Neuro-fuzzy Data Mining for Target Group Selection in Retail Banking

Johannes Ruhland and Thomas Wittmann

Lehrstuhl für Wirtschaftsinformatik, Friedrich-Schiller-Universität Jena
Carl-Zeiß-Str. 3, 07743 Jena, Germany
Tel.: +49 - 3641 - 943313, Fax: +49 - 3641 - 943312
email: j.ruhland@wiwi.uni-jena.de, t.wittmann@wiwi.uni-jena.de

Abstract: Data Mining Algorithms are capable of ‘pressing the crude data coal into diamonds of knowledge’. Neuro-Fuzzy Systems (NFS) in particular promise to combine the benefits of both fuzzy systems and neural networks, and are thus able to learn IF-THEN-rules, which are easy to interpret, from data. Hence, they are a very promising Data Mining Approach. In this case study we describe how to support a bank’s new direct mailing campaign based on data about their customers and their reactions on a past campaign with a Neuro-Fuzzy System. We will describe how Neuro-Fuzzy Systems can be used as Data Mining tools to extract descriptions of interesting target groups for this bank. We will also show which preprocessing and postprocessing steps are indispensable to make this Neuro-Fuzzy Data Mining kernel work.

1 Introduction

Companies doing database marketing experience target group selection as a core problem. At the same time they are often confronted with a huge amount of data stored in their data banks. These could be a rich source of knowldege, if only properly used. The new field of research, called Knowledge Discovery in Databases (KDD) aims at closing this gap by developing and integrating Data Mining Algorithms, which are capable of ‘pressing the crude data coal into diamonds of knowledge’. In this case study we describe how to support a bank’s new direct mailing campaign based on data about their customers and their reactions on a past campaign. To promote a new product, one of Germany´s leading retail banks had conducted a locally confined but otherwise large mailing campaign. To efficiently extend this action to the whole of the country, a forecast of reaction probablilty based on demographic and customer history data is required. The database consists of 186.162 cases (656 of them being respondents and the rest non-respondents) and 28 selected attributes, e.g. date of birth, sum of transactions etc., as well as the responding behaviour.

Neuro-Fuzzy Systems (NFS) promise to combine the benefits of both fuzzy systems and neural networks, because the hybrid NFS-architecture can be adapted and

J.M. Zytkow and J. Rauch (Eds.): PKDD’99, LNAI 1704, pp. 530-535, 1999.
© Springer-Verlag Berlin Heidelberg 1999

also interpreted during learning as well as afterwards [3]. Here we focus on a specific approach called NEFCLASS (NEuro Fuzzy CLASSification), which is available as freeware from the Institute of Information and Communication Systems, Faculty of Computer Science at the University of Magdeburg, Germany (http://fuzzy.cs.uni-magdeburg.de/welcome.html). NEFCLASS is an interactive tool for data analysis and determines the correct class or category of a given input pattern by a fuzzy system that is mapped on a feed-forward three-layered Multilayer Perceptron [14].

The effectiveness of NFS is empirically proven for small files like the iris data (see for example [2] or [4]). But when it comes to analyzing real-life problems, their main advantages, the ease of understandability and the capability to process automatically large data bases remains a much acclaimed desire rather than a proven fact, though. Therefore in a previous study [7] we have identified the benefits and shortcomings of NFS for Data Mining based on a real-life data file and criteria, which we have derived from practitioners requirements. Regarding classification quality, we have found Neuro-Fuzzy Systems like NEFCLASS as good as other algorithms tested. Their advantages are ease of interpretation (fuzzy IF–THEN rules) and their ability to easily integrate a priori knowledge to enhance performance and/or classification quality. However, two severe problems may jeopardize their success, especially for large databases, the inability to handle missing values and to identify the most relevant attributes for the rules which leads to a combinatorical explosion in both run time and in breadth and depth of the rule base. It has turned out that pre- and postprocessing efforts are indispensible for NFS (like most Data Mining algorithms) to be adequate for Data Mining. The Knowledge Discovery in Databases paradigm as described by [1] offers a conceptual framework for the design of a knowledge extracting system that consists of preprocessing steps, a Data Mining kernel and postprocessing measures.

2 Data Preprocessing

Missing values can cause severe problems for Data Mining. This holds true for Data Warehouses in particular, where data are collected from many different, often heterogeneous sources. Different solutions have been developed in the past to solve this problem. To us, the most promising approach is to explicitly impute the missing data. We have tried to use the infomation of a decision tree for imputation. This decision tree is developed by the C5.0 algorithm [5], which is by far more efficient than traditional clustering approaches. We interpret the leaves of the tree as homgeneous groups of cases. To put widespread experience in a nutshell, this method can efficiently and robustly model data dependencies. When building the tree the algorithm can also utilize cases with missing values by making assumptions about the distribution of the variable affected which leads to the fractional numbers in the leaves, as shown in figure 1 [5].

To use this tree for imputation we first have to compute a quality score for each eligible imputation value (or imputation value constellation in case more than one attribute is missing) which will be based on the similarity between the case with the missing data and each path down the tree to a leaf. In detail, we compute similarity as the weighted sum of the distance at every knot (top knots receive a higher weight) including a proxy-knot for the class variable to generate absolutely homogeneous end leaves. This score is multiplied by a „significance value", a heuristic quality index considering the length of the path and the total number of cases in the leaf.

profession group	age group	class:
missing	3	bad customer

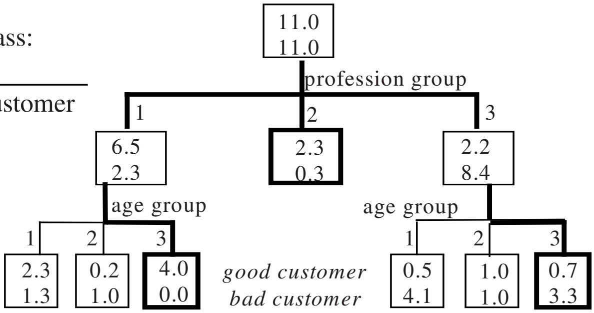


Fig. 1. C5.0-decision-tree based imputation.

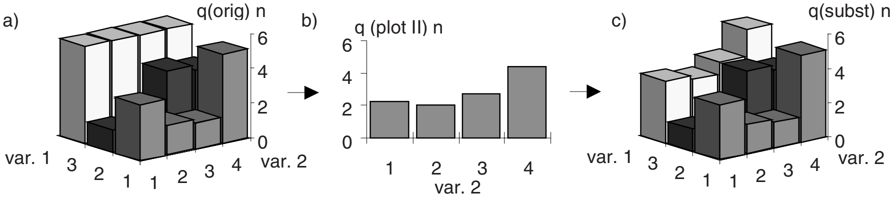


Fig. 2. Quality plot and defuzzification by Global Max Dice.

Secondly, we have to identify the best imputation value (constellation) by the quality scores of all possible imputation values. To visualize this problem one can draw a (n+1)-dimensional plot for a case with n missing values (see Figure 2 a displaying our score as a function of value constellations). To find a suitable imputation value set different methods are possible, which we have borrowed from the defuzzification strategies in fuzzy set theory [8]. The most promising approach is to use the constellation with the maximum quality score for imputation. This will not always yield unique results, because the decision tree does not have to be developed completely. Hence, some variables might not be defined. In figure 2 a) for example, attribute 2 will not enter the tree if attribute 1 equals 3. We can use the best constellation that is completely defined (*Single Max*) ((1,4) in figure 2 a) or do an unrestricted search for the maximum and in case this constellation is not unique, use a surrogate procedure on the not-yet-defined attribute(s). The surrogate value can be the attribute's global mean (*Global Max Mean*) (leading to (3,2) for the example in figure 2 a) or the value with the highest average quality, the average being computed over the defined variables (*Global Max Dice*) ((3,4) in Figure 2 a). Applied to the case of figure 2 a) this means averaging over variable 1 to get the (marginal) quality distribution displayed in figure 2 b).

$$\mathrm{Q}_{\mathrm{n}}^{\mathrm{subst}} = \mathrm{Q}_{\nu}^{\mathrm{orig}} * \left\{ 1 - \tau \left[\frac{Q_{\mathrm{max}}^{\mathrm{plot\ II}} - Q_{\mathrm{n}}^{\mathrm{p}lot\ \mathrm{II}}}{Q_{\mathrm{max}}^{\mathrm{plot\ II}} - Q_{\mathrm{min}}^{\mathrm{plot\ II}}} \right] \right\} . \qquad \mathbf{(1)}$$

$\mathrm{Q}_{\mathrm{n}}^{\mathrm{subst}} =$ surrogate quality score of imputation value (constellation) n,
$\mathrm{Q}_{\mathrm{v}}^{\mathrm{orig}} =$ original quality score of v; this not completely defined constellation covers constellation n,
$\mathrm{Q}_{\mathrm{n}}^{\mathrm{plot\ II}} =$ quality score of imputation value (constellation) n in plot II,
$\mathrm{Q}_{\mathrm{max}}^{\mathrm{plot\ II}} =$ maximum quality score in plot II,
$\mathrm{Q}_{\mathrm{min}}^{\mathrm{plot\ II}} =$ minimum quality score in plot II,
$\tau =$ factor determining how much the optimal imputation value (constellation) found by Global Max dominantes other imputation values (constellations).

In essence, this amounts to using the marginal score distribution of variable 2 to modulate branches where the decision tree is not fully developed. This definition of a surrogate value is not so essential, because the fact, that the decision tree algorithm did not take into consideration that attribute shows its small discriminating power. But the maximum approach has a distinctive benefit, because it allows some kind of multiple imputation [6] without really imputing different values for one missing data by a simple extension of the method. For each missing value constellation, the n best value tuples are stored together with their quality scores (n can be chosen freely, n=1 corresponds to single imputation) and choose one of them randomly according to its quality score. In doing so we avoid the often criticized reduction of variance inherent to ordinary, deterministic imputation methods [6]. In SPSS for example, this problem is solved by artificially adding variance enlarging noise, whereas the method explained above seems much more elegant and appropriate.

In our case we had 4 attributes with missing data in 0.3%, 27.5% , 50.7% and 86.4% of all cases. Therefore, we would have lost 95% of all cases if using the listwise deletion technique. This would lead to a very high loss of information, especially because all but nine (=1.4%) of all respondents would have to be dropped. The traditional imputation algorithms build their models only on the complete cases. But only 5% of all cases are complete; thus we can hardly trust the regression parameters or clusters. The C5.0-algorithm also uses cases with missing data, although this is done in a very brute force way. This is why we have chosen this method. We have used *Global Max Dice* for defuzzification with n=3, i.e. multiple imputation.

Due to the reasons mention above, we had to focus on relevant attributes and cases. Different approaches to attribute selection have been proposed. To put our experience in a nutshell, we can state an efficiency-effectiveness-dilemma, proving none of the single solutions to be optimal. To overcome this dilemma, we have developed a model combining different approaches in a stepwise procedure [9]. In each step we eliminate cases with methods that are more sophisticated, but also less efficient than the method in the predecessor step. The choice of methods in each step depends to a large extend on the data situation. In our application example we have used measures of variable entropy, a χ^2-test for independence of the output variable from each potential influence (considered in isolation), clustering of variables, identification of the most informative attributes by a C5.0 decision tree and a backward elimination with wrapper evaluation. After having selected the relevant attributes, we have drawn a training sample by pure random sampling consisting of about 300 respondents and 300 non-respondents.

3 Data Mining with NEFCLASS and Rule Postprocessing

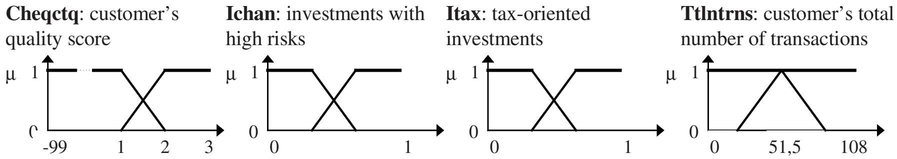
NEFCLASS learns the number of rule units and their connections, i.e. the rules. If the user restricts the maximum number of rules, the best rules are selected according to an optimisation criterion [3]. This maximum rule number and the number of fuzzy sets per input variable are the most important user defined parameters. The optimal number of rules heavily depends on the attribute subset used. Therefore, we have optimized this parameter after every fifth iteration of the variable backward elimination process. The number of fuzzy sets per input variable was generally set to two. For the binary variables this is an obvious choice. For continuous attributes we also identified two fuzzy sets as being sufficient. Using more fuzzy sets will enhance classification quality just marginally, but exponentially increases the number of rules. After having identified the optimal parameters and the optimal attribute subset, we have finally trained NEFCLASS coming up with the rule base shown in figure 3 b).

a)

Cheqctq	Ichan	Itax	Ttlntrns		
large	small	small	large	→	respondent
small	large	small	large	→	respondent
small	small	small	large	→	respondent
large	small	small	small	→	respondent
large	large	small	small	→	respondent
large	large	large	small	→	respondent
small	large	small	small	→	non-resp.
small	small	small	small	→	non-resp.
small	small	large	small	→	non-resp.
large	small	large	small	→	non-resp.

b)

Ttlntrns	Cheqctq	Ichan	Itax		
large				→	respondent
small	large	large		→	respondent
small	large	small	small	→	respondent
small	large	small	large	→	non-resp.
small	small			→	non-resp.


Cheqctq: customer's quality score

Ichan: investments with high risks

Itax: tax-oriented investments

Ttlntrns: customer's total number of transactions

Fig. 3. Original and aggregated rule base including membership functions.

The classification quality was quite good with 72.7% of correctly classified cases in the validation set (respondents: 68.7%; non-respondents: 77,7%). The rules (for the sake of simplification in a matrix notation) and the membership functions are shown in figure 5a). The rules shown in figure 5 reflect some interesting knowledge about the bank's target groups. In figure 5b) we further aggregated the rule base manually. A tool is being developed, that will not only aggregate rules automatically, but also represent them to the user in an adequate and interactive way.

4 Conclusion

The results of this study affirmed the previous experience that Neuro-Fuzzy Systems are not able to outperform alternative approaches, e.g. neural nets and discriminant analysis, with respect to classification quality. But they provide a rule base that is very compact and well understandable. Extensive preprocessing activities have been necessary, especially concerning the imputation of missing values and selection of relevant attributes and cases. Besides, we believe that these tools are also of general relevance. Intelligent postprocessing can further enhance the resulting rule base's power of expression. In this paper, we have shown some promising approaches for these steps as well as their effectiveness and efficiency. Of course, we are still far away from an integrated data flow. But our experience with the single modules described above is very promising. However, these first results have to be further validated for different Neuro-Fuzzy Systems and different data situations. Our final goal is to integrate these modular solutions into a comprehensive KDD tool box.

This report is based on a project promoted by funds of the Thuringian Ministry for Science, Research and Culture. The authors are responsible for the content of this publication.

References

1. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Knowledge Discovery and Data Mining: Towards a Unifying Framework, In: Simoudis, E., Han, J. (Hrsg.), Proceedings of Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, August 2-4, 1996, Menlo Park 1996, p. 82-88
2. Halgamuge, S. K., Mari, A., Glesner, M., Fast Perceptron learning by Fuzzy Controlled Dynamic Adaption of Network Parameters, In: Kruse, R., Gebhardt, J., Palm, R. (Hrsg.), Fuzzy Systems in Computer Science, Wiesbaden 1994, p. 129-139
3. Nauck, D., Klawonn, F., Kruse, R., Neuronale Netze und Fuzzy-Systeme. 2. Aufl., Braunschweig Wiesbaden 1996
4. Nauck, D., Nauck, U., Kruse, R., Generating Classification Rules with the Neuro-Fuzzy System NEFCLASS, In: Proceedings Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS'96), Berkley, June 19-22, 1996
5. Quinlan, J. R., C4.5. Programs for Machine Learning, San Mateo, 1993
6. Rubin, D. B., Multiple Imputation for Nonresponse in Surveys, New York 1987.
7. Ruhland, J., Wittmann, T., Neurofuzzy Systems In Large Databases - A comparison of Alternative Algorithms for a real-life Classification Problem, In: Proceedings EUFIT'97, Aachen, September 8-11 1997, p. 1517-1521
8. Wittmann, T., Knowledge Discovery in Databases mit Neuro-Fuzzy-Systemen. Entwurf für einen integrierten Ansatz zum Data Mining in betrieblichen Datenbanken. Friedrich-Schiller-Universität Jena, Wirtschaftswiss. Fakultät, Diskussionspapier Serie A, Nr 98/07, Mai 1998
9. Wittmann, T., Ruhland, J., Untersuchung der Zusammenhänge zwischen Fahrzeugmerkmalen und Störungsanfälligkeiten mittels Neuro Fuzzy Systemen, in: Kuhl, J., Nissen, V., Tietze, M., Soft Computing in Produktion und Materialwirtschaft, Göttingen 1998, p.71-85