Comparison of different non-statistical classification methods

Ondřej Popelka[1], Jiří Hřebíček[2], Michael Štencl, Michal Hodinka, Oldřich Trenz

Abstract. In this article, we aim to compare different methods usable for solving classification problems. A substantial number of methods that are not based on mathematical statistics may be used. Exploring these methods is interesting, because they are often capable of solving problems, which are not easily solvable using classificators based purely on mathematical statistics.
There are many approaches available such as support vector machines, neural networks, evolutionary algorithms, parallel coordinates, etc. In this article, we concentrate on describing different neural network approaches, parallel coordinates and genetic algorithms. Neural networks come in many flavors (e.g. multi-layer perceptron, non-linear autoregressive networks) and they have achieved some recognition. Genetic algorithms also have been used for classification many times before, but with mixed results. In this article, we describe and evaluate different capabilities of these methods when used for economic data. This for example includes identification of hidden data structures, dealing with outliers and noise.

Keywords: classification, decision trees, neural networks, parallel coordinates, corporate performance, sustainability reporting.

JEL Classification: C44
AMS Classification: 90C15

1 Introduction

One of our current research projects is project No P403/11/1103 "Construction of Methods for Multi-factorial Assessment of Company Complex Performance in Selected Sectors" solved by Faculty of Business and Economics (FBE) at Mendel University in Brno in cooperation with Faculty of Business and Management (FBM) of Brno University of Technology (BUT). The project is funded by Czech Science Foundation and solved during the years 2011–2014. There are six main research targets defined in [14], one of which is the construction of quantitative and qualitative methods of the multifactor measurement of corporate performance.

To achieve the stated research goal we have analyzed corporate performance factors in chosen companies in the real estate and construction sectors. Only companies that have successfully implemented international management standards [13] such as ISO 9000 (quality), ISO 14000 and EMAS (environmental), ISO 18000 (health and safety) were evaluated. The performance factors including Environmental, Social and Governance (ESG) factors [18], [5] are being transformed to *Key Performance Indicators* (KPIs) [11], [15], which are organized into different standardized categories (economic, environmental, social, etc.).

One of the tasks, which are part of the development of multifactor measurements of company performance is solving of a classification problem on a multivariate dataset. In this article, we describe several methods usable for classification and our experiments with these methods. We focus on data-mining methods, because our primary sources of data are company questionnaires and reports. Therefore, our data may contain errors, missing values and other features, which may present [24] a difficulty for processing by pure statistical methods such as k-means.

2 Experiments

In our experiments, we decided to include these methods: neural networks, decision trees, support vector machines and genetic algorithms. This choice was motivated by some preliminary research [24] and by the methods used by the authors of the test dataset [19]. We have used a testing dataset for two reasons – firstly because we aim to conduct an independent comparison and secondly because our data collection is still in progress. In [19] the authors used the Support vector machine (SVM) method, Naive Bayes and a Decision Tree

[1] Department of informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 613 00 Brno, e-mail: ondrej.popelka@mendelu.cz, jiri.hrebicek@mendelu.cz, michael.stencl@mendelu.cz, michal.hodinka@mendelu.cz, oldrich.trenz@mendelu.cz.

to analyze the presented data. Software used was *rminer* library for R tool [7]. For our experiment we have used standard implementations in Weka data mining software suite for all algorithms apart from Grammatical Evolution (chapter 2.5), because that has no standardized implementation. The SVM algorithm is implemented using the LibSVM library.

2.1 Source data

As mentioned above, the dataset used for experiments in this paper, was collected by the authors of [19]. It contains results of direct bank marketing campaigns. Specifically it includes 17 campaigns of a Portuguese bank conducted between May 2008 and November 2010. These were always conducted over the phone and the customer was offered a long-term deposit application. Originally the dataset contained 79 354 contacts, out of which contacts with missing data or inconclusive results were discarded leading to a dataset with 45 211 instances with two possible outcomes – either the client signed for the long-term deposit or not.

The dataset contains 16 input variables. There are eight variables related to the client:

- age (numeric),
- job type (categorical),
- marital status (categorical),
- education (categorical),
- whether the client has credit in default (binary),
- average yearly balance in Euros (numeric),
- whether the client has a loan (binary),
- whether the client has a personal loan (binary).

Four variables relate to the last contact of the current campaign:

- contact communication type (categorical),
- last contact day of the month (numeric),
- last contact month of year (categorical),
- last contact duration in seconds (numeric).

There are four more variables regarding the campaign:

- number of contacts performed during this campaign and for this client (numeric),
- number of days that passed by after the client was last contacted from a previous campaign (numeric),
- number of contacts performed before this campaign and for this client (numeric),
- outcome of the previous marketing campaign (categorical) [19].

As mentioned above, the output variable corresponds to campaign output, which has been reduced to a binary output. For testing of the classification algorithms, the dataset was split into 2/3 of training data and 1/3 of test data. This leaves us with 31 489 randomly chosen instances for training and 13 722 instances for testing. Baseline classifier (ZeroR classifier) which selects all instances into the largest class has an accuracy of 88.30%. This is caused by strong skew of the output variable. It corresponds to the fact than only 11.70% of bank clients agreed to sign for the long-term deposit (in the unfiltered data, the success rate is 8%) [19]. It is important to note that such strong skew in the dataset provides a considerable challenge for the classification algorithm.

2.2 Decision trees

As a base starting method, the decision trees were used, because this is a very widely used classification technique. A decision tree algorithm is a technique which recursively splits the dataset instances into classes until all the data are assigned to a class. Basic approaches are breadth-first or depth-first greedy searches. Several algorithms may be used to construct a decision tree. The most common method [25] is the C4.5 algorithm [22][25], which is an improved version of the older ID3 algorithm. Furthermore, there are a number of algorithms, which claim to have better accuracy, such as ADTree [10] or Random Forests [3]. An in-depth comparison of several decision tree algorithms can be found in [2] and [22]. For the sake of comparison, we have also tested a BFTree algorithm that implements best-first decision tree classifier.

An Alternating decision tree algorithm (ADTree) was designed especially so that the resulting tree is intelligible to humans [10]. It is a combination of a weak classifier based on decision trees and decision stumps with a boosting learning algorithm (ADTBoost). This is done by replacing an ordinary decision tree with an alternating tree that uses prediction sums sign instead of the class label itself as a leaf node. Therefore, each instance is mapped to a real value prediction, which is a sum of the predictions of the base rules in the instance set. The actual label is replaced by the sign of the sum of the predictions. This transformation has two main

effects. Without an extension, the ADTree supports only two-class problems (binary class). Each rule has assigned a confidence value – classification margin – that suggests the reliability of the rule.

2.3 Support Vector Machine

Support vector machine is a learning algorithm originally designed only for linear classification models. The algorithm tries to find objects (support vectors) which define hyper-planes separating a set of any two classes with maximum margin. Nonlinear classification problems may be solved by transforming the search space into a transformed feature space. This transformation is done using the kernel basis function, where several options are available [16]. Several options are available for the partitioning kernel function – linear function, polynomial function with different degrees, B-spline function and radial basis function (RBF) [16]. However, during our experiments none of the mentioned transformations led to a classification better than the one of the baseline classifier. This is further discussed in the Results section.

2.4 Neural networks

Neural networks are one of the best-known representatives of learning algorithms and as such, they may be divided into two basic classes – unsupervised and supervised networks. The former are commonly represented by Competitive layers or Self-organizing maps which automatically find relationships within the input vectors [12]. Given the fact that the processed data are labeled and we have a priori knowledge of the output classes, we have concentrated on supervised networks, which exhibit better performance in such cases [9]. There are many flavors of supervised networks, where by far the most common one is still Multilayer Perceptron (MLP) networks with back-propagation learning algorithm. Additionally in our previous research [23], we have been testing other approaches such as Radial Basis Function (RBF) and Nonlinear autoregressive neural network (NAR).

In [23][23], it has been demonstrated, that RBF neural networks exhibit similar performance to MLP networks while having a substantially better performance. RBF networks are feed-forward networks in which perceptrons are replaced by local units. These local units use radial functions which, given a central point, provide the same output for arguments with the same distance from the central point. RBF networks generally have only one hidden layer that makes the back-propagation learning algorithm much simpler and faster.

2.5 Genetic algorithms

Genetic algorithms represent a broad area of methods and algorithms. Genetic algorithms are mostly used as optimization algorithms. However, they may also be used for classification. There have been many experiments performed in this area such as [6]. However, genetic algorithms have suffered a noticeable decline in popularity. More precisely rather than being used as a sole classification algorithm, they are used as part of hybrid methods, for example in: [17], [4] and [1].

We have however decided to use a slightly more specific flavor of genetic algorithms – grammatical evolution. This algorithm uses a genetic algorithm to control a generative context-free grammar. That is, a context-free grammar is defined, which may be used to generate strings in an arbitrary format, for example “if (condition and condition) then class1 else class2”. This is defined using a set of terminals, non-terminals and production rules that define the language in which the strings may be generated. A genetic algorithm is then used to generate various strings of the language, which are evaluated and assigned a fitness value (in this case, classification accuracy was used).

The algorithm is extensively described in [20]. Further, it is extended to generate numeric constants by using two-level grammatical evolution as described in [21]. This is important when solving this classification problem, because the conditions are to be in format “if (age > 20) then will-sign else won’t sign”. That is, each condition expression consists of a variable (see chapter 2.1) and a constant value of that variable which defines class border. A limitation in the current implementation requires that the constant values are numeric, therefore for this experiment, the text attributes were converted to discrete numeric variables.

As stated above, the algorithm output is arbitrary, so it may be same as a decision tree model or it may be generated so that it is directly usable in a common programming language (C++, Java, PHP, etc.). On the other hand, this means that the task is much more difficult. A terminal criterion for the genetic algorithm was set to either classification accuracy of 89.2% or computation time of 8000 seconds. We have set the time criteria arbitrarily to an acceptable time of computation (which is still about 1000 times bigger than computation of ADTree). All of the ten runs performed on GE algorithm stopped on the time criterion and have not reached classification accuracy better than the baseline classifier. This is a bit of a disappointment, but not surprising, Because GE is a very generic string generation algorithm not optimized for classification problems.

3 Results

In our experiments, we have investigated several methods for automatically classifying a multivariate dataset. The results are summarized in Table 1 that shows different tested methods. For each method the *classification accuracy* (amount of correctly classified instances), *the time required* to finish the classification and *ROC area* (area under the ROC curve) is shown.

Algorithm	Classification accuracy	Time required [s]	ROC Area
ADTree	89.57%	6.6	0.889
RandomForest	89.57%	3.5	0.892
BFTree	89.61%	100.1	0.730
C4.5 (J48)	90.20%	2.7	0.839
RBF	89.27%	4.0	0.826
MLP	89.26%	706.8	0.877
SVM	88.23%	2618.6	0.500
GE	88.23%	> 8000	0.500

Table 1: Results of comparison, average over 10 runs for best performing configuration.

In our comparison, we have focused on methods that can provide the information about the exact classification rules. The method should provide at least the information about more significant and less significant for the classification. This kind of information is generally best obtained from decision trees that provide classification rules in a format easy to visualize and understand.

Experiments with decision trees can be considered highly successful. We have obtained slightly better results than the authors of the data in [19] had, who achieved ROC Area of 0.868. As mentioned above, we are also interested in providing the classification rules in readable format. For this, it is interesting to compare the sizes of decision trees, which are shown in Table 2. There is a strong disproportion of the size of the decision tree across multiple algorithms. The Random Forest algorithm is not shown in the results, because it consists of several trees. However, the total number of nodes was higher than 10 000. An easy to understand set of rules is obtained from the Alternating decision tree (ADTree). From this set, it is clear that important variables with high discriminative power are: *duration of the call*, *month of contact*, *previous outcome*, *whether the customer has a housing loan* and *contact type*. This is consistent with the original results.

Algorithm	Number of nodes	Number of leaves
ADTree	31	21
BFTree	279	140
C4.5	1168	1716

Table 2: The size of decision trees for different algorithms.

The SVM based classifier proved the least successful as it did classify the same as baseline classifier. This is a considerable surprise, since in the original paper [19] the SVM based classifier was the best performing (although on a smaller dataset). Overcoming the high memory consumption problem of the SVM classifier was enabled by using 64-bit architecture. A possible explanation of the poor classification performance is that in [19] there were 29 input variables used. In our experiments, we have used only 16 publicly available input variables. We will investigate this problem in our future work. Because the classification was not successful, we did not attempt to extract the decision rules, although this is possible using [8].

The results of neural networks are perfectly comparable with those of decision trees in terms of both precision and performance. However, the RBF network greatly outperforms the MLP type of network, which is consistent with the results of [23]. Unfortunately, a big trade of RBF networks is that it does not provide any information about the classification rules. The MLP network does not provide the rules either, but still some information may be extracted from the weights assigned to different variables. From this, we can infer, that *duration of the call*, *month of contact* and *previous outcome* or *housing loan* are variables consistently with above average weights. Again, this result is consistent with both decision trees and original results.

The importance or discriminative power of variables may also be inferred from preliminary data analysis. We used the principles of exploratory data analysis (EDA) and visualizations such as parallel coordinates and Polyviz visualizations. Although without additional software, the importance of the variables cannot be quantized. Still these tools provide an invaluable insight into the dataset, which is useful especially for visual data mining and further to verify the results of the decision tree algorithms.

The conclusion can be stated that the favorite method so far is Alternating decision tree because with computational time under 10 seconds (for 45 211 instances) it belongs to the group of algorithms with very good performance. In addition, the classification performance is the second best of the tested algorithms. Furthermore, the classification rules are provided in easy to understand format and their amount is completely acceptable to a human being. This is a very important property of the algorithm, because in our task of constructing multifactor measurements of company performance we need to ensure that such performance measurements are presentable to the company management. It may come as a surprise that the more sophisticated methods were outperformed by simpler methods; nevertheless, we still consider it an important result.

Acknowledgements

This paper is supported by the Czech Science Foundation. Name of the Project: Construction of Methods for Multifactor Assessment of Company Complex Performance in Selected Sectors. Registration No: P403/11/2085.

References

[1] Ahna, H., and Kim, K.: Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach, *Applied Soft Computing* Volume 9, Issue 2 (March 2009), 599–607.

[2] Anyanwu, M. N., and Shiva, S.: Comparative Analysis of Serial Decision Tree Classification Algorithms, *International Journal of Computer Science and Security (IJCSS)* Volume 3 Issue 3 (2009), ISSN 1985-1553, 230–240.

[3] Breiman, L.: Random Forests, *Machine Learning* Volume 45 Issue 1 (2001), Springer, 5–32.

[4] Chen, M. Classification Techniques of Neural Networks Using Improved Genetic Algorithms. In Proceedings of WGEC '08. Second International Conference on Genetic and Evolutionary Computing, 2008, ISBN 978-0-7695-3334-6, 115–119.

[5] Chvátalová, Z., Kocmanová, A., and Dočekalová, M.: Corporate Sustainability Reporting and Measuring Corporate Performance. In: *Proceedings of Environmental Software Systems. Frameworks of eEnvironment. 9th IFIP WG 5.11 International Symposium. ISESS 2011*, Springer, Heidelberg, 2011, 398–406.

[6] Corcoran, A.L.: Using real-valued genetic algorithms to evolve rule sets for classification. In: *Proceedings of Evolutionary Computation - IEEE World Congress on Computational Intelligence*, Orlando, USA. 1994, ISBN 0-7803-1899-4, 120–124.

[7] Cortez, P.: Data Mining with Neural Networks and Support Vector Machines using the R/rminer Tool. In: *Proceedings of the 10th Industrial Conference on Data Mining*, Berlin, Germany, July 2010, Springer, LNAI 6171, 572–583.

[8] Cortez, P., and Embrechts, M.: Opening Black Box Data Mining Models Using Sensitivity Analysis. In: *Proceedings of IEEE Symposium on Computational Intelligence and Data Mining*, pp. 341–348. Paris, France, 2011.

[9] Fejfar, J., Šťastný, J. Cepl, M. Time series classification using k-Nearest Neighbours, Multilayer Perceptron and Learning Vector Quantization algorithms. In: *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis* Volume 60 Issue 2 (2012), ISSN 1211-8516, 69–72.

[10] Freund, Y., Mason, L.: The alternating decision tree learning algorithm. In: *Proceeding of the Sixteenth International Conference on Machine Learning*, Bled, Slovenia, 1999, 124–133.

[11] Garz, H., Schnella, F., and Frank, R.: KPIs for ESG. *A Guideline for the Integration of ESG into Financial Analysis and Corporate Validation*. Version 3.0, Frankfurt, DVFA/EFFAS, [Online], Available: http://www.dvfa.de/files/die_dvfa/kommissionen/non_financials/application/pdf/KPIs_ESG_FINAL.pdf, 2010.

[12] Haykin, S. O.: *Neural Networks and Learning Machines*, 3rd edition, Prentice Hall, ISBN 9780131471399, 2009.

[13] Hřebíček, J., Soukopová, J., and Kutová, E.: Standardization of Key Performance Indicators for Environmental Management and Reporting in the Czech Republic. *International Journal of Energy and Environment* Volume 4 Issue 4 (2010), 169–176.

[14] Hřebíček, J., Soukopová, J., Štencl, M., and Trenz, O.: Corporate Performance Evaluation and Reporting. In: *Proceedings International Conference on Environment, Economics, Energy, Devices, Systems, Communications, Computers, Pure and Applied Mathematics* (*WSEAS)*, Wisconsin, USA, 2011, 338–343.

[15] Hřebíček, J., Soukopová, J., Štencl, M., and Trenz, O.: "Corporate Key Performance Indicators for Environmental Management and Reporting," *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis* Volume 59 (February 2011), 99–108.
[16] Ivanciuc, O.: Applications of Support Vector Machines in Chemistry. In: *Reviews in Computational Chemistry* Volume 23 (2007), Wiley-VCH, Weinheim. 291–400.
[17] Kalia, H., Dehuri, S., and Ghosh, A.: Multi-Objective Genetic Algorithms for Fuzzy Classification Rule Mining: A Survey. *The IUP Journal of Information Technology* Volume 7 Issue 1 (March 2011), 7–34.
[18] Kocmanová, A., Dočekalová, M., Němeček, P., and Šimberová, I.: Sustainability: Environmental, Social and Corporate Governance Performance in Czech SMEs. In: *Proceedings of The 15th World Multi-Conference on Systemics, Cybernetics and Informatics* (WMSCI 2011), Orlando, USA, 2011, 94–99.
[19] Moro, S., Laureano, R., and Cortez, P.: Using data mining for bank direct marketing : an application of the CRISP-DM methodology. In: *Proceedings of the European Simulation and Modelling Conference – ESM'2011 (*EUROSIS), Guimarães, Portugal, October, 2011, 117–121,
[20] Popelka, O., and Ošmera, P.: Parallel Grammatical Evolution for Circuit Optimization. *Lecture Notes in Computer Science* Volume06 Issue 5216 (2008), ISSN 0302-9743. 425–430.
[21] Popelka, O. Two-level optimization using parallel grammatical evolution and differential evolution. In: *Proceedings of Mendel '07, 13th International Conference on Soft Computing*, Brno, 2007, ISBN 978-80-214-3473-8, 88-92.
[22] Quinlan, R.: C4.5: *Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA. 1993.
[23] Štencl, M., Popelka, O., and Šťastný, J.: Comparison of time series forecasting with artificial neural network and statistical approach. In *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis* Volume 59 Issue 2 (2011), ISSN 1211-8516, 347-352.
[24] Štencl, M., and Šťastný, J.: Artificial Neural Networks Numerical Forecasting of Economic Time Series. *Artificial Neural Networks – Application. Artificial Neural Network*. Riejka, Croatia, InTech, 2011. ISBN 978-953-307-188-6. pp. 13–28.
[25] Venkatadri, M., and Lokanatha, C. R.: A Comparative Study On Decision Tree Classification Algorithms In Data Mining. *International Journal Of Computer Applications In Engineering, Technology And Sciences (IJ-CA-ETS)* Volume 2 Issue 2 (2010), ISSN 0974-3596. 24–29.