1 APPLICATION OF SOFT COMPUTING TECHNIQUES TO CLASSIFICATION OF LICENSED SUBJECTS

Jiří Kubalík[1], Marcel Jiřina[1], Oldřich Starý[2], Lenka Lhotská[1], Jan Suchý[1]
[1] *Department of Cybernetics, CTU Prague*
Technická 2, 166 27 Prague 6, Czech Republic
e-mail: kubalik@labe.felk.cvut.cz
[2] *Faculty of Electrical Engineering, CTU Prague*
Technická 2, 166 27 Prague 6, Czech Republic
e-mail: staryo@fel.cvut.cz

This paper presents an application of soft computing techniques to the construction of decision support tool used for identifying the economically unstable licensed subjects. The work has been initiated by the Czech Energy Regulatory Office whose main mission is to guard the regular heat supply without significant disturbances. Thus the main goal is to develop a tool for automatic identification of the companies that could cancel the supply due to economic problems without detailed examination of each company. In order to achieve the goal two approaches have been chosen. The first one is based on development of an aggregate evaluation criterion for assessing the firms. The other one uses artificial neural networks and multivariate decision trees induced with genetic programming for classification of the firms.

1. INTRODUCTION

The presented work has been initiated by the Czech Energy Regulatory Office (ERO) whose main mission is to guard the regular heat supply without significant disturbances. Authors were involved in developing the methodology for marking the possibly problematic licensed heat and co-generation facilities that can have some problems with the financial and economic stability and therefore the energy supply could be threatened in the near future. The ERO gathers the big amount of both technical and economical data but it is difficult if not impossible to process all the information for thousands of licensed subjects. Thus the main goal is to develop a tool for automatic identification of the companies that could cancel the supply due to economic problems without detailed examination of each company.

In order to achieve the goal two approaches have been chosen. The first one is based on development of an aggregate evaluation criterion for assessing the firms.

The solved problem can be restated as a knowledge mining task, where given the existing database of firms' records one wants to extract the knowledge of what is a good and what is a bad firm (measured in terms of economic stability). If each record is assigned an indicator that expresses its stability then the task belongs to the

class of supervised learning. Once a model acquiring the knowledge contained in the presented training database is built it can be used for classification of new records with unknown economic stability. In this work we use *artificial neural networks* and *multivariate decision trees* for modeling the knowledge. The multivariate decision trees are generated by *genetic programming*.

The rest of this paper is organised as follows. The next section introduces the Aggregate Evaluation Criterion, followed by sections describing the multivariate decision trees induced with genetic programming and the implementation of an artificial neural networks. We then outline the dataset and the utilized experimental methodology. The following section provides the results achieved with the decision trees and neural networks and the paper closes with conclusions.

2. AGGREGATE EVALUATION CRITERION

The original data set provided by ERO consists of raw descriptions of firms without indication of their economic stability. In order the data could be used for learning the decision tree and neural network model an economic stability value of each firm of the given training data have to be determined.

A set of five relevant risk factors and stability criteria has been selected at first. It consists of measure of long-term indebtedness, short-term financial position, operational return on assets, average equipment amortization and sales stability. Then the function called *aggregate evaluation criterion* (AEC) of financial stability that transforms the five criteria into just one has been developed. This function was used to assess the stability of the given firms. The evaluated firms can be sorted using this function, the firms with the maximum value signals to ERO to focus to them. Several firms have been examined in detail in order to approve a correctness of the AEC. It turned out that AEC gives a reasonably correct ranking of firms.

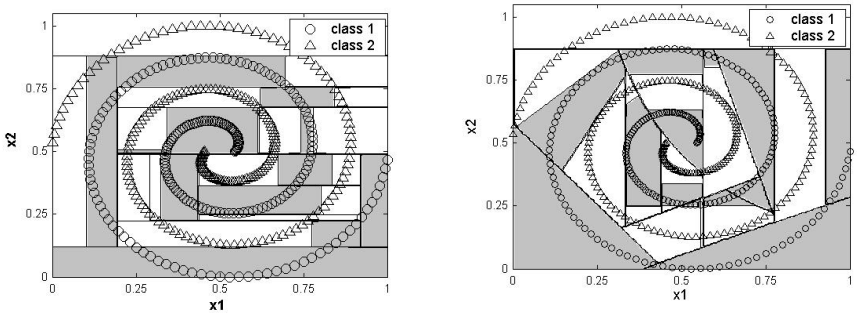
The next step in the development of the decision support system was to transform the knowledge contained in the database of labeled records into the form of (1) multivariate decision trees and () artificial neural network, which will be used for classification of the firms in the future.

3. MULTIVARIATE DECISION TREES

Decision tree (Quinlan, 1986) is a tree whose internal nodes are tests (on input attributes) and whose leaf nodes are categories. Each branch (path to another node) represents a value that the attribute might take. To classify a case, the root node is tested as a true-or-false decision point. Depending on the result of the test associated with the node, the case is passed down the appropriate branch, and the process continues. When a terminal node is reached, its stored value is the answer. A decision tree is constructed top-down. In each step a test for the actual node is chosen - starting with the root node - which best separates the given examples by classes. Usually, the quality of the test is measured by the information gain. The test applied to the dataset in the given node splits the data into several subsets, each of them representing the data of the corresponding child node. Every child node is further expanded – the best split function is found, generating its descendants – until the stopping condition is fulfilled in the newly generated node. The stopping criterion is usually defined as the maximum acceptable amount of information

contained in the node's data. So the process ends in a given node when the data contain samples belonging to only one class or some class significantly dominates in the node's data. The node is then assigned the identifier of that class.

In standard decision trees the test in the inner node is a test on the value of certain attribute "$attr_i < v_{ij}$", where v_{ij} is the value chosen from the domain of the i-th attribute. In this work we use *multivariate decision trees* (MDTs) with the tests of the form "$f(a_1,\ldots,a_n) < 0$", where f is an arbitrary function of the input attributes a_i using specified operations and operators (Brodley & Utgoff, 1995). Obviously such decision trees are more general than the standard decision trees, which allows to better model the given training data, see Figure 1.



a) b)

Figure 1 – Illustration of partitioning of the pattern space of synthetic data of two spirals a) by the standard decision tree and b) by the multivariate decision tree. The multivariate decision tree uses test functions composed of operators +, -, *, and /.

Utilization of genetic programming

When constructing the MDT the crucial point is to find the optimal function when new inner node is to be added into the tree. For this purpose genetic programming (GP) has been used in this work. GP is a powerful technique for automatically generating computer programs (Koza, 1992), (Bot & Langdon, 2000). GP operates on population of candidate solutions, each represented as a hierarchical parser tree – expressions representing test functions are evolved here. The complexity of evolved trees is determined by the set of operators and elementary functions. All individuals are evaluated i.e. assigned a fitness value expressing a performance measure of the represented solution. In our application the fitness reflects the quality of the split generated by given test function. A population of diverse individuals is then evolved generation by generation by means of reproduction, crossover and mutation operators. The process of evolving the population runs until some stopping criterion is fulfilled; usually it runs for some pre-specified number of generations. The best solution encountered during the whole run is then returned as the final solution.

4. ARTIFICIAL NEURAL NETWORKS

There are many paradigms of artificial neural networks. The most known and widely used is the multilayer perceptron networks (MLP), see e.g. (Bishop, 1995), (Rojas,

1996). The MLP consists of several layers of neurons called perceptron. Each perceptron calculates a post-synaptic activity (potential) as a weighted sum of its inputs and generates an output by means of an activation function. The activation function is often a logistic sigmoid or hyperbolic tangent. Frequently, the activation function of the output layer is linear, mostly the simple identity. A network of interconnected perceptrons represents powerful computational system capable of solving complex nonlinear tasks.

Thresholding

Thresholding is a process of assigning a class identifier to input pattern. Each of the five output neurons corresponding to one class can take a real value from the interval ⟨0.0, 1.0⟩. Generally, the resultant class is just the one that corresponds to the output neuron with highest value of its output.

In this work the situation is a bit different. As the boundary between adjacent classes are not sharp it may happen that for some input patterns the response of the network would be misleading in a sense that the correct class membership differs from the one that corresponds to the output neuron with the highest value. In such situations the above mentioned simple *winner-takes-it-all* strategy fails and thus is inappropriate for our purposes.

To resolve this problem we used a thresholding method that works as follows. First, two output neurons with the maximal value – the two most probable classes – are found. Then a relative difference between the two outputs is calculated. If the relative difference is less than a given threshold – indicating that the difference between the two most probable verdicts is not significant enough – then the pattern is assigned the higher (worse) class of the two ones. Otherwise, the pattern is assigned the class corresponding to the output neuron with the highest value. This thresholding strategy can be interpreted so that if any doubts of which class should be assigned to the pattern then the more pessimistic one is chosen – the worse rating is assigned to the firm. The threshold value 0.2 was used in this work.

5. REAL DATA AND EXPERIMENTAL METHODOLOGY

Classification of licensed subjects (firms) is based on five input parameters that describe different features of individual licensed subjects. To each licensed subject a category is assigned. The firms have been split into five classes according to their AEC value. The best firms with the lowest AEC value belong to class one, the worst ones belong to class five. Such a categorization is required by the ERO. The primary goal of the classification task is to identify the most unstable firms as reliably as possible in order not to miss any incompetent licensed subject.

First, the raw data were preprocessed so that each input parameter was saturated to minimal and maximal values and then the range of each input parameter was linearly scaled to <0.0, 1.0>. This adjustment is generally appropriate to eliminate large differences in the parameters.

Table 1 - Numbers of patterns for individual classes

Class	1	2	3	4	5
#patterns	16	286	332	54	16
% of patterns	2.3	40.5	47.2	7.7	2.3

The database provided by ERO consists of 704 records with highly unbalanced distribution of the classes, see Table 1. Classes 2 and 3 strongly dominate in the data whilst class 5 has only 16 records, i.e. this class represents only 2.3 per cent of the database. Obviously such a distribution of the classes is very bad.

Multivariate Decision Trees

When learning a model describing the data the whole dataset is split into training and test data so that the training data are used for training purposes and the test data are used for evaluating the final model. It is evident that applying such a concept on our data would hardly yield some general description of the training set that would correctly classify records from the test set as well. Due to this fact we decided to decompose the classification task into two parts. In the first step a *classifier I* that classifies to the following four classes is used:

- class A (corresponds to the original class 1),
- class B (corresponds to the original class 2),
- class C (corresponds to the original class 3),
- class D (union of original classes 4 and 5).

In the second step the records labeled as class D will be processed by *classifier II* which will separate class 4 records from class 5 ones. In both steps the distribution of classes in processed data is better than in the original dataset. In case of the classifier I the most important class D receives 10% of records in the data and in case of the classifier II the most important class 5 has 23% of records in the data.

In addition each of the classifiers consists of several MDTs in order to increase the robustness of the classifier. In order to determine the final verdict of such an *ensemble classifier* a simple majority rule is used – the most frequent class identifier out of the four returned answers is considered the final output of the classifier. In case of a draw the higher class identifier is taken as the final output of the classifier. This set up causes the classification to work in favor of the higher classes so a firm is always assigned the worse rating when it is on the edge.

The concept of the ensemble classifiers requires that the individual MDTs are as distinct as possible. Otherwise there would be no profit from combining multiple trees. The simplest way to obtain a set of unique trees is to use different training data for induction of each tree. The whole data set has been split into four equally sized disjunctive parts, each of them with the same proportion of classes as in the original dataset. For the learning purposes three data partitions were used the last one was used for testing the generated tree. This leads to *four different learning scenarios* that would hopefully generate four different MDTs. The division of data into four parts has been chosen with respect to the number of records of class 5, which is 16. Thus each of the learning scenarios has 12 records of class 5 in the training set and 4 records of class 5 in test set. A number of MDTs were generated for each *learning scenarios* and best representatives of each scenario were used in the final ensemble classifier. This should ensure that each MDT is to some extent unique so the final classifier should generalized well on new unseen data.

Artificial Neural Networks

When used the neural network approach the original data with 704 records were split by random into three disjunctive sets: training, validation and test sets. The database was divided into these classes in the proportion 2:1:1.

We tested three and four layered MLPs. Each MLP has five inputs and five outputs. The classification to individual classes is thus performed in code one-from-*N*. The number of hidden neurons varied from 15 to 30. The best results were obtained by the three layered MLP with 30 hidden neurons.

The post-synaptic activity is calculated by means of weighted sum of inputs in both hidden and output neurons. The activation function for the hidden layer is hyperbolic tangent and for the output layer is calculated by means of the *Softmax* rule, i.e. normalized exponential function along all outputs. This ensures that the individual outputs are in the range 0-1. Therefore, values of the outputs can be interpreted as probabilities of membership to the individual classes.

The combination of methods of the *backpropagation* (100 epochs) and conjugate gradient descent (20 epochs) was used for training of the MLP. The learning rate was 0.01 and momentum term 0.3.

6. RESULTS

This section presents results achieved with the classifier based on MDTs and provides a comparison with results of the ANN classifier. The experiments were performed using 4-fold cross-validation in order to get four different learning scenarios as described above. For each of the four training-test data configurations ten experiments were carried out and the success rates averaged over the 40 experiments are shown in tables.

Table 2 provides results achieved with the trees generated for classifier I. The average number of inner nodes of the generated trees was 12. Average accuracies on both training and test datasets show that the induced trees classify classes B (2) and C (3) with much better accuracy than the classes A (1) and D (4&5), which is given by the distribution of the classes in the data. Column "D/5" says that on average 83% and 78% of data belonging to class 5 were correctly classified as class D in training and test datasets, respectively. In other words, 83% (78%) of data belonging to class 5 proceeded to the next step where the classifier II is involved.

Table 2 – Success rates of MDTs generated for classifier I

Class	A	B	C	D / 5	Total
Training data [%]	22	89	88	**65 / 83**	85
Test data [%]	9	80	82	**56 / 78**	77

Table 3 – Success rates of classifier II

Class	4	5	Total
Training data [%]	100	**100**	100
Test data [%]	85	**38**	74

Table 3 shows average success rates of trees generated for classifier II. The average number of inner nodes (test functions) of the generated trees was 7. The trees were trained only on data labeled by classifier I as D. It shows that the trees were perfectly trained to classify the training data. In contrast the performance on the test data drops to 38%. Rather poor generalization ability results from an insufficient number of training data of class 5 – learning from 12 positive samples in 5-dimensional space can hardly be successful. However, the average number of

correctly classified samples of class 5 is $1.00 \times 12 + 0.38 \times 4 = 13.5$. This is sufficient accuracy with respect to the fact that the final classifier II as well as the classifier I will be assembled from four trees, each trained on partially different data (different learning scenario).

Ensemble classifiers I and II were assembled from trees chosen according to the following criteria:

° Simple trees measured by the number of inner nodes were preferred. This criterion ensures the possibility the tree is over-learned to the training data is reduced.

° Trees best classifying data of class 5 were preferred.

Performance of final ensemble classifiers I and II as well as the classification accuracy of the compound classifier are presented in Table 4. We observe a considerable improvement in the accuracy of classification of class 3, 4 and 5 when compared to the accuracy achieved with single trees. The most important observation is that classifier I correctly classifies 80% of data of class D (classes 4&5). Among the data labeled as D all records of class 5 are present, which are perfectly identified by classifier II. On the other hand, data belonging to class 1 are not classified at all. This is because we did not make any special arrangement with the aim to correctly learn class 1 as we did for class 5.

Table 4 – Success rates of the ensemble classifiers and the compound classifier

Class	1	2	3	4	5	Total
Ensemble classifier I [%]	0	81	95	80 / **100**		**85**
Ensemble classifier II [%]	-	-	-	97	**100**	**≈100**
Compound classifier [%]	0	81	95	70	**100**	**85**

Table 5 - Statistics of classification on training, validation, and test set

Class	1	2	3	4	5
Training set [%]	89	96	92	85	**78**
Validation set [%]	100	95	91	29	**75**
Test set [%]	75	94	85	69	**67**

Table 6 - Overall statistics for individual classes

Class	1	2	3	4	5	Total
Success rate [%]	88	95	90	74	**75**	90

Results achieved with ANN are summarized in Tables 5-6. Table 5 shows results on training, validation and testing sets. Table 6 shows overall results over all sets. The use of a simple neural network like MLP with 30 hidden neurons seems to be sufficient. The pattern space can be separated properly by hyper-planes and their combinations. The achieved overall quality of classification (app. 90 %.) is very good with respect to the given real problem and the available data. A drawback of the task is that the last fifth class contains only 16 patterns so it is difficult do make serious conclusion about classification to this class. Fortunately, utilizing the suggested thresholding can improve classification to this class. In practice, only some licensed subjects from the class 4 are classified to the class 5 but no licensed subjects from classes lower than 4 are wrongly classified to the class 5.

7. CONCLUSIONS

This paper presents an application of soft computing techniques to the construction of decision support tool used for identifying the economically unstable licensed subjects. The original data consists of raw descriptions of subjects without indication of their economic stability. First, the *aggregate evaluation criterion* has been developed for assessing the firms. Then the data labeled by AEC were used learning the classifiers based (1) on the *multivariate decision trees* and (2) on the *artificial neural network*. Achieved results show the classifiers work well on the given data. Moreover the proposed compound classifier based on multivariate decision trees is very robust so it is expected to work well on new previously unseen data as well.

It is evident that both the multivariate decision tree model and the neural network model generated using the data labeled by the AEC can only approximate the AEC. From this point of view the utilization of the models might seem useless. On the other hand, the AEC is a static model with its own parameters that are hard to tune for particular data. In particular the AEC is tailored to the currently available data and as such it might become irrelevant for the task when the situation for which it was developed changes - new data are provided or new evidence about the current firms is revealed. In such case the multivariate decision trees and neural networks would be preferred for the following reasons:

- Both types of models can be easily regenerated in order to fit well the training data when they change. As this is a long-term project new and updated data will be provided each year so the ability to adapt the model is very important.
- New factors characterizing the performance of firms can be used and easily incorporated into the models. Example of which might be the trends in attributes as the history of existing firms will be available after few years.
- Ensemble classifiers can be constructed. As the results achieved with ensemble decision tree based classifiers show a proper combination of a number of individual classifiers leads to robust classifier.

Moreover, the MDT and ANN models can be used for validating the AEC on the current data so that if both models return a different economic stability value than the AEC does for given firm it might signal the AEC is not well formed for the data.

8. ACKNOWLEDGMENTS

This research work was supported by the Grant Agency of the Czech Republic within the project No. 102/02/0132.

9. REFERENCES

1. Quinlan, J.R. Induction of decision trees. *Machine Learning*, 1, pp. 81-106, 1986.
2. Brodley, C.E., Utgoff, P.E. Multivariate decision trees. *Machine Learning*, 19, pp. 45-77, 1995.
3. Koza, J. Genetic Programming: on the programming of computers by means of natural selection. Cambridge, MA: The MIT Press, 1992.
4. Bot, M.C.J., Langdon, W.B. Application of genetic programming to induction of linear classification trees. *Proceedings of EuroGP 2000*, LNCS 1802, pp. 247-258. Springer, 2000.
5. Bishop, C. M. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.
6. Rojas, R. *Neural Networks: A Systematic Introduction*. Springer-Verlag, Berlin, Heidelberg, New York, 1996.