JOURNAL
OF
APPLIED
QUANTITATIVE
METHODS

APPLYING FUZZY C-MEANS AND ARTIFICIAL NEURAL NETWORKS FOR ANALYZING THE NON-BANKING FINANCIAL INSTITUTIONS' SECTOR IN ROMANIA

Adrian COSTEA
Ph.D., Academy of Economic Studies, Statistics and Econometrics Department

E-mail: adrian.costea@csie.ase.ro

ABSTRACT:

In this paper we apply a neural approach to develop classification models in order to assess the performance of non-banking financial institutions (NFIs) in Romania. Our objective is twofold: to empirically validate our methodology and understand how different financial factors can and do contribute to the NFIs' movements from one performance class to another.

Keywords: non-banking financial institutions, performance evaluation, neural networks, class prediction

1. INTRODUCTION

In this paper we analyze comparatively the financial performance of non-banking financial institutions (NFIs) in Romania using Data Mining (DM) techniques. The NFIs' financial performance is measured according to our previous work (Costea, 2011a) in terms of financial ratios that define three performance dimensions: capital adequacy, assets' quality and profitability. Our methodology consists of two stages. In the first stage we apply a DM clustering techniques, namely Fuzzy C-Means (FCM) algorithm, in order to find performance clusters within the data. According to Costea (2011b), FCM algorithm performed better than other clustering algorithms in terms of the formed clusters when applied on the same NFIs' performance dataset. At this stage we characterize each cluster in terms of average characteristics of the observations that are allocated in that particular cluster and attach to each observation a label (performance class variable) that identifies the observation as belonging to the cluster. In the second stage, we apply feed-forward artificial neural networks (ANNs) algorithms in order to map the input space to the newly created performance class variable so that we might be able to predict the performance of different NFIs as data become available. For a detailed technical explanation of the FCM algorithm we refer the reader to Costea (2011b). In the next Section, we present a brief explanation of the ANNs for classification.

2. ARTIFICIAL NEURAL NETWORK CLASSIFICATION ALGORITHMS

The generic classification model based on neural approaches is depicted in Figure 1 (adapted from Costea, 2005). As it can be seen from Figure 1, when building classification models, firstly, we perform some preliminary steps: we separate the data into training (*TR*) and test (*TS*) sets, we construct the performance class variable by applying a clustering method (in our case – FCM algorithm). Then, we determine the proper ANN architecture. This step consists of determining the proper number of hidden layers, and the appropriate number of neurons in each hidden layer. Also, we decide how to code the class variable: using one neuron or as many neurons as the number of performance classes (clusters). Finally, we train and test the ANN using different values for the parameters involved in training.

According to Basheer & Hajmeer (2000, p. 22), the choice of the number of hidden layers and the number of neurons in each hidden layer depend on "input/output vector sizes, size of training and test subsets, and, more importantly, the problem of non-linearity". Basheer & Hajmeer (2000) presents a list of papers that provide different rules of thumb regarding the correspondence between the number of hidden neurons (*NH*) and the number of input (*NI*) and output (*NO*) neurons or the number of training samples (N_{TRN}). For example, Lachtermacher & Fuller (1995) links the number of input and hidden neurons *(NI, NH)* for one output *ANN* with the number of training samples N_{TRN} with the formula: $0.11N_{TRN} < NH(NI+1) < 0.30N_{TRN}$. Upadhyaya & Eryurek (1992) connect the total number of weights N_w with the number of training samples with the formula: $N_w = N_{TRN} \log_2(N_{TRN})$. Masters (1994) calculates the number of hidden neurons in the hidden layer around the geometric mean of the number of inputs (*NI*) and of outputs (*NO*). Choosing these parameters is more art than science. We base our decision on the advice given by Basheer & Hajmeer's (2000, p. 23): "the most popular approach to finding the optimal number of hidden nodes is by trial and error with one of the above rules". For example, in this study we chose the Lachtermarcher & Fuller (1995) rule and varied *NH* depending on the size of the training set. Concerning the number of hidden layers, we performed in each case a number of experiments for ANN architectures with one and two hidden layers to see what the appropriate number of hidden layers is. Depending on the dataset used, an ANN with one or two hidden layers performed better in terms of the training mean square error. The three hidden layer cases was discarded to avoid the increase in network complexity given we obtain high training accuracy rates for less complex ones.

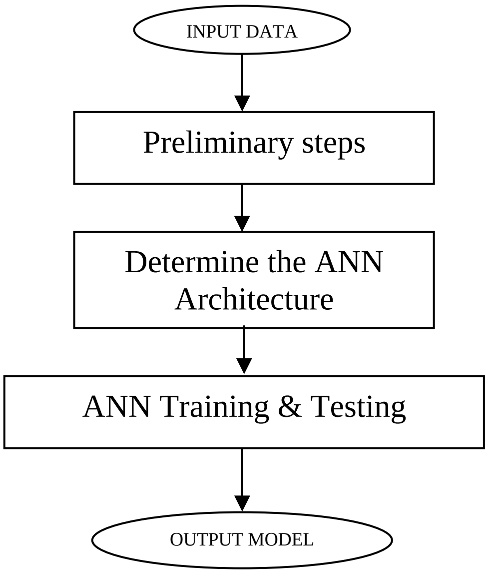

Figure 1. ANN generic classification model (Source: Costea, 2005)

We have used the sigmoid and linear activation functions for the hidden and output layers respectively, as this combination of activation functions provided the best results in our experiments. We base our decision in choosing the best training algorithm on the studies that have been written were different algorithms were compared in order to find the best algorithm for a particular problem (Demuth & Beale, 2001; Nastac & Koskivaara, 2003; Costea, 2003). In Costea (2003) we compared four training algorithms in terms of error rates and convergence speed. Our findings suggest that there is a negative correlation between error rates and the convergence speed. Therefore, in choosing the training algorithm, one should seek a compromise between these two factors. We observed that the Scaled Conjugate Gradient (SCG) algorithm (Moller, 1993) performs well over a wide variety of problems. The SCG is not the fastest algorithm, but it does not require large computational memory and it has a good convergence. Furthermore, in order to avoid the network over-fitting the training samples, we apply the validation stop method: we separate the training data in effective training (*TRe*) and validation (*VAL*) datasets and the training process stops when the difference between the effective training error and the validation error is greater than a small value given as a parameter. Moreover, it is well known that for validation stop, one must be careful not to use an algorithm that converges too rapidly (Hagan *et al*., 1996; Demuth & Beale, 2001). The SCG is well suited for the validation stop method.

3. THE NFIS FINANCIAL PERFORMANCE DATASET

In this application we use three performance dimensions to evaluate a NFI: capital adequacy (C), assets' quality (A) and profitability (P). We select different indicators for each dimension based on the analysis of the periodic financial statements of the NFIs. In the following table we present the indicators for each performance dimension.

Table 1. The performance dimension and the corresponding financial ratios

Dimension	Indicators
Capital adequacy	1. Equity ratio (Leverage) = own capital / total assets (net value) 2. Own capital / equity 3. Indebtedness sources = borrowings / own capital
Assets' quality	1. Loans granted to clients (net value) / total assets (net value) 2. Loan granted to clients (net value) / total borrowings 3. Past due and doubtful loans (net value) / total loans portfolio (net value) 4. Past due and doubtful claims (net value) / total assets (net value) 5. Past due and doubtful claims (net value) / own capital
Profitability	1. Return on assets (ROA) = net income / total assets (net value) 2. Return on equity (ROE) = net profit / own capital 3. The rate of profit = gross profit / total revenues 4. Activity cost = total costs / total revenues

The next step of the analysis is to choose the best set of indicators for each dimension and collect the data necessary to calculate these indicators. We have changed indicators number 3 for the "degree of capitalization" dimension and number 3 for the "profitability" dimension by replacing the denominator with Total Assets (net value). We have done this in order to be able to interpret the indicators since the former denominator (own capital) could take negative values. At the same time we have eliminated the indicator 5 for the "assets' quality" dimension for the same reason. Finally, we have 11 indicators: 3 for the degree of capitalization, 4 for assets' quality and 4 for profitability. The data were collected quarterly from 2007 to 2012 for the NFIs registered in the Special Register that have been active since the introduction of the regulatory framework for these institutions in Romania. In total there were 68 NFIs that met the above criteria and 990 observations. Out of these 990 observations, 5 observations were discarded due to lack of data for certain financial indicators.

4. EXPERIMENT

In this experiment we try to evaluate comparatively the performance of 68 Romanian NFIs registered in the Special Register that have been active since 2006, the first year when this sector has been regulated in Romania. This analysis can help the Supervision Department of the National Bank of Romania to allocate more efficiently its resources. Identifying poorly performing NFIs would support supervisors to concentrate on a smaller number of NFIs that face difficulties. Other authors have studied the sectoral dynamics of non-performing loans (e.g.: Moinescu & Codirlasu, 2012) having similar research goals.

As the Figure 1 shows the first step of the methodology consists of some preliminary steps. Our dataset that consist of 11x985 observations has been transformed by levelling the extreme values for each variables in the [-20, 20] interval. We have done this in order to avoid the algorithms' results being affected by these extreme values.

In the next step, we apply FCM algorithm in order to build cluster with similar performance. We chose 4 clusters as we have done with a version of the same dataset in our previous work (Costea, 201x). The other parameters of FCM were as follows: m = 1.5, no_of_iterations = 10000, the limit for the stopping criterion = 0.00001. After we run the FCM algorithm on the 11x985 dataset we obtained the following structure of the clusters: cluster 1 (95 observations), cluster 2 (770 observations), cluster 3 (59 observations), and cluster 4 (61 observations). Based on the clusterization we have constructed the class variable by associating to each observation the number of the cluster that the observation belongs to.

In order to have an uniform number of observations in each cluster to train the classification model we selected 59 observations (the number of observations in the smallest cluster) from each cluster, totalling 236 observations. Also, at this stage, we have split the data in training (*TR*) and testing (*TS*) sets by selecting one testing instance for every nine training instances. Thus, we obtained randomly 212 observations for training and the rest for testing (24 observations).

The next step of the methodology was to determine the proper architecture for the ANN-based classification model that maps the 11-dimensional input space to the newly constructed performance class variable. In our experiments regarding the application of ANNs for classification (performed using Matlab's Neural Networks toolbox) we have kept all parameters of the ANNs constant (the learning algorithm - SCG, the performance goal of the classifier, the maximum number of epochs), except the number of neurons in the hidden layers (*NH* when we had one hidden layer and NH_1, NH_2 when we had two hidden layers).

Next, we present the empirical procedure to determine the architecture for an ANN with two hidden layers. Firstly, we performed three trainings in order to find the best ANN architecture. For each training we have split further the training set (*TR*) in the effective training set (*TRe*) and the validation set (*VAL*), obtaining each time approximately 186 observations for effective training and 26 observations for validation (we have used validation stop method as stopping criterion). We followed the Lachtermarcher & Fuller (1995) rule and varied NH_1 and NH_2 from 5 to 8 and trained the network for each ANN architecture based on the effective training dataset. We saved the best ANN architecture in terms of mean squared error for the effective training dataset (MSE_{TRe}) and if the mean squared error based on the validation set (MSE_{VAL}) is less than $6/5*MSE_{TRe}$. This condition has been imposed in order to avoid saving ANN architectures for which the effective training and validation mean squared error are too far from each other. The final ANN architecture consisted of 8 neurons on the first hidden layer and 5 neurons on the second hidden layer.

Finally, at the last methodological step, we have trained the obtained ANN with the same-way generated effective training, validation and testing datasets and obtained the following accuracy rates: effective training dataset accuracy rate (ACR_{TRe}) = 100 percent, validation dataset accuracy rate (ACR_{VAL}) = 100 percent, total training dataset accuracy rate (ACR_{TR}) = 100 percent and testing dataset accuracy rate (ACR_{TS}) = 95.83 percent. The high values for the accuracy rates and the small difference between testing and training accuracy rates show that we obtained a very good classification model. Moreover the empirical procedure to find the best ANN architecture has been validated by the same high accuracy rates. Based on the chosen architecture we can test different values for the other ANN parameters and further improve the performance of the ANN-based classifiers.

5. CONCLUSIONS

In this study we have applied Data Mining to formalize the process of assessing comparatively the performance of non-banking financial institutions in Romania. We addressed this research problem by associating two Data Mining tasks: a clustering task by which we followed a description strategy showing what is the current situation of the NFIs' sector and a classification task used for creating a mapping between the performance class variable and the multidimensional input space.

For the clustering phase we employed a fuzzy logic algorithm called Fuzzy C-Means algorithm and identified four performance clusters. Based on the average characteristics of the input variables we characterized each individual cluster. For the classification phase we selected an even number of observation in each cluster to allow the classifier to learn the characteristics of each cluster. As classification technique we used feed-forward neural networks trained using variants of backpropagation algorithm (e.g.: the Scaled Conjugate Gradient algorithm).

A secondary goal of this study was to find a procedure to determine the proper neural network architecture for our particular research problem. We obtained very high training and testing accuracy rates and small differences between these rates. Compared with other classification models applied on the same dataset in our previous work, the neural network-based model is the best in terms of training and testing accuracy. However, the explanatory capabilities of the decision trees have to be taken into account in the process of choosing the best model.

ACKNOWLEDGEMENTS:

This work was supported from the European Social Fund through Sectoral Operational Programme Human Resources Development 2007-2013, project number POSDRU/89/1.5/S/59184 „Performance and excellence in postdoctoral research in Romanian economics science domain".

References

1. Basheer IA, Hajmeer M., **Artificial neural networks: fundamentals, computing, design, and application**. *Journal of Microbiological Methods* 43, 2000, 3-31.
2. Costea A., **Economic Performance Classification Using Neural Networks**. *Proceedings of the Sixth International Symposium on Economic Informatics*, I. Ivan. and I. Rosca (eds), Academy of Economic Studies Press, Bucharest, Romania, 2003.
3. Costea A., **Computational Intelligence Methods for Quantitative Data Mining**, Turku Centre for Computer Science (TUCS), PhD thesis no. 67.,2005
4. Costea A., **Prudential supervision of non-banking financial institutions: developments and prospects**. *Romanian Statistical Review* 7, 2011, 81-100.
5. Costea A., **A fuzzy-logic approach to assessing the performance of non-banking financial institutions in Romania**, *Proceedings of First International*

Conference on Financial Modeling – FM 2011, organizers: Society for Financial and Actuarial Mathematics, Sofia, Bulgaria, editor: P. Stoynov, publisher: PERUN-SPRINT Ltd., November 19-20, 2011, pp. 48-54, ISSN: 1314-5916.

6. Costea A., **Performance benchmarking of non-banking financial institutions by means of Self-Organising Map algorithm**, accepted for publication in *East-West Journal of Economics and Business*, Volume XVI, Number 1, 2013, ISSN 1108-2992. The paper has been presented to the 2nd international conference International Conference on International Business (ICIB 2011), organized by the Department of International and European Studies, University of Macedonia, Thessaloniki, Greece, European Center for the Development of Vocational Training, Thessaloniki, Greece, Faculty of Political Science, University of Messina, Messina, Italy and CRIISEA, University of Picardie, Amiens, France, 19-21 May 2011, Thessaloniki, Greece.

7. Demuth H, Beale M., **Neural Network Toolbox**. The MathWorks Inc, Natick Press: MA, USA, 2001.

8. Hagan MT, Demuth HB, Beale M., **Neural Network Design**. PWS Publishing Company, Boston, USA, 1996,

9. Lachtermacher G, Fuller JD., **Backpropagation in time series forecasting**. *Journal of Forecasting* 14, 1995, 381–393.

10. Masters T., ***Practical Neural Network Recipes in*** **C++.** Academic Press, Boston, MA, 1994,

11. Moinescu B, Codirlasu A., **Assessing the Sectoral Dynamics of Non-performing Loans: Signs from Financial and Real Economy**, *Theoretical and Applied Economics*, Volume XIX (2012), No. 2(567), pp. 69-80.

12. Moller MF., **A scaled conjugate gradient algorithm for fast supervised learning**. *Neural Networks* 6, 1993, 525-533.

13. Nastac I, Koskivaara E., **A Neural Network Model for Prediction: Architecture and Training Analysis**. *TUCS Technical Report* 521, April 2003. [Available at: http://www.tucs.fi/publications/insight.php?id=tNaKo03a&table=techreport]. (Accessed on: 21.11.2012).

14. Upadhyaya BR, Eryurek E., **Application of neural network for sensory validation and plant monitoring**. *Neural Technology* 97: 170–176.