RISK MANAGEMENT BASED ON EXPERT RULES AND DATA-MINING: A CASE STUDY IN INSURANCE

Hennie Daniels [1,2] and Han van Dissel[2]

[1]Tilburg University, CentER for Economic Research,Tilburg, PO Box 90153, 5000 LE The Netherlands, phone: +31 13 466 2026, e-mail: daniels@kub.nl
[2]Erasmus University Rotterdam, Rotterdam School of Management, The Netherlands.

ABSTRACT

Correctness, transparency and effectiveness are the principal attributes of knowledge derived from databases using data mining. In the current data mining research there is a focus on efficiency improvement of algorithms for knowledge discovery. However, improving the algorithms is often not sufficient. The limitations of data mining can only be dissolved by the integration of knowledge of experts in the field, encoded in some accessible way, with knowledge derived from patterns in the databases. In this paper we discuss an approach for combining expert knowledge and knowledge derived from transactional databases. The approach proposed is applicable to a wide variety of risk management problems. We illustrate the approach with a case study on fraud detection in an insurance company. The case clearly shows that the combination of expert knowledge with monotomic neural networks leads to significant performance improvements.

1. INTRODUCTION

The goal of a data mining system is to derive useful knowledge that is implicitly present in large company databases. In recent years there has been a lot of interest in theory, software and applications of data mining in virtually all business areas where large amounts of data are recorded (Han and Kamber, 2001; Fayyad et al., 1996). In this paper a data mining system is to be understood as the complete system: the database or data-warehouse, software for mining and analyses, the knowledge derived from it and the part of the system supporting final decision making in a business setting.

Apart from the well-known limitations concerned with data quality, one often encounters major problems with the application of a data mining system if the knowledge discovery process is conducted by a blind search based on a brute force approach. Frequently occurring causes are:

- Incompatibility of the model derived from transaction databases with knowledge embedded in corporate policy rules and business regulations. In many administrative tasks there is a need to comply with existing legislation or business policy rules. These rules must be enforced in the business processes. However, knowledge derived with data mining algorithms from the databases may be incompatible with such rules.
- Lack of interpretability of the model. Managers often require that the final model is easy to understand and in general do not accept black-box models. Quite often, it is more important to gain insight in the decision problem, than to have accurate predictions.
- Knowledge representation at the wrong level of detail. Data mining algorithms often yield structures or models that are intractable for human decision makers due to their complexity.

Consequently, there is a growing interest in integrating the traditional data mining software, which derives knowledge purely from the data alone, with more descriptive approaches for encoding domain knowledge or meta-knowledge, that guide the search process. There is a great scope for integration of knowledge based on experience and intuition of domain experts (or knowledge from other sources) encoded in some accessible way, with knowledge derived from conventional data mining algorithms (Feelders et al, 2000).

Here we will develop an approach for the integration of expert rules based on the knowledge of human domain experts and knowledge implicitly present in cases stored in a database. This approach can be applied to a large variety of cases such as:

- Risk assessment in the presence of both qualitative knowledge and legal or contractual constraints.
- Classification and description of customer groups in evaluation decision processes, such as credit loan evaluation, risk-assessment and fraud detection.
- All kinds of price models for trend analysis or automatic trading employed in combination with transaction databases.

The remainder of the paper is organised as follows. In section 2 a general overview of the type of knowledge that can be combined with data-mining systems is given. In section 3 we focus on risk management models. It is explained how normative knowledge and knowledge of domain experts can be combined to assign a risk score to artifacts like claims, loans etc. Section 4 deals with the implementation of the monotonicity constraint in the risk management model. In section 5 the results of our approach in a case study in insurance are presented.

2. TYPES OF KNOWLEDGE

In the data mining literature the notions of domain knowledge, background knowledge, and prior knowledge are commonly used to denote different types of knowledge. We make a broad distinction between:

- Normative knowledge about the model to be constructed.
- Knowledge about the data generating processes.
- Knowledge reducing costs and improving search efficiency.

Normative knowledge may be important if the objective of data mining is to find a decision model that will be used for decision making, for example in acceptance/rejection decisions. Usually, expert knowledge is available about which factors are important to take into account in the decision model. This knowledge is often based on experience of experts and can be tacit or encoded. A common sense requirement for decision rules derived from such knowledge is that the decision rule should be monotonic with respect to certain variables. For example, in loan acceptance the decision rule should be monotone with respect to income, because it is not acceptable that an applicant with high income is rejected, whereas another applicant with low income and otherwise equal characteristics is accepted. Similarly, in the case study presented in this paper (section 5) the classifier should assign higher risk to cases for which more indicators apply, i.e. the risk is monotone with respect to the indicators.

Knowledge of the data generating process, which is often called data expertise, is also an important type of domain knowledge. Data expertise is required to explain strange patterns and remove pollution, e.g. caused by data conversion or merging of databases. For example, in a case where a large insurance company took over a small competitor, the insurance policy databases were joined. The start-date of the policies of the small company were set equal to the conversion date, because only the most recent mutation date was recorded by the small company. Without the knowledge of conversion, one might believe that there was an enormous "sales peak" in the year of conversion (Feelders et al., 2000).

The last category mentioned concerns the trade-off between the cost of measurement of variables and the gain of obtaining additional information. Such trade-offs occur frequently in the context of medical diagnosis (Summons, et al., 1998). In that case one would like to consider both the amount of information and cost of measurement of a variable in model construction. The hierarchical structure of the knowledge domain can often be applied to derive rules at a higher aggregation level. This increases the efficiency of the search process and improves the transparency of the model.

In the next section we discuss how expert rules of thumb can be combined with neural networks. This yields a flexible architecture that is applicable to a wide range of problems.

3. COMPUTATION OF RISK SCORE

In most expert systems knowledge is stored in so called production rules. The rules correspond to chunks of articulated expert knowledge. The usual standardized form in which the rules are encoded is the CNF (conjunctive normal form) syntax. Each rule consists of a IF and THEN part:

R: **IF** (A or B or …) and (C or D or …) and … **THEN** RHS

Here A,B,C,D etc. are predicates that contain variables of the domain and RHS is the right hand side of the rule which stands for a conclusion of the rule. In expert systems the rules interact with the user in a reasoning process and may reach conclusions just like experts can do in the domain of expertise.
In the case discussed in this paper (section 5) the production rules correspond to economic risk indicators such as indicators of fraud. We will show that rules of this form can be combined with historical cases in a database to construct a model for risk scoring.
Suppose the rules correspond to risk indicators articulated by experts and are numbered R_1,R_2,R_3 etc.. These rules can only yield the result true or false when applied to a certain case. *The more rules apply to a case, the larger the risk.*
In general, experts do not know how the risk indicators should be combined to obtain a final risk score. Therefore, the individual risk indicators make up the input of a neural network that computes the total risk score. The network is trained on the patterns in a database for which the risk is known. If the rules are sound and the patterns in the database do not contain too much noise, the computed risk score approximates the real risk (figure 1).
In many applications it is required that the risk score depends monotonically on the risk indicators. This monotonicity property is important in many economic decision problems and is further explained in the next section. Consequently, the neural network should be constructed in such a way that this constraint is satisfied (Daniels and Kamp, 1999).

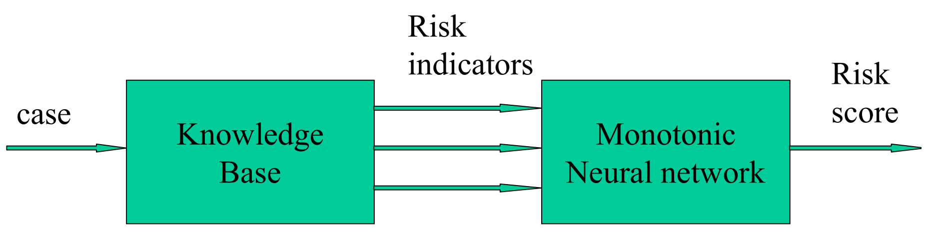

Figure 1. Risk score module.

4. MONOTONICITY

In many economic regression and classification problems it is known that the dependent variable has a distribution that is monotonic with respect to the independent variables. For instance, economic theory would state that people tend to buy less of a product if its price increases (ceteris paribus), so there would be a negative relationship between price and demand. Another well-known example is the dependence of labour wages as a function of age and education (Mukarjee and Stern, 1994). The strength of the relationships and the precise functional form are not always dictated by economic theory.
In cases where we are dealing with a risk management problem like in this paper we want to derive a classification rule C(R) that assigns a risk class (score) to each subset R of risk indicators. Monotonicity of C is then defined by:

$$R^1 \geq R^2 \Rightarrow \mathrm{C}(\mathrm{R}^1) \geq \mathrm{C}(\mathrm{R}^2)$$

Here $R^1 \geq R^2$ means that R^2 is a subset of R^1.
In the architecture depicted in figure 1 this property is guaranteed by the monotonicity of the neural network (Daniels and Kamp, 1999; Wang, 1994). Similar concepts have also been studied in the context of decision trees (Ben-David, 1995; Nunez ,1991).

5. CASE STUDY

5.1 The WBF foundation.

The case study described below is typical for fraud in insurance firms. Car insurance policies cover damage inflicted by motor vehicles. In most cases the victim of an accident claims against the insurance company of the liable driver. However, in practice there are many accidents were the liable driver cannot be traced even after police investigation. In those cases people may submit a claim to the Waarborgfonds Motorverkeer (further referred to as "WBF"). WBF is a special insurance foundation in the Netherlands, which deals with accidents were the liable party is unknown. They cover various types of damage, such as material damage, personal injury and even damage to the environment. The foundation is financially supported by all insurance companies in the Netherlands. Each year the WBF handles about 60,000 claims.
For regular insurance companies the average fraud ratio is around 8%. For the WBF this fraud ratio is probably higher, since there is only one party that can be questioned for information. This fraud percentage is estimated by fraud experts to be around 12%. At the moment only 4% of the fraudulent claims are identified. A risk management system should increase this percentage.
Currently, employees that handle the claims are instructed by fraud experts to hand over cases that are suspicious. At the time the prototype of the risk management system was developed the process of deciding which cases are suspicious and which are not was rather intuitive and error prone. Therefore, the risk management system should automatically subdivide the incoming stream of claims in suspicious claims and non-suspicious claims. (figure 2). A claim is considered to be suspicious if the risk score is above a certain threshold (0.5 was suggested by the WBF).

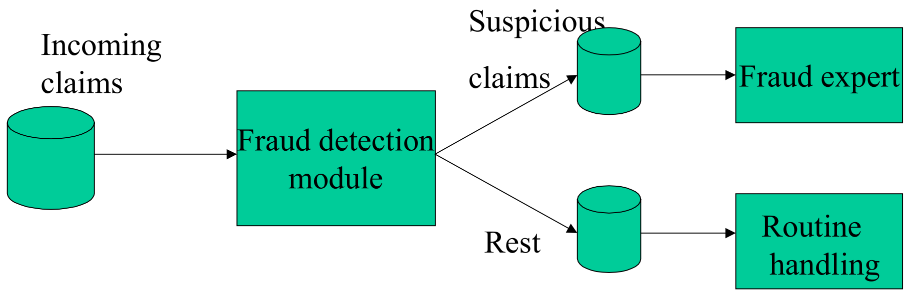

Figure 2. Flow of claims

5.2 Acquisition of risk indicators.

The risk indicators were collected by interviewing fraud experts. The experts would typically articulate a number of simple rules of the thumb, such as: “A young driver, driving a motor vehicle during the weekend, who had to give way to an oncoming car”. In CNF the rule reads:

IF driver.age < 25 AND accident.day = “FRI” OR “SAT” OR “SUN” AND cause = “GIVING WAY” THEN true ELSE false

This rule covers the typical case where a young male driver borrows dad’s or mum’s car in the weekend, to visit a disco with his friends. On his way back he misses the bend due to high speed and the car is severely damaged. He is afraid to inform his dad about the real cause of the accident and claims that an oncoming car forced him to give way and he subsequently landed in the verge. Normally this would be a case covered by the WBF, but experienced experts would consider it suspicious.

More complicated rules were derived by protocol analysis. Here the expert treats cases of proven fraud and writes down all fraud indicators that are applicable to the case at hand. Several risk indicators take into account information indirectly connected to the case, for example the claim history of the client, information about the vehicle involved and information about witnesses. The claim history of the client can be extracted from a special database, and if an exceptional high number of claims were recorded, this is considered an additional risk factor. The system also checks if the case at hand is connected to other claims. There exist so-called circular chains of claims, where the witness in one case is the claimer in the other.

The online connection of the rule base with databases is essential to achieve a good performance. In the first place to take into account all data with information value, and secondly to improve efficiency (in practice the manual consultation of databases is rather time consuming). In total 16 risk indicators are implemented in the prototype.

5.3 Database with claim records

The WBF has a large transactional database of claims processed in the past. Unfortunately, in many cases essential information is missing. For training the neural network of the system we selected a subset of 200 records of good quality. 100 being proven fraudulent claims and 100 most probably non-fraudulent.

In table 1 part of the input to the neural network is shown. The table shows 10 rows, that each correspond to a case. The complete table has 200 rows. The number 1 respectively 0 in the column indicates whether the corresponding rule applies or not. In the table only six of the sixteen identified rules are listed. The fraud index indicates if the case corresponds to a fraudulent claim (1) or not (0).

R4	R5	R12	R13	R14	R16	f-index
0	0	0	0	1	0	1
0	0	1	1	0	1	1
1	0	0	0	0	0	1
1	1	0	0	0	1	1
0	1	1	0	1	0	1
0	0	0	1	0	0	0
0	0	1	0	0	0	0
1	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	0

Table 1:The outcome of six rules applied to 10 cases.

5.4 Training of the neural network and results.

Using the same input data, we compared in a simulation study ordinary neural networks and monotonic neural networks with 5, 10, 15 and 20 hidden neurons in the hidden layer. In the training

process we used 5-fold cross-validation and the results reported are averaged over the 5 different subdivisions of the data. The output of the two neural networks is between 0 and 1, because we used a sigmoid activation function for the output neuron. For non-fraudulent cases the output is correct if it is <0.5 (then the risk score is set to 0). For fraudulent cases the output is correct if it is >0.5.
In table 2 (normal neural networks) and 3 (monotonic neural networks) the results of the simulation studies are shown. It is very clear that the performance of monotonic neural networks is much better. This is due to fact that normal neural networks have a tendency to overfit the data if the number of neurons is high. This tendency is suppressed in monotonic neural networks, without spoiling the learning capability.

Hidden neurons	5	10	15	20
Per.correct in-sample	38%	45%	59%	68%
Per.correct out-sample	40%	46%	46%	33%
R^2 train	0.71	0.74	0.77	0.83
R^2 test	0.69	0.62	0.53	0.48

Table 2 Performance of the normal neural network.

Hidden neurons	5	10	15	20
Per.correct in-sample	32%	43%	57%	66%
Per.correct out-sample	37%	46%	52%	67%
R^2 train	0.66	0.69	0.75	0.80
R^2 test	0.64	0.61	0.72	0.78

Table 3 Performance of the monotonic neural network.

6. CONCLUSIONS

The goal of data mining is to derive valuable business knowledge from patterns in databases. In the majority of cases there is theoretical and domain dependent knowledge available. In this paper we have shown that the effectiveness of data mining systems can be substantially improved as compared to data mining systems based on blind search only, by including normative knowledge about the model to be constructed and knowledge of experienced domain experts. We explicitly studied this approach for risk management problems with a case study in insurance.
The advantage of the approach is twofold. First, the otherwise blind search in databases is now guided by expert experience leading to substantially improved results. Second, since in general experts find it difficult to combine decision rules into a single risk score, the framework discussed offers the possibility to combine and fine-tuned expert knowledge using real cases.

ACKNOWLEDGEMENT

Arno van de Camp is kindly acknowledged for his contribution on knowledge acquisition and implementation.

REFERENCES

Ben-David A. (1995). Monotonicity Maintenance in Information-Theoretic Machine Learning Algorithms. *Machine Learning*, **19,** 29-43.

Daniels H. A. M. and B. Kamp (1999). Application of MLP networks to bond rating and house pricing. *Neural Computation and Applications*, **8**, 226-234.

U Fayyad, S. Piatetsky, P. Shapiro, P. Smyth and Uthurusamy (1996). Advances in knowledge discovery and data mining. AAAI Press.

Feelders, A., H.A.M. Daniels, and M. Holsheimer (2000). Methological and practical aspects of data mining. *Information & Management*, **37**, 271-281.

Han J. and M. Kamber (2001). Data-Mining: Concepts and Techniques, Morgan Kaufmann Publishers.

Mukarjee, H. and S. Stern (1994). Feasible Nonparametric Esimation of Multiargument Monotone Functions", *Journal of the American Statistical Association*, **89**, 425, 77-80.

Nunez, M (1991). The Use of Background Knowledge in Decision Tree Induction, *Machine Learning*, **6**, 231-250.

Summons, P., G. Gibbon, H. Villanueva and J. Falconer (1998) Causal reasoning in a medical knowledge based system. In: *Proceedings of the 6th European Conference on Information Systems*, Aix-en-Provence, France, June 4-6, 1154-1165.

Wang, S.(1994). A neural network method of density estimation for univariate unimodal data, *Neural Computation & Applications*, **2**, 160- 167.