of the tree building the accuracy and the running time was measured and compared. The results are shown in the following graph.

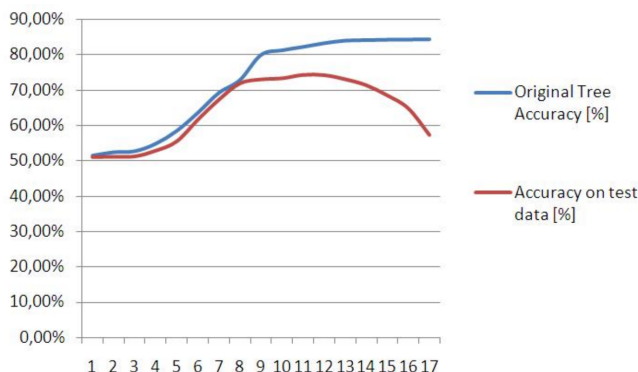


Fig. 6. Over-fitting of the brute force model.

There are two methods for the solution of the problem. The first one is the so called Chi-square test. The key point of the

method is that during the building of the tree it must be measured by a statistical method if the added new node is significant. The second method is the so called pruning, which means basically the pruning 'backwards' of the total constructed decision tree.

In order to optimize I used the mentioned pruning method, because it is better taking into account the speciality of the problem, than the Chi-square method. The main point of the reduced-error pruning is that the database must be separated by the Holdout method into teaching and testing sets and based on the teaching set the decision tree can be built. In this case the database contains data from 400 companies and it is separated to 250 teaching and 150 testing samples.

The pruning method contains the following steps:

- 1 We should start from the bottom of the tree to the top and measure every inner node, if eliminating the node and the partial tree, whether the error can be reduced on the test data.
- 2 If yes, than we have to eliminate the node and the belonging partial tree.
- 3 The process must be continued till the error can be reduced on the test data.

Using this method we could achieve a thinner structured tree. This structure has all the advantages that the earlier model had, but it can solve the problem in a more optimal way for the running time and the memory usage. The next graph (Fig. 7) shows the classification accuracy of the brute force and the sophisticated model by each layer.

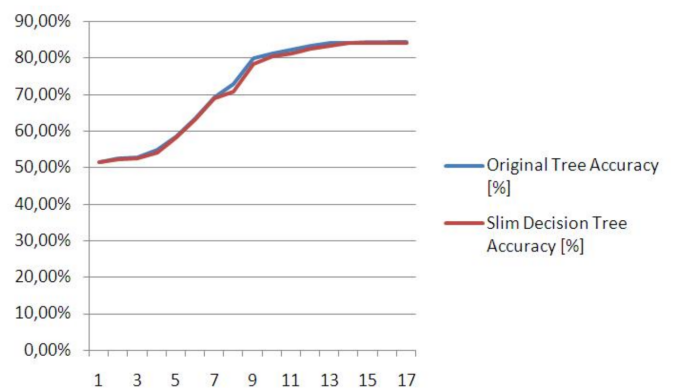


Fig. 7. Comparison of the Brute force model and the sophisticated model

The next table (Table 5) shows the classification accuracy of the sophisticated model using 17 financial ratios:

3 Summary

The results of the research show that the hybrid decision trees combined with neural networks can be applied for financial default forecasts as well. This area of the artificial intelligence can be also applied next to the classical economics models. It has to be pointed out that the final model combines more classical models in a new way, and it is important to note that only after modifications of the economical models according to the problem's

Tab. 5. The classification accuracy of the sophisticated model.

Type of results	Sophisticated's model results
Wrong classified solvent companies (pieces)	30.01
(percent)	15.05%
Wrong classified insolvent companies (pieces)	34.02
(percent)	17.01%
Total wrong classified (pieces)	64.12
(percent)	16.03%
Classification accuracy (percent)	83.97%

speciality could have been the 83.97% classification accuracy achieved.

This decision tree model built from 400 company data, the introduced construction and the pruning method tested on 150 company data was able to achieve higher classification accuracy ratio than the traditional discriminant analysis model. It should not be forgotten that the training and the test was measured only on a relative small sample database. Further research areas can build a sophisticated method from bigger sample-set and measure the classification accuracy by other parameters. Furthermore it is important to examine other structured models for example how to use sparse combination of basis functions [15] for solving this problem or how to adapt a complex-valued neural networks with independent component analysis [16] (ICA) for this financial forecast problem.

References

- 1 **Mitchell T M**, *Machine Learning*, McGraw-Hill, 1997.
- 2 **Virág M, Hajdú O**, *Pénzügyi mutatószámokon alapuló csődmódelleszámítások*, Bankszemle **XV** (1996), no. 5, 42–53.
- 3 **Kovács E**, *Pénzügyi adatok statisztikai elemzése*, Budapesti Corvinus Egyetem Pénzügyi és Számviteli Intézet, 2006.
- 4 **Kovács G**, *Diszkriminanciaanalízis 2009. évi adatok alapján*, Online Study (2009.10.08), <http://kozgazdasag.uw.hu/kovacsg/diszkrim2009/diszkrim2009.html>.
- 5 **Quinlan J R**, *Induction of decision trees*, *Machine Learning* **1** (1986), 81–106.
- 6 **Futó I**, *Mesterséges intelligencia*, AULA Kiadó Rt., 1999.
- 7 **Sethi I K**, *Entropy nets: From decision trees to neural networks*, *Proc. IEEE* **78** (1990), 1605–1613, DOI 10.1109/5.58346.
- 8 **Brent R P**, *Fast training algorithms for multilayer neural nets*, *IEEE Trans. Neural Networks* **2** (1991), 346–354, DOI 10.1109/72.97911.
- 9 **Park Y**, *A comparison of neural-net classifiers and linear tree classifiers: Their similarities and differences*, *Pattern Recognition* **27** (1994), 1493–1503, DOI 10.1016/0031-3203(94)90127-9.
- 10 **Golea M, Marchand M**, *A growth algorithm for neural-network decision trees*, *Europhys. Lett.* **12** (1990), 205–210, DOI 10.1209/0295-5075/12/3/003.
- 11 **Odom M D, Sharda R**, *A Neural Network Model For Bankruptcy Prediction*, *Proceeding of the International Joint Conference on Neural Networks*, Vol. II. IEEE Neural Networks Council, posted on 1990, 163–171, DOI 10.1109/IJCNN.1990.137710, (to appear in print).

- 12 **Tam K, Kiang M**, *Managerial applications of the neural networks: The case of bank failure predictions*, *Management Science* **38** (1992), no. 7, 416–430, DOI 10.1287/mnsc.38.7.926.
- 13 **Virág M, Kristóf T**, *Az első hazai csődmódelleszámítása neurális hálókkal segítségével*, *Közgazdasági Szemle* **LII.** (2005), 144–162.
- 14 **Gregorics T**, *Mesterséges intelligencia alapjai*, Electronic version of a university lecture (2010.01.15), <http://people.inf.elte.hu/gt/mi/neuron/neuron.pdf>.
- 15 **Kovács K, Kocsor A**, *Classification using a sparse combination of basis functions*, *Acta Cybernetica* **17** (2005), no. 2, 311–323.
- 16 **Szabó Z, Lőrincz A**, *Complex Independent Process Analysis*, *Acta Cybernetica* **19** (2009), no. 1, 177–190.