Georg-August-Universität Göttingen
Institut für Wirtschaftsinformatik
Professor Dr. Matthias Schumann

Platz der Göttinger Sieben 5
37073 Göttingen
Telefon: + 49 551 39 - 4433
+ 49 551 39 - 4442
Telefax: + 49 551 39 - 9735
www.wi2.wiso.uni-goettingen.de

Arbeitsbericht Nr. 01/2002
Hrsg.: Matthias Schumann

Yang Liu

A framework of data mining application process for credit scoring

Contents

Figures

Tables

1 Introduction

The credit scoring models involve the techniques that are called today the techniques of data mining. Classification methods are the most commonly used data mining techniques that applied in the domain of credit scoring to predict the default probabilities of credit takers. Many methods, such as linear and logistic regression, decision trees, neural networks, etc. have been used for developing credit scoring models. The search for commercial advantage in the credit industry has led to interest in a new and emerging technology --- data mining.

Data mining as an approach to support computer-based decision making is actually not a purely new technology but borrows many algorithms from statistics, artificial intelligence and other fields. It is not the algorithms of data mining but the idea of automatically getting knowledge from large databases is revolutionary. Nowdays, large amount of clean and well-documented data in organizations and more cost-effective IT solutions in terms of storage and processing ability make this idea more realistic: new algorithms from research centers and universities are able to enter into commercial software (cf. Cabe98, P. 11). Although the implementation of totally automatic knowledge discovery from database is still far away from expected ideality, this new concept and the continuous research endeavors on it give the opportunity for the future's revolution in computer-based decision making.

The research of data mining necessarily involves many different areas, including the background areas of different data mining techniques like statistic, machine learning, the areas of computer science like database, parallel computing aiming at assisting and speeding data mining. In this paper data mining is considered as a decision support process that enable users to solve business problems (specifically, credit scoring problem). From this view the process of data analysis by employing data mining approach is mainly concerned. Research on the process of data mining is much rarely mentioned comparing with other aspects of data mining. However, the successfulness of a data mining application in practice sometimes depends decisively on the strategies of controlling the data mining process.

In the view of data mining, the simply using of one or several algorithms on the available data set is substitute by a complex time consuming process, which is full of trials and iterations. With the maturity of the mining algorithms and the best understanding of the problem domains, the most challenging is how to control the mining process. The control of mining process is related to but still beyond the mining algorithms and the domain knowledge. In this sense, the aim of data mining is not only the good performance of particular algorithms, but also to get the most applicable results with the least time and cost.

There are some essentially similar descriptions on the process of data mining in literatures. Most of them gave a list of steps of the data mining procedure which lacks for detailed discussions of how these steps to be accomplished: which mining techniques are involved,

which evaluation criterion are used to determine whether going to the next step or iterating to earlier steps. Some other research in literatures presented some empirical studies that give many tricks for some small steps in the mining procedure. They focused only on the specific data set and specific mining techniques.

Credit scoring can be looked as the type of classification problem of data mining. Meanwhile, its practical applications associate many problems relevant to the credit industry. Due to the complex decision process credit scoring has always been based on a pragmatic approach: A solution can not be the optimal one for everywhere, only for specific circumstances. The process of credit scoring is not standardized. A serious problem with this nonstandard model building process is an aimless, repeated and expensive data analysis process that cannot yet guarantee an optimal model solution.

Data Mining approach not only utilizes a multi-strategy combining multiple statistics and machine learning algorithms of classification, but also provides a framework for data analysis process, which includes necessary pre- and post-processing of the real-world large data sets in order to support a standard model building process.

This paper aims to introduce the data mining concept, especially the classification problem of data mining and develop a systematic data mining process framework that is applied particularly on the credit scoring problem.

In Chapter 2 the concept of data mining is introduced. A general data mining process is given, which will be specialized in Chapter 5 for the problem of credit scoring. It is denoted that as one of the application areas of data mining, the credit scoring problem is related to the classification decision. In Chapter 3 the classification problem is specially introduced. The professional backgrounds and the approaches of model generation are explained in order to give a deep understanding of the classification model building. In chapter 4 five classification algorithms are described. After that a comparison is given for the introduced methods allowing us to identify the relative advantages and disadvantages of these methods and their applicability in credit scoring. In Chapter 5 an overall framework of data mining process for the problem of credit scoring is formed. The general framework is described with diagrams and explanations.

Summarization and conclusions are given in the final. Proposals for further research are also pointed out.

2 Data mining

2.1 The concept of data mining

Using machine learning, statistical analysis, and other modeling techniques, the patterns and relationships in data can be found. The activities to discover hidden knowledge contained in data sets have been attempted by researchers in different disciplines for a long time.

By the end of the 1980s, a new term, Knowledge Discovery in Databases (KDD), was coined (cf. Fraw91, P. 1) and quickly adopted by artificial intelligence and machine learning practitioners to cover the overall process of extracting knowledge from databases, from setting the business goal to eventual analysis of the results. In this context, the word "data mining" was used for one step in the KDD process --- the step when the mining algorithms were applied. This interpretation was formalized in 1994 (cf. Fayy96, P. 6). Recently, as a result of the increasing attention of vendors and the popular trade press in this area, the word "data mining" has been used and has come to mean, like KDD, the overall process of extracting knowledge from databases (cf. Cabe98, P. 14-15).

This paper adopts this recent interpretation of data mining. This interpretation emphasizes that data mining is not just a set of mining algorithms, but rather a process: A process that aims at solving a definite problem or making a decision, utilizes various mathematical and computer techniques to analyze the relevant data stored in large databases, finds a solution based on the discovered patterns in data and applies the solution to the predefined problem.

Data mining is not a new term, which has been used for a long time, especially by statisticians (cf. FaUt95, P. xiii). But the idea of extracting knowledge from database has revolutionary meaning for modern enterprises. In order to make use of the data to facilitate business decision making, more and more enterprises are building their data warehouse by reorganizing their stored operational data and bringing in other external data. An ideal expectation of data mining technology is to automatically discover decision supporting knowledge from the huge data saved in the large databases.

Some data mining definitions contain the term "automatic", such as, "data mining is a set of techniques used in an automated approach to exhaustively explore and bring to the surface complex relationships in very large datasets" (cf. Moxo96). This is likely to be misunderstood as implying that answers will magically appear when a mining tool is applied to a database. However, the overall process of data mining is far away from "automatic", but involves so many human interventions that some authors regard the process of data mining as a combination of art and science (cf. WeIn98, P. 21).

2.2 The general process of data mining

To clarify what is the root of 'automatic' in the concept of data mining, the overall process of data mining with three stages is examined (cf. Figure 2.2/1).

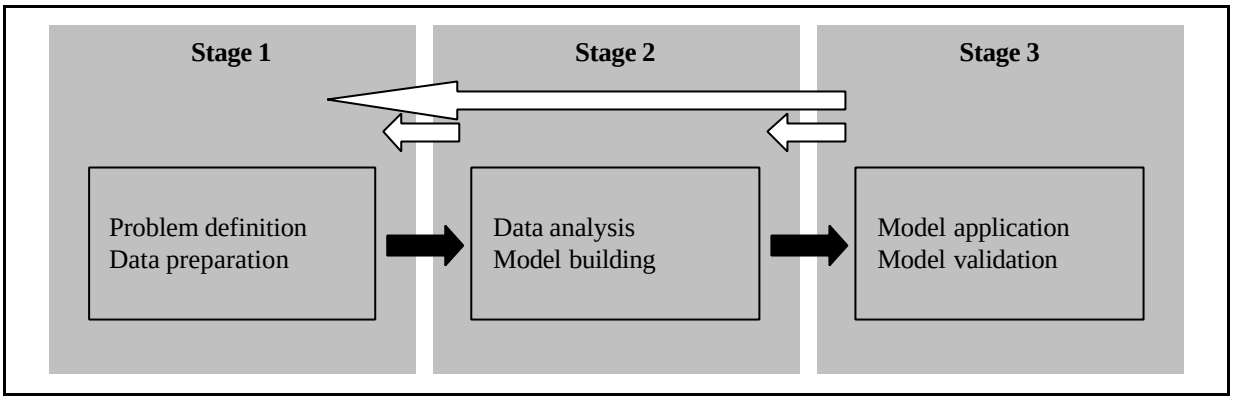


Figure 2.2/1: The overall process of data mining

At first stage, the business problem is identified to determine the goal. Based on this goal relevant data are prepared. They can either be drawn directly from the available data in the enterprise data warehouse or collected from different sources. Anyway, they must be transformed into the standard form that is acceptable to the techniques of data mining. At second stage, various data mining techniques are used to analyze data and build models. In this stage, an iterative process is carried out to find the best solution to the defined problem. At last stage, the found solution is applied in practice and its effect is validated.

Data mining may be a repeated process in practice, when for example, the data are not up-to-date, new data should be collected (return to stage 1), or when incremental techniques are used, models will be rebuilt with some incremental samples (return to stage 2). In the applications in the financial area, this repeated process is more often due to the dynamic aspects of financial data.

Stage 1 and stage 3 are application specific and certainly cannot be automatic. The ideal objective of data mining is to realize the automation in stage 2.

Techniques used in stage 2 are divided into two classes: core algorithms and peripheral techniques (cf. Figure 2.2/2). Core algorithms are used to create models. Peripheral techniques are used to preprocess the input data, to illustrate and evaluate the obtained models.

The objective of stage 2 is to create optimal models for the defined problem. Which solution is optimal depends on the defined problem and the application-oriented evaluation criterions. The optimal models cannot simply be obtained by applying a model algorithm to the data set. A wide variety of peripheral techniques and tools are used in order to get final model,

including any pre- and post-processing techniques that can help to identify relationships in a data set and evaluate the obtained results, even some queries and descriptive statistical methods for data summarization and description.

Usually, "automation" occurs in the core algorithms, "automatic" means only that some of the core algorithms can discover patterns and relationships through a data-driven approach without any assumptions about the data (normality, linearity, etc.) (cf. Chapter. 3.3.2).

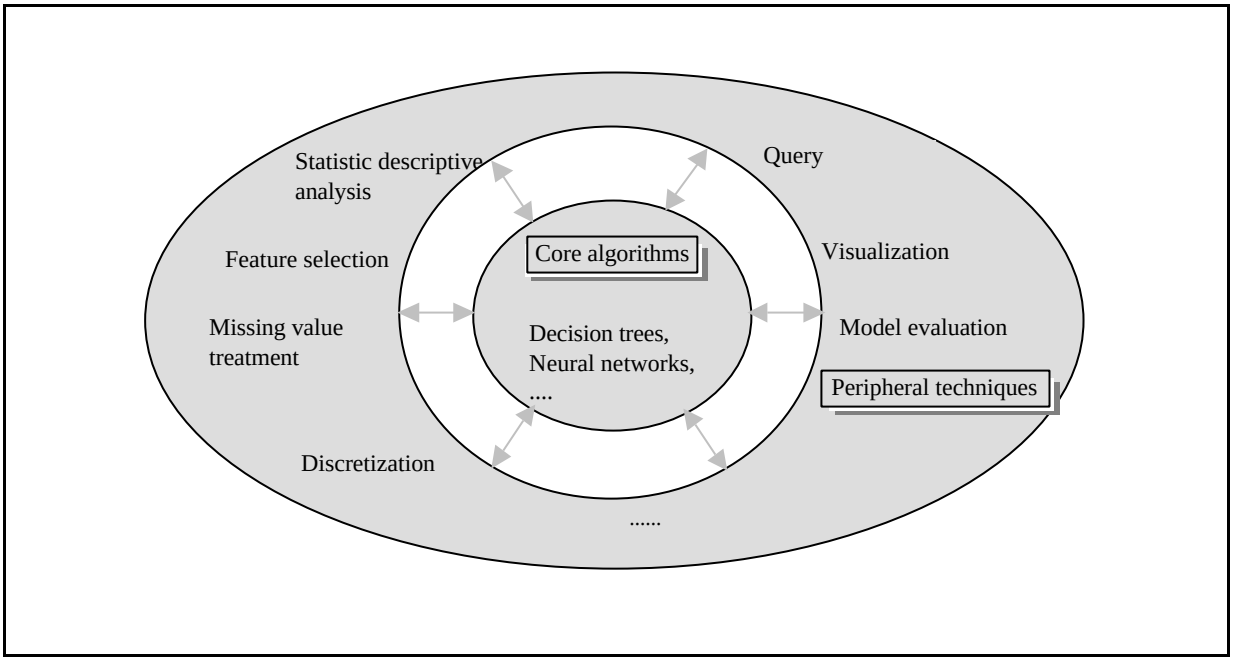


Figure 2.2/2: Core algorithms and peripheral techniques

An overall automation in data mining is till now far away from realization. It has been seen that new developed data mining algorithms try to automatize some pre-processing tasks, e.g. some decision tree algorithms (cf. Quin93) can automatically treat missing values in the data. However, most of peripheral techniques are separated from core algorithms. In order to get more reliable results any data mining applications should unavoidably experience a process that consists of many steps. Various techniques need to be used in each steps and many judgments need to be made during this process, e.g. whether a kind of peripheral technique should be applied and which techniques is most suitable. To make these judgments not only experiences are required but also many time-consuming experiments are necessary. Since new techniques appear continuously, the uses and choices of these techniques during the data mining process are becoming more difficult.

2.3 The application architecture of data mining

Data mining applications cover many aspects of problems. A three-layer application architecture can describe the multiplicity of data mining applications (cf. Moxo96; Boeh97, P. 33). There are many business areas that can adopt data mining technology, such as, credit

evaluation, database marketing, customer relationship management (CRM), etc. The type of problems in the data mining applications can be categorized into classification, clustering, deviation, association, etc. To solve these problems many core algorithms and peripheral techniques are used (cf. Figure 2.3/1).

The gray areas in Figure 2.3/1 show the research range of this paper: the application of data mining classification techniques in the area of credit scoring. For credit scoring, the relevant problem is classification decision. Banks and other credit institutions usually classify credit customers into several risk levels based on the relevant information about them. A credit decision will be made according to the risk classification. For example, the risk class of a new credit applicant decides whether the application will be approved; the risk class of an existing credit taker predicts its future default behavior and decides what actions should be taken to prevent or reduce the expected loss.

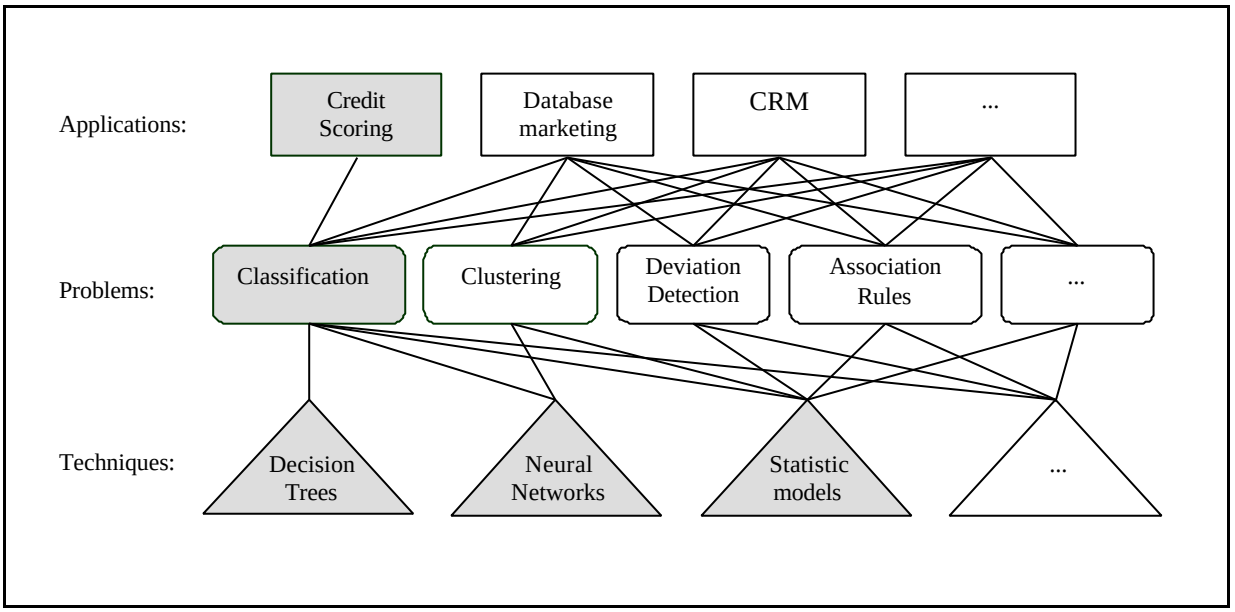


Figure 2.3/1: The application architecture of data mining

3 Classification decisions

3.1 Two ways of computer-based classification decisions

The task of classification decision simply refers to the process of assigning things into one of multiple categories or classes based on their properties. To make this classification decision, people almost always rely on prior experiences. These prior experiences can be obtained in two very different ways --- inductive and deductive, which lead to the following two approaches of computer-based classification decisions (cf. WeKu91, P. 3-4).

The deductive way is implemented by knowledge-based system, sometimes also called expert system (cf. FrSc01, P. 722). The decision is made automatically by computer system based on the knowledge obtained by interviewing the relevant experts.

In the inductive way, classification models are constructed inductively by generalizing from numerous recorded specific examples, i.e., by discovering and analyzing patterns found in prior solved cases (cf. Quin93, P. 1). The models then can be used to decide the class membership of an unknown case. Constructing the classification models in the inductive way is one of the main tasks of data mining.

The basic differences between these two approaches are shown in Figure 3.1/1. The knowledge in a knowledge-based system is acquired through what is called 'knowledge engineering' and saved in the system, then the decision for new cases are made deductively according to the knowledge saved in the system, which are usually in the form of a series of 'if.. then...' rules. The decision making is supposed to be totally automatic with only little interface by users in special cases. Data Mining approach discovers the knowledge inductively from data, the form of the knowledge are models or classifiers, which may be in different forms, such as decision trees, or trained neural networks, depending on what kind of classification technique is used. The overall process of data mining is computer-supported, but since completely automatic knowledge discovering is still not realized (cf. Chapter 2.2), the assistance and interface of domain experts during the process of data mining are necessary.

Both approaches of classification decisions have their limitations. Knowledge-based systems are often criticized for being limited in their abilities to surpass the level of existing experts. Another often cited problem is the great effort required to build and maintain a knowledge base, and the shortage of trained knowledge engineers to interview experts and capture their knowledge in a set of decision rules or other representational elements. This process, known as knowledge acquisition, is quite time-consuming, leading to lengthy development, which

must be continued if the system is to be maintained in routine use with a high level of performance.

Inductive classification is restricted with the prediction accuracy due to some deficiencies in the learning data (cf. chapter 3.2.2). Moreover, the supports of experts are necessary with the selecting of samples and with the interpretation of results. The automatically discovered rules are not always reasonable and therefore, often need to be reviewed and revised by the domain experts (cf. FrSc01, P. 722). The argument in favor of inductive learning from collections of samples of solved cases is that they might exceed the performance of experts because they have the potential to discover new relationships by examining the record of successfully solved cases. Besides, the process of learning automatically holds out the promise of incorporating knowledge into the system without the need of a knowledge engineer[1].

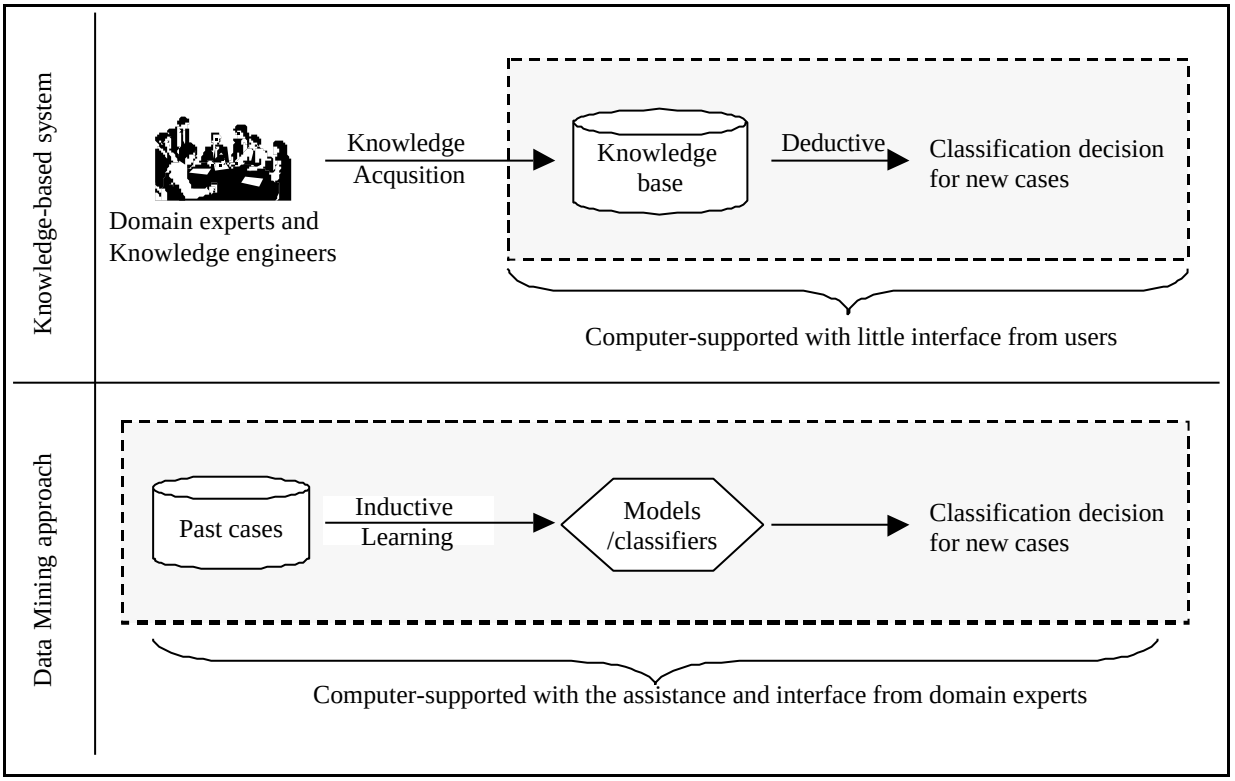


Figure 3.1/1: Deductive and inductive classification decision making

The two choices have their own characteristics and can be competent for different situations. For the decision problem that domain experts have confirmed experiences which are easily identified and expressed in formal rules, and the decision rules are stable with the varying of time, the knowledge-based system can be implemented and maintained effectively. On the

[1] It is worth remarking here that some of inductive learning methods can be used as tools for knowledge acquisition.

contrary, the inductive data mining approach can be addressed to the classification problems in which large amount of past examples with known classes are available. Besides, the experiences from experts are incomplete or not verified, or not easy to be expressed in a formal language.

Both computer-based classification approaches have been applied to the credit evaluation problem. Some knowledge-based systems have been built to support the examination of credit worthiness and credit rating (cf. Müll96, P. 81-130). The application of inductive methods is usually termed as credit scoring, which is popular in the area of consumer credit and small business credit.

3.2 The inductive classification

3.2.1 Description of the classification procedure

In the rest of this paper, classification refers to inductive classification. The problem of classification in this sense has been studied extensively by the statisticians as well as the database and artificial intelligence communities. In statistics the classification problem is sometimes called the prediction problem, and in the field of machine learning it is often called supervised concept learning (cf. WeKu91, P. 4), since it adjusts the parameters of learning models according to the known value of output, i.e., the learning process is guided by the provided examples. It is distinguished from unsupervised learning or clustering in which the classes are inferred from the data[2].

The classification procedure is described as follows:

The input data is referred to as the training set, which contains a plurality of records (also called examples, cases), each of which contains:

1. a known class label.
2. multiple variables (also called features, predictors), which, presumably, contain sufficient information to distinguish among the classes.

The training set is used in order to build a model of the class variable based upon the other variables (cf. AgYu99, P. 19). The model is then used to predict the class label of future cases (or the probabilities that they belong to a class). The classification problem concerns the construction of a model (also called a classifier) that will be applied to a continuing sequence of cases, in which each new case must be assigned to one of a set of pre-defined classes on the basis of observed variables or features.

Figure 3.2.1/1 illustrates the procedure of classification simply with the situation where only two variables X_1 and X_2 are used to predict a case's future class, which is either 1 (' bad' customers) or 0 ('good' customers). These two variables define axes of the sample space. A sample space is provided in Figure 3.2.1/1-a, in which data are plotted. The purpose of classification is finding boundaries that encompass only examples belonging to a given class (see Figure 3.2.1/1-b). If the relationships between the predictors and the classes remain constant in the future, accurate future decisions can be made by these boundaries. Suppose the examples in Figure 3.2.1/1-a are partitioned as in Figure 3.2.1/1-b. In this illustration, the new case plotted as "?" would be predicted to belong to class 0 ('good' customers) (cf. KaCo97, P. 468).

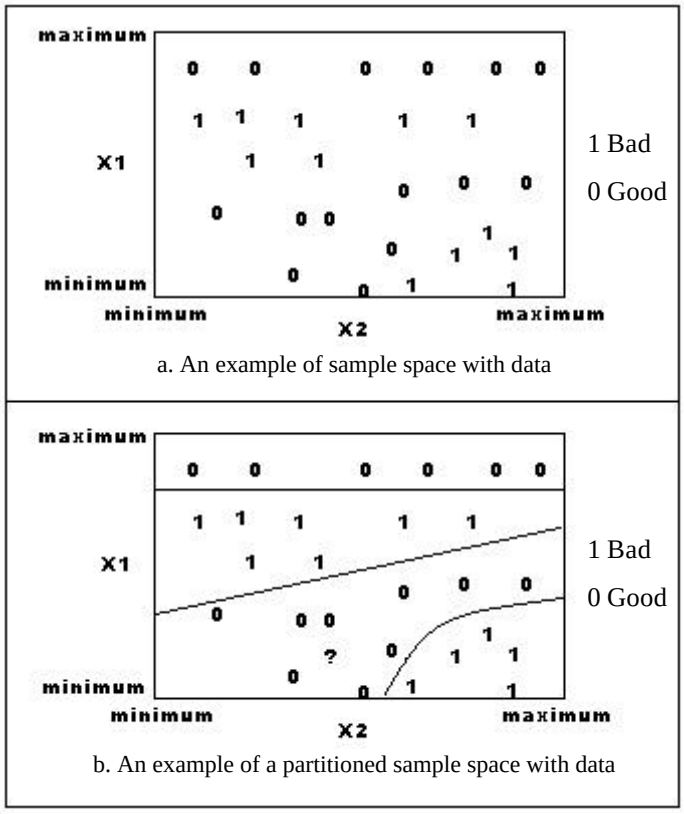


Figure 3.2.1/1: Illustration of the procedure of classification (cf. KaCo97, P. 468)

3.2.2 Reasons for inaccurate classifications

Many approaches have been researched and applied to find accurate boundaries to classify the examples. However, there are strong practical reasons to expect that the absolutely

[2] The construction of a classification model from a set of data for which the true classes are known has also been variously termed as pattern recognition, discrimination. Other authors, especially those in the community of machine learning, have referred to these techniques as inductive learning, empirical learning, or case-based reasoning (cf. Mich94, P. 1).

accurate classification does not exist. There are three problems in practice to prevent perfect predictions by classification learning:

- Samples: On one hand, the samples with known class are limited. Classification learning is based on the previous samples with known class which are called learning data. If we know the classes of all possible points in the feature space, for example, the class of a case is decided by feature X_1 and X_2. If unlimited learning data are available, every point in the two dimensional space has a known class, each of them could be stored in a table, and for a new case one would simply look up in the table the corresponding class of the same previous sample (cf. WeKu91, P. 7). But this is unfortunately impossible in real world, the samples with known class are often quite restricted in practice. On the other hand, the collected samples are often not representative for the population to be analyzed. The performance of the learned classifier with the unrepresentative samples can not be good. Figure 3.2.2/1 shows how limited and unrepresentative samples prevent accurate predictions. The ideal classification boundary is a simple straight line. But if we use the sample points with circles which are limited and unrepresentative, learning algorithms will find an inaccurate classification boundary, i.e., the prediction on new cases with this boundary will be inaccurate.

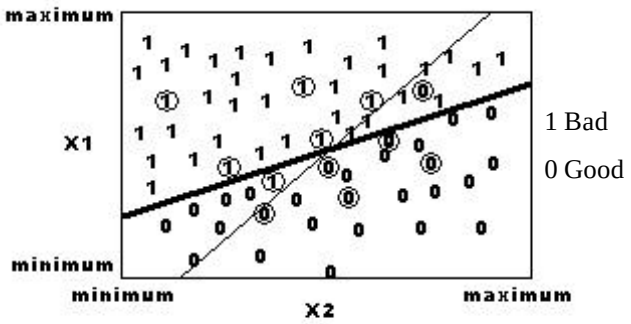


Figure 3.2.2/1: Inaccurate prediction with limited and unrepresentative samples

- Features: The predictive capability of the features is fundamental to the success of any learning system (cf. WeKu91, P. 11). In real-world decision making, the boundaries of classes are often overlap. From an analytical perspective, this means that it is quite possible that similar or even identical samples of prior cases may fall into different classes, i.e., there may be ambiguity within the samples. If many of the samples are ambiguous for a given set of features, that may suggest that these features have poor predictive power, and no good solution to the classification problem may be possible with them alone (cf. WeKu91, P. 8). Figure 3.2.2/2 shows that for the given samples no

accurate classification boundary can be found. The two features X_1 and X_2 might be not relevant with the predicted class or only these two features might be incomplete, other features must be included in order to find accurate classification boundaries. Consider the firm with bankrupt-appearing characteristics but does not go bankrupt. This may happen when important characteristic (e.g. an economic up-turn) was not a dimension of the sample space and thus was not incorporated in the decision model (cf. KaCo97, P. 469). If the classes are not well separated by the available features, the probability function of Pr(insolvent risks | feature vector) is rather flat function, so that the decision surface separating the classes will not be accurately estimated. In such cases, some highly flexible methods are vulnerable to over-fitting the training data (cf. HaHe97, P. 535).

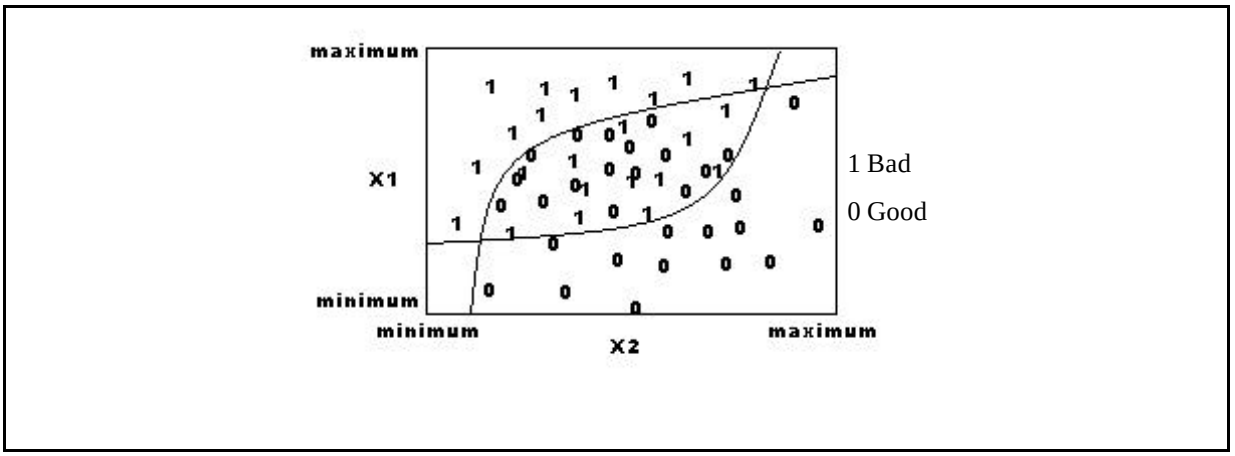


Figure 3.2.2/2: Inaccurate prediction with overlapped boundaries

- Class definition: The uncertain definition of the class membership of the samples adds the complicacy of the classification learning. The basic requirement of the classification methods is that the data be presented in the form of samples composed of features with the correct classification. In many applications, the correct classification is not absolutely known. In credit scoring, for example, the correct risk class may become known after a period of time, which is called 'outcome period' (cf. Liuy01, Chapter 4.2). The endpoint of 'outcome period' may not be certain, and an expert opinion must set the endpoint (i.e., the assumed correct conclusions).

In brief, the classification learning can not give a perfect prediction because of the above mentioned practical problems. Actually, what can be learned directly from the sample data alone is limited, if it ignores the context within which problem solving is carried out (cf. WeKu91, P. 3). The potential for successful classification depends on not only the classification techniques but also on the data samples selected for analysis, where domain experts can play an important role.

3.3 Various groups of classification techniques

3.3.1 The professional backgrounds

A wide variety of approaches has been taken towards the task of classification. These have largely involved different professional and academic groups, and emphasized different issues. Three main historical strands of research can be identified: statistical, machine learning and neural networks (cf. Mich94, P. 2-3).

Statistical approaches

In the statistical community two main phases of work on classification can be identified. In the first, “classical” phase, researchers concentrated on derivatives of Fisher’s early work on linear discrimination. In the second, “modern” phase, more flexible methods appeared, many of which attempt to provide an estimate of the joint distribution of the features within each class, which can in turn provide a classification rule. K-nearest-neighbors method is a kind of modern statistical classification approach.

Some statistical methods assume an explicit underlying probability distribution, e. g. linear discriminant analysis. In addition, some human intervention is assumed with regard to variable selection and transformation.

Regression models can be thought to be a kind of statistical classification approach. They provide continuous outputs rather than just a simply classification, e. g. logistic regression provides a probability of being in each class.

Machine learning

Machine Learning methods are generally the automatic procedures based on logical or binary operations. It is arguable which classification techniques should come under the Machine Learning umbrella. Attention has focussed on decision-tree approaches and rule induction algorithms, in which classification results from a sequence of 'if... then...' logical decisions. Models learn a task from a series of examples, which are capable of representing the most complex problem given sufficient data. Other techniques, such as genetic algorithms and inductive logic procedures, are currently under active development and in principle would allow us to deal with more general types of data, including cases where the number and type of attributes may vary, and where additional layers of learning are superimposed, with hierarchical structure of attributes and classes and so on.

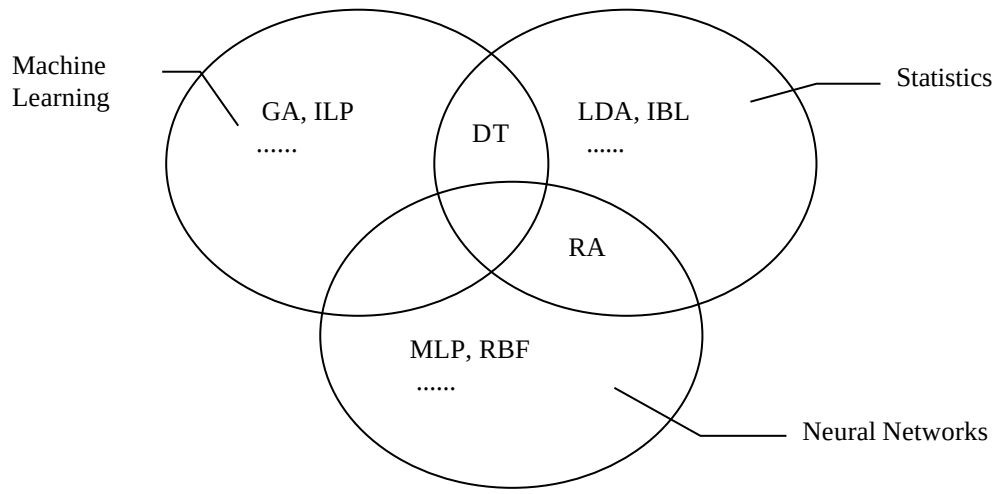
Machine Learning aims to generate classifying expressions simple enough to be understood easily by the human. They must mimic human reasoning sufficiently to provide insight into the decision process. Like statistical approaches, background knowledge may be exploited in development, but operation is assumed without human intervention.

Neural networks

The field of neural networks has arisen from diverse sources and been applied widely in different areas. A broad class of techniques can come under this heading, but, generally, neural networks consist of layers of interconnected nodes, each node producing a non-linear function of its input. The input to a node may come from other nodes or directly from the input data. Also, some nodes are denoted as the outputs of the network. The complete network therefore represents a very complex set of interdependencies which may incorporate any degree of non-linearity.

Neural network approaches combine the complexity of some of the statistical techniques with the objective of imitating human intelligence: however, this is done at a more uncontrolled level, the learning process is guided only by the data without any pre-assumptions. Hence it is difficult to make learned concepts transparent to the user.

A distinction is often made between supervised and unsupervised learning. The former describes the case where the training data are accompanied by labels indicating the class of event. Examples for the supervised learning networks are Multi-Layer Perceptron, the Cascade Correlation learning architecture, and Radial Basis Function networks. Unsupervised learning refers to the case where the data are not accompanied with class labels. Such models include Gaussian mixture models and Kohonen networks (cf. Mich94, P. 85).



GA: Genetic algorithms, ILP: Inductive logic procedures, DT: Decision trees, LDA: Linear Discriminant Analysis, IBL: Instance-based learning, RA: Regression analysis, MLP: Multi-layered perceptron, RBF: Radial Basis Function

Figure 3.3.1/1: Techniques and their professional backgrounds

The three groups of classification techniques are outlined by Michie et al. (cf. Mich94, P. 2-3). However, as mentioned by Michie et al. in their conclusion of this kind of categorization, the correspondence between type of technique and professional background is sometimes obscure. For example, techniques that use decision trees have been developed in parallel both within the machine learning community and within the statistical profession. Similarly strong parallels may be drawn between advanced regression techniques developed in statistics, and neural network models with a background in artificial intelligence. Therefore, when deciding on a group for a specific technique, its professional pedigree is sometimes ignored and classified according to its essential nature. The relationship of some techniques and their professional backgrounds are shown in the Figure 3.3.1/1.

3.3.2 The approaches of model generations

From the view of the approaches, by which the models are generated, the techniques can be classified into two categories (cf. Table 3.3.2/1).

Categories	Techniques	Algorithms
Verification-based (parametric) (theory-driven)	Regression	Linear regression...
	Discriminant Analysis	Linear Discriminant Analysis...
Discovery-based (non-parametric) (data-driven)	Decision Trees	C 4.5, M5...
	Neural Networks	Multi-layer propagation...
	Lazy Learning (Instance-based learning)	K-Nearest-Neighbors, Locally weighted regression...
	Evolutionary Computing	Genetic Algorithm...

Table 3.3.2/1: Two approaches of model generations for some classification techniques

Some of the classification techniques support a verification-based approach, in which the user hypothesizes about specific data interrelationships and then uses the tools to verify those hypotheses. In these techniques one has to specify the model and then the algorithm is used to determine the parameters of the specified model. These techniques can be classified as model-driven approaches, parametric methods or theory-driven methods (cf. Desa97, P. 324). The parametric methods usually utilize the idea of parameter estimation in statistics. For example, Linear Discriminant Analysis methods assume the normal form of the underlying population density distribution. Under this assumption the parameters of linear discriminant function can be estimated.

Other techniques in the contrary use discovery-based approaches in which pattern-matching and other algorithms are employed to determine the complex relationships in the data (cf.

Moxo96). Although each of these methods does assume a certain form of underlying model for the classifier or its learning capabilities, it not only fits the parameters of the model but often changes the structure of the model as the data feeding to it. The methods employ the power of the computer to search and iterate until they achieve a good fit to the data (cf. WeKu91, P. 13). Since they create the model automatically based on the patterns found in the data, they are also called non-parametric methods or data-driven methods, i.e. a pre-specification of the model is not required (cf. Desa97, P. 325). For example, an multi-layer propagation network 'learns' the relationships inherent in the data presented to it, and the generic algorithm provides a nonlinear classification function using a search procedure borrowed from natural phenomena.

The data-driven approach seems particularly attractive in solving the problem with little knowledge about the statistical properties of the data. Traditional theory-driven approach like statistical model development includes time-consuming manual data review activities such as searching for non-linear relationships and detecting interactions among predictor variables (cf. Desa97, P. 325).

On the other hand, the data-driven approach, which is better as generalizing complex non-linear data relationships, may produce models that can be relatively large, idiosyncratic, and difficult to interpret. For the data-driven approach, models are built only based on sample data, thus the model is easy to be over-fitted with the non-representative or noisy sample data, and the likelihood of poor generalization to new cases is greatly increased. The task then becomes to find the proper degree of model complexity that fits to the sample data without being over-specific to the sample. For example, the pruning of a decision tree to get a simple structure, the selection of a larger k in instance-based learning algorithms when the sample data are noisy. These are necessary to adjust the complexity of models to suit the problem at hand, otherwise the models will over-fit the sample data and will be biased.

Table 3.3.2/2 lists the main different aspects of data-driven and model-driven approaches.

	Data-driven	**Model-driven**
Identified relationships	complex, non-linear	straightforward
Model creation	automatic from data	hypothesized
Main task	model complexity adjusting	parameter estimation
Weakness	sensitive to noisy data over-fitting	time-consuming manual review incompetent for non-linear relations

Table 3.3.2/2: Comparisons of data-driven and model-driven approaches

Theoretically, some data-driven learning methods can form models that can classify the given data with 100% accuracy, in general at the expense of creating a very complex structure. In practice, however, complex structures do not always perform well when tested on unseen data, and this is one case of general phenomenon of over-fitting the data (cf. Mich94, P. 107). Figure 3.3.2/1 illustrates the problem of over-fitting. Suppose we get a classification boundary by a data-driven approach, which can discover non-linear relationship. The boundary can classify the sample data very well, with 100% accurate rate. But when it is used to classify new cases (represented as 1, 0), it has poorer ability even than a simple linear classification boundary.

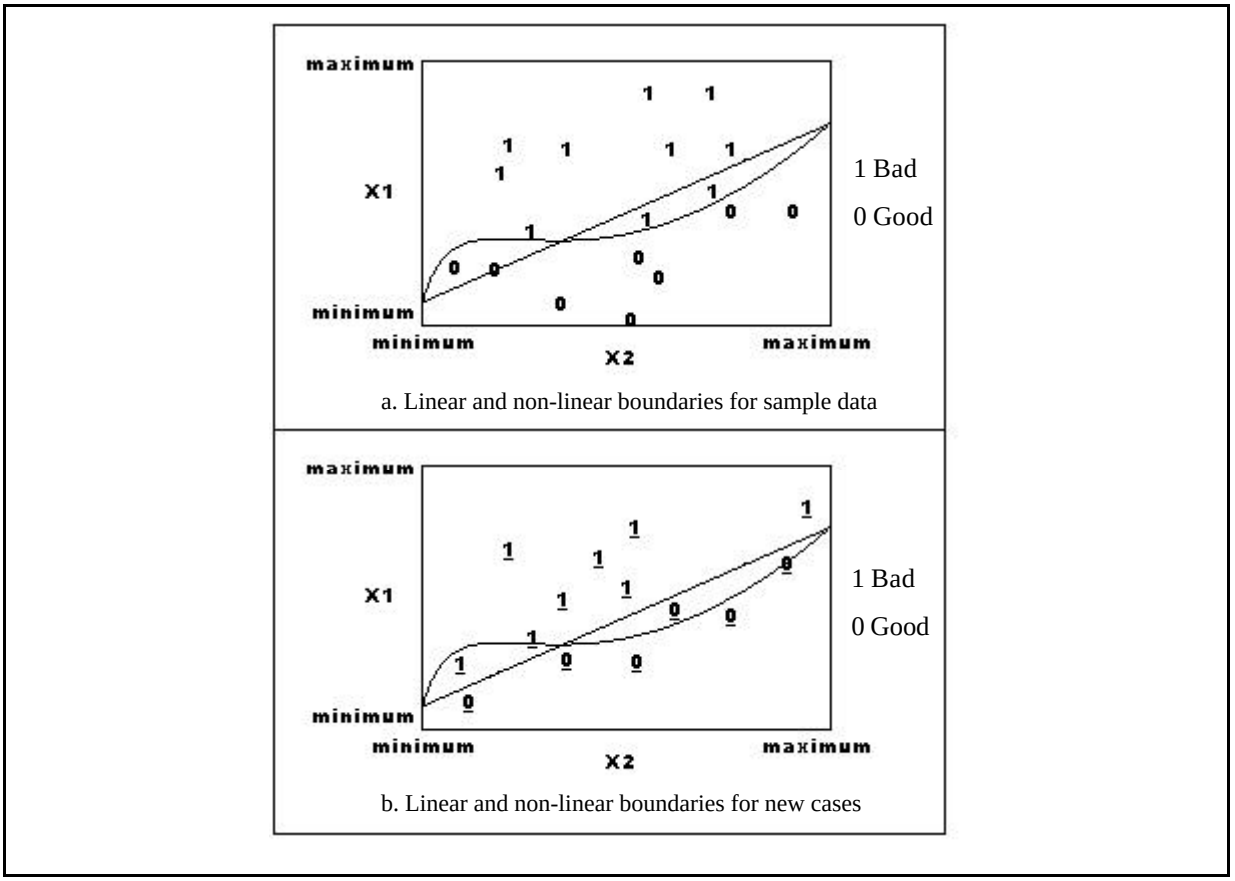


Figure 3.3.2/1: Over-fitting in data-driven approach

4 Various classification algorithms for credit scoring

This section provides an overview of five classification algorithms that have been applied in the credit scoring model. These techniques are chosen as examples due to their popularity, being representative for the available data mining techniques and their prevalence on the building of credit scoring models. Two statistical techniques (Linear Discriminant Analysis and Logistic Regression) are introduced because they are traditionally most frequently used and usually employed to benchmark the performance of other data mining techniques (cf. KaCo97, P. 469). Note that there are many current and evolving variations of the algorithms described below, this overview therefore focuses on and describes only one variant for each algorithm in order to simplify discussion. As the detail algorithms are not the main concern of this paper, only conceptual description of the algorithms are given. References are provided for the readers interested in more details.

Firstly, the used notations in the following discussions are given here:

1. $X = (x_1, x_2, ..., x_k)$: the feature vector of the cases in the given data set D, k is the number of features.
2. Case(X): a case whose feature vector has a given value X.
3. c_i, ($i=1,2,...,l$): the i_{th} class in the given data set D, assuming the cases in D can be classified into l classes.
4. $N_{ci,}$ ($i=1,2,...l$): the number of cases that belong to i_{th} class in the given data set D.
5. $Pr(X)$: the unconditional probability for a given case(X).
6. $Pr(X|c_i)$: the conditional probability of the case(X) given the class c_i.
7. $Pr(c_i)$: the priori probability. It represents the probability that any case will belong to class c_i.
8. $Pr(c_i|X)$: the posteriori probability. It represents the probability that a case(X) will belong to the class c_i.

4.1 Discriminant Analysis: Bayesian Linear Discriminant Analysis

Discriminant analysis falls into a category of classifiers, whose forms are discriminant functions. Bayesian Linear Discriminant Analysis is one of these methods[3]. It is a traditional

[3] Discriminant analysis contains a group of methods (linear discriminant, quadratic discriminant and logistic discriminant). An introduction refers to Mich94, Chapter 3. Fisher found another commonly used linear discriminant function, which obtains the same coefficients as in the Bayesian linear discriminant rule in case of two classes (cf. Thom00, P. 154; Mich94, P. 21).

method used to establish credit scoring models and often used as a benchmark when comparing with other model building techniques.

4.1.1 Basic Bayes' classification rule:

The following descriptions refer to Cios98, P. 131-147.

A classification rule can be stated as:

Assign case(X) to class c_j when:

$$\Pr(c_j \mid X) > \Pr(c_i \mid X), \quad i=1,2,...,l;\ i \neq j \qquad \text{(Rule 1)}$$

Using Bayes' theorem[4], the Bayes' classification rule can be obtained as:

Assign a case(X) to a class c_j when:

$$\Pr(X \mid c_j)\Pr(c_j) > \Pr(X \mid c_i)\Pr(c_i),\ i=1,2,...,l;\ i \neq j \qquad \text{(Rule 2)}$$

Suppose the cost of misclassifying a class c_i case as a class c_j case is $L_{ij.}$ The expected cost of misclassifications associated with making a decision that a case(X) belongs to a class c_j is:

$$EC_j = \sum_{i=1}^{l} L_{ij} \Pr(c_i \mid X) \qquad \text{(4.1.1-1)}$$

Classification should be based on the principle that the expected cost of misclassifications is minimized. The minimum cost Bayes' classification rule can be stated as:

Assign a case(X) to a class c_j when: $EC_j < EC_i, \quad i=1,2,...,l,\ i \neq j$ (Rule 1')

Using Bayes' theorem we have the classification rule for two classes problem ($l=2$)[5]:

Assign a case(X) to class c_1 when:

$$L_{21}\Pr(X \mid c_2)\Pr(c_2) < L_{12}\Pr(X \mid c_1)\Pr(c_1), \quad \text{otherwise to } c_2. \qquad \text{(Rule 2')}$$

Rule 2 is the classification rule for minimization of probability of the classification error; Rule 2' is the classification rule for minimizing the expected cost of misclassification.

4.1.2 Linear Discriminant Analysis

The Linear Discriminant Analysis (LDA) introduced here based on the Bayes' classification rule given in Chapter 4.1.1. The following descriptions refer to Yoba00, P. 112-113 and Desa97, P. 325-328.

Two notations we are going to use are:

---

[4] Bayes' theorem is $\Pr(c_i \mid X) = \Pr(X \mid c_i)\Pr(c_i)/\Pr(X)$.

[5] For the credit evaluation problem, in the situation of more than two classes, it is unrealistic to computing the cost L_{ij}.

1. $f(\mathbf{X}|c_i)$: the class conditional probability density function. It represents the distribution of the feature vector within each class c_i.
2. $f(\mathbf{X})$: the unconditional probability density function for the feature vector.

If we assume that $f(\mathbf{X}|c_i)$ fits a multivariate normal Gaussian distribution and with the same covariance matrix for different classes c_i , i=1,2,...,l, then $\Pr(X|c_i)$ is given by the expression:

$$\hat{\Pr}(X \mid c_i) = f(\mathbf{X} \mid c_i)|_{\mathbf{X}=X} = \frac{1}{(2\pi)^{\frac{n}{2}} |\sum|^{\frac{1}{2}}} \exp[-\frac{1}{2}(X-\mu_i)^T \sum^{-1}(X-\mu_i)], \quad i=1,2...l, \qquad (4.1.2\text{-}1)$$

where μ_i is the mean vector of the feature vector for the i_{th} class. $\sum$ is the covariance matrix of the feature vector for all classes.

We can use the equivalent form of Rule 2 by taking a natural logarithmic function for two sides of the inequation in Rule 2:

Assign a case(X) to a class c_j when: $\ln \frac{\Pr(X \mid c_j)}{\Pr(X \mid c_i)} > \ln \frac{\Pr(c_i)}{\Pr(c_j)}, \quad i=1,2,...,l, i \neq j$ (Rule 3)

The modified classification rule of Rule 2' for two classes (l=2) is as this:

Assign a case(X) to a class c_1 when: $\ln \frac{\Pr(X \mid c_1)}{\Pr(X \mid c_2)} > \ln \frac{L_{21} \Pr(c_2)}{L_{12} \Pr(c_1)}$,

to c_2 otherwise. (Rule 3')

Substituting equation (4.1.2-1) into Rule 3 and Rule 3' gives the following forms of classification rule:

Assign a case(X) to a class c_j when:

$$X^T \Sigma^{-1}(\mu_j - \mu_i) > \frac{1}{2}(\mu_j + \mu_i)^T \Sigma^{-1}(\mu_j - \mu_i) + \ln \frac{\Pr(ci)}{\Pr(cj)}, i=1,2,...l, i \neq j \qquad \text{(Rule 4)}$$

Assign a case(X) to a class c_1 when:

$$X^T \Sigma^{-1}(\mu_1 - \mu_2) > \frac{1}{2}(\mu_1 + \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2) + \ln \frac{L_{21} * \Pr(c_2)}{L_{12} * \Pr(c_1)} ,$$

to c_2 otherwise. (Rule 4')

Rule 4 is the final linear discriminant rule for more than two classes without cost factors over different misclassifications. Rule 4' is the final linear discriminant rule for two classes with cost factors over different misclassifications.

The unknown μ_i and $\sum$ in Rules can be estimated by the sample mean and sample covariance matrix. The estimation of the priori probabilities ratio ($\Pr(c_2)/ \Pr(c_1)$) and the computation of the misclassification cost ratio (L_{21}/L_{12}) have major effect in practice. The former has been estimated as sample proportions N_{c2}/N_{c1}, which may be biased for the

overall population in the case that the sample is not a random sample (cf. RoGl94, P. 592). The latter is more difficult to decide in practice.

Since the rules are linear in X (the technique is therefore called Linear Discriminant Analysis), Figure 4.1.2/1 illustrates LDA with two classes and two-dimensional variable space. A line (a hyperplane in the k-dimensional variable space) is given by the linear discriminant function to separate 'good' and 'bad' cases. We can change the cutoff score, i.e. move the line, to divide two classes.

In fact, the determination of cutoff score can substitute the determination of $\Pr(c_2)/\Pr(c_1)$ and L_{21}/L_{12}. The cutoff score represents the position of the line, which is determined by the constant of the discriminant linear function. $\Pr(c_2)/\Pr(c_1)$ and L_{21}/L_{12} are part of this constant. That means when a cutoff score is decided there is no need to compute $\Pr(c_2)/\Pr(c_1)$ and L_{21}/L_{12}.

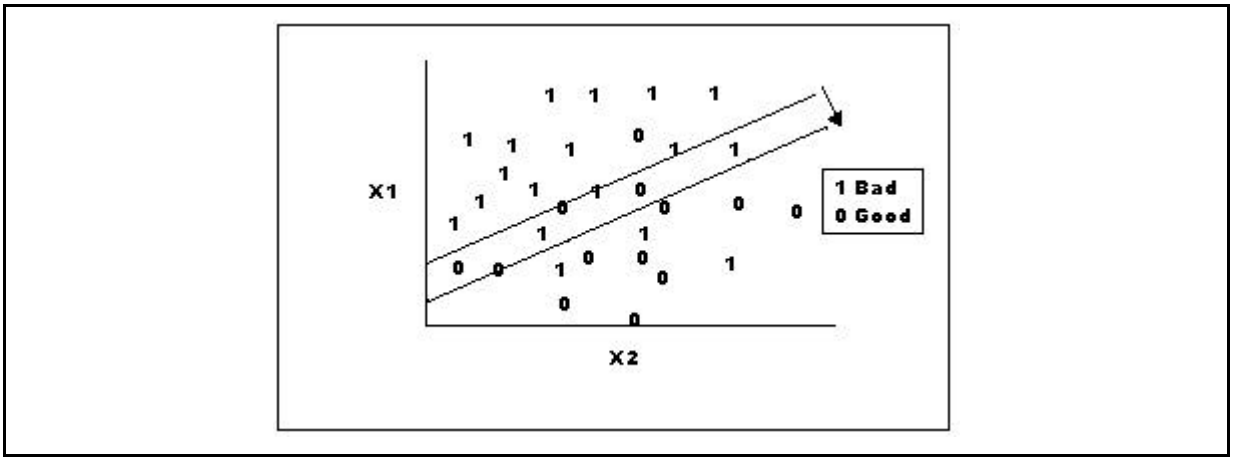


Figure 4.1.2/1: Linear Discriminant Analysis (cf. RoGl94, P. 592)

4.2 Logistic Regression

Logistic regression has been researched to apply to credit scoring problems. Some research showed that Logistic regression demonstrates better performance than Linear discriminant analysis in credit scoring (cf. Kron98, P. 30). In addition, Logistic regression allows the interpretation of the output variable as a probabilistic binary value. That is, if we predict whether a new customer will be a defaulter, the prediction does not simply come out as a primitive yes/no answer, but as a more refined estimated probability that this event will take place. This prediction output is continuous, therefore, for classification, a relevant threshold value (cutoff point) must be determined.

Logistic regression is a variation of ordinary regression, useful when the observed outcome is restricted to two values, which usually represent the occurrence or non-occurrence of

some outcome event, (usually coded as 1 or 0, respectively). It produces a formula that predicts the probability of the occurrence as a function of the independent variables.

If the credit customers can be categorized into two groups: default (c_1=1) or no default (c_2=0). Logistic regression can predict $\Pr(c_1|X)$, which represents the probability that a case X will become default, where $X = (x_1, x_2, ..., x_k)$ is assumed to be the characteristics of a customer. $\Pr(c_1|X)$ is succinctly substituted with P in the following descriptions.

Just like linear regression, logistic regression gives each x_i a coefficient w_i which measures the contribution of each x_i to variations in P. But what we want to predict from x_i is not an ordinary numerical value, but rather the probability (P). This probability cannot be used as the dependent variable in an ordinary regression, e.g. a simple linear regression, because x_i may be unlimited in range. If we expressed P as a linear function of them, we might then find ourselves predicting that P is greater than 1 (which cannot be true, as probabilities can only take values between 0 and 1).

We get over this problem by making a logistic transformation of P, also called taking the logit of P. Logit (P) is the log (to base *e*) of the odds or likelihood ratio. It is defined as:

$$\text{logit}(P) = \log(P/1-P), \tag{4.2-1}$$

where P can only range from 0 to 1, logit (P) ranges from $-\infty$ to ∞.

This leads to the Logistic regression approach where one matches the log of the probability odds by a linear combination of the characteristic variables, i. e.

$$\text{logit}(P) = \log(P/1-P) = w_0 + w_1x_1 + w_2x_2 + w_3x_3 + ... + w_kx_k \tag{4.2-2}$$

The probability P is then computed as:

$$P = \frac{\exp(w_0 + \sum_{i=1}^{k} w_i x_i)}{1 + \exp(w_0 + \sum_{i=1}^{k} w_i x_i)}. \tag{4.2-3}$$

Parameters w_i are estimated by maximum likelihood method. Maximum Likelihood estimation was historically a difficulty since it requires non-linear optimizing techniques using iterative procedures and is computationally more intensive than linear regression, but with computing power available now this is not a problem (cf. Thom00, P. 156).

4.3 Instance-based learning

In the instance-based learning algorithms, the training instances are memorized and when a decision need to be made on a new case, a searching is taken among the saved instances to find the pattern that resembles most strongly to the new case (cf. WiFr00, P. 72). The

"learning" takes place at the time of application, therefore it is also called lazy learning or memory-based learning.

The instance-based learning method has several attractive properties which make it suitable for the credit scoring problem. The non-parametric nature of the method enables modeling of irregularities in the risk function over the feature space. It is a fairly intuitive procedure and as such could be easily explained to business managers who would need to approve its implementation, i.e. it is conceptually simple and straightforward to implement. It can be used dynamically and incrementally, with automatic updating of the model as the cases evolve (cf. Heha97, P. 306).

4.3.1 k-Nearest-Neighbors (k-NN) [6]

The following descriptions refer to HeHa97, P. 306.

Let the probability that a new case with feature vector X will belong to the class c_i be denoted by $\Pr(c_i|X)$. Then a k-NN estimate of $\Pr(c_i|X)$ is given by k_{ci}/k, where k_{ci} is the number of cases from class c_i among the k most similar examples to X, chosen from a 'training set' of cases with known classes. A distance function is used to calculate the similarity between each training case and the new case. k cases are determined with the k most short distances, in which k is a fixed, small number. The standard classification rule is as follows:

Assign a case(X) to a class c_i when:

$$k_{ci} = \max\{k_{c1}, k_{c2} ..., k_{cl}\} \qquad \text{(Rule 5)}$$

The class of the new case is predicted as the class of the majority of these k cases. Figure 4.3.1/1 illustrates the k-NN method with two features.

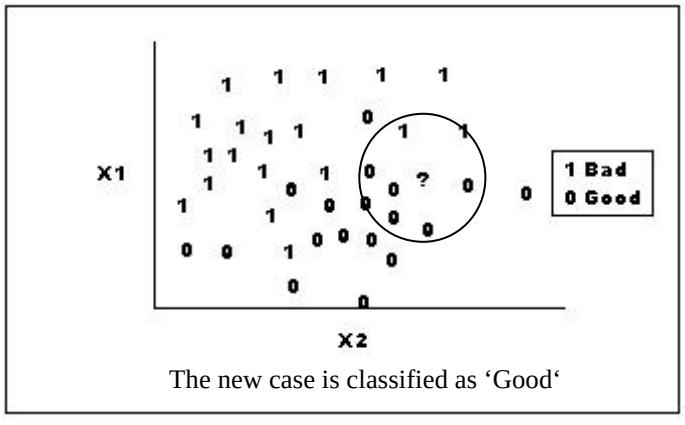


Figure 4.3.1/1: k-Nearest-Neighbours method

[6] The statistical basic theory of k-NN refers to Cios98, P. 173.

There are several alternatives in the defining of the distance functions. Here some simple ones are given. Distance between a case with attribute values $x_1^{(1)}$, $x_2^{(1)}$,..., $x_m^{(1)}$ (where m is the number of features) and one with values $x_1^{(2)}$, $x_2^{(2)}$,..., $x_m^{(2)}$ is defined as:

$\sqrt{(x_1^{(1)} - x_1^{(2)})^2 + (x_2^{(1)} - x_2^{(2)})^2 + ... + (x_m^{(1)} - x_m^{(2)})^2}$. (Euclidean distance)

or $\sqrt{|x_1^{(1)} - x_1^{(2)}| + |x_2^{(1)} - x_2^{(2)}| + ... + |x_m^{(1)} - x_m^{(2)}|}$. (Manhattan distance)

4.3.2 Locally weighted regression (LWR)

Locally weighted regression is an instance-based method for numeric prediction. Since the results of LWR are continuous numeric values, it gives not only the classification of cases but also the scores which can be considered as the probability of belonging to a class. In the application of credit scoring models, the continuous scores are more useful than a binary good/bad classification. Following is the brief description of the LWR algorithm.

For a new test case, LWR method generates local linear regression models by weighting the instances in the neighborhood of the test case. More specifically, it weighted the training cases according to their distance to the test case and build a linear regression model on the weighted data.

There are some choices of the weighting of training cases. A commonly used way is to weight the cases according to the inverse of their Euclidean distance from the test case. Another possibility is to use the Euclidean distance in conjunction with a Gaussian kernel function. Thus, the training cases nearer to the test case have a higher weight, and those far away have a lower one.

An important parameter is the "smoothing parameter" --- the distance is multiplied with the inverse of this parameter. If this parameter is small, only cases very close to the test case receive high weight; if this parameter is set to be large, more cases will also have significant impact on the linear regression model. Using a suitable parameter the noisy in the data set can be smoothed. The smoothing parameter can be set to be the distance of the k_{th} nearest case. In this way, if the inverse of distance is used as the weights, the k nearest cases have weights larger than 1, and the other cases have weight smaller than 1.

4.4 Model Trees: M5

Decision Tree methods have been developed both in statistics (cf. Brei84) and machine learning (cf. Quin93) areas. The decision trees that used for numeric prediction can produce continuous scores, which are more useful for credit scoring, since the models can give a continuous default risk. M5 is an algorithm that can produce model trees for numeric predction, which is first described by Quinlan (cf. Quin92, P. 343). Although model trees

produce continuous numeric values, they can be applied to classification problems by treating the continuous values as the approximated probability of belonging to a class (cf. Fran97, P. 63). Following is the descriptions of the model trees algorithm M5, The descriptions refer to WiFr00, P. 201-208.

M5 algorithm consists of two steps: building the tree and pruning the tree.

First a basic tree is built as follows:

The training sample is recursively partitioned into subsets, until the leaf nodes contain only homogeneous cases or some other reasonable stopping criterion appears. For a classification tree, the homogeneity means the cases are from a single class, while for numeric prediction tree, the homogeneity of cases is measure with the intrasubset variation in the numeric values of the cases within the subset.

The partitions are typically made on a single variable in such a way: A particular variable (split variable) and a particular value of the variable (critical value) is chosen to split the cases into two subsets. In order to search the split variable and the critical value, a measure which is called – standard deviation reduction (SDR) — is employed to indicate the change of variation in the subsets before and after a partition.

Standard deviation reduction (SDR) is calculated by:

$$\mathrm{SDR} = \mathrm{sd}(\mathrm{T}) - \sum_{\mathrm{i}=1}^{2} \frac{|\mathrm{T_i}|}{|\mathrm{T}|} * \mathrm{sd}(\mathrm{T_i}), \tag{4.4-1}$$

where T_1, T_2 are the subsets that result from splitting the set T. sd(*) stands for the standard deviation of the numeric values in the sets.

The split variable and critical value with the maximal SDR are chosen. The splitting process continues until the standard deviation of the subset is very small (less than 5% of the standard deviation of the original set) or just a few cases (4 or fewer) remain in the subset.

After the basic tree has been constructed, the pruning of the tree gives each node of the tree a regression model. The process is as follows:

For each node of the unpruned tree a linear regression model is built using only the variables that are used in the subtree of this node. The parameters of the linear regression functions are calculated using standard regression.

An estimate of the expected error is calculated for each node in the tree with:

$$\mathrm{E_{Error}} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} |\,\text{difference between predicted value and actual value}\,|}{\mathrm{n}} * \frac{\mathrm{n}+\mathrm{v}}{\mathrm{n}-\mathrm{v}} \tag{4.4-2}$$

Using the linear model for prediction, the absolute difference between the predicted value and the actual value is averaged over each of the training cases that reaches that node. The

averaged error is multiplied by the factor $(n+v)/(n-v)$. Here n is the number of training cases that reach that node, v is the number of parameters in the linear model at that node. The final linear model is determined by minimizing the E_{error}. Due to the factor $(n+v)/(n-v)$, the linear model may be simplified because dropping a term decreases the factor (the second part of the error), the decrease may be larger than the inevitable increase in the averaged error (the first part of the error). Terms in the linear function are dropped one by one so long as the E_{error} decreases.

Finally, once the simplified linear models is placed for each interior node. The tree is pruned back from the leaves. If the E_{error} of a node is smaller than the E_{error} of the subtree below, the subtree is replaced with this single node. The E_{error} of a subtree is the linear combination of the E_{error} of each of the nodes in the subtree by weighting each node with the proportion of the cases that reach it.

The algorithm generates a conventional decision tree with linear regression models at each of its leaves. When a model tree is used to predict the value for a test case. the decision is made by following the tree down to a leaf using the case's variable value. The linear model that based on some variables is applied to the case to yield a predicted value.

To compensate for the sharp discontinuities between adjacent linear models at the leaves, a smoothing process is accomplished with this calculation through making use of the linear models at each interior node:

$$p' = \frac{np + kq}{n + k} \tag{4.4-3}$$

where p' is the prediction passed up to the next higher node, p is the prediction passed to this node from below, q is the prediction by the model at this node, n is the number of training cases that reach the node below, and k is a smoothing constant. In this way, the predicted value at a leave is smoothed by combining it with the predicted values along the path back to the root.

4.5 Neural Networks: Multi-layer perceptron

There are many neural network techniques, multi-layer perceptron with back-propagation (MLP-BP) is one of the choices for classification decisions. MLP-BP is the most popular overall, due in part, to its being extremely robust in its modeling capability (cf. KaCo97, P. 469).

The multi-layered perceptron neural network consists of input, hidden, and output layers of interconnected neurons. Neurons in the one layer are combined according to a set of weights and fed to the next layer. During the training phase, the data is fed to the neural network one

by one, and the weights of the neurons are modified based on the error rates of the resulting outputs. Multiple passes are required over the data in order to train the network. As a result, the training times are quite large. Discussions below refer to Mich94, P. 87-91 and Kenn98, P. 10-25.

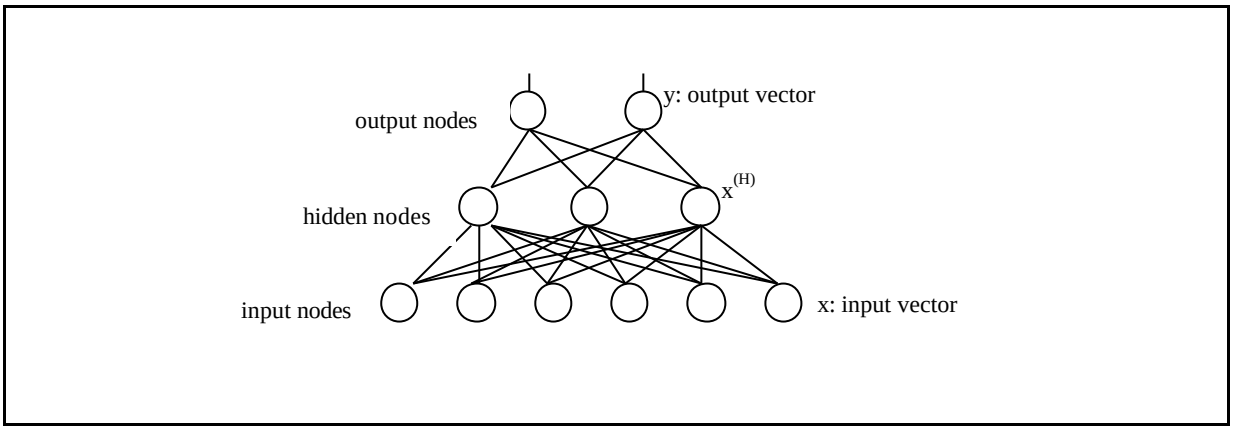


Figure 4.5/1: The structure of a standard one-hidden-layer perceptron

Figure 4.5/1 shows the structure of a standard one-hidden-layer perceptron. The inputs form the input nodes of the network; the outputs are taken from output nodes. The middle layer of nodes, visible to neither the inputs nor the outputs, is termed the hidden layer, and unlike the input and output layers, its size is not fixed.

The operation of this network is specified by:

$$x_i^{(H)} = f^{(H)}\left(\sum_j w_{ij}^{(HI)}\, x_j\right), x_i^{(H)} \text{ is the } i_{th} \text{ hidden node}, x_j \text{ is the } j_{th} \text{ input node.} \quad (4.5\text{-}1)$$

$$y_i = f^{(O)}\left(\sum_j w_{ij}^{(OH)}\, x_j^{(H)}\right), y_i \text{ is the } i_{th} \text{ output node.} \quad (4.5\text{-}2)$$

This specifies how input pattern vector $\mathbf{x}$ is mapped into output pattern vector $\mathbf{y}$, via the hidden pattern vector $\mathbf{x}^{(H)}$, in a manner parameterized by the two layers of weights $\mathbf{w}^{(HI)}$ and $\mathbf{w}^{(OH)}$. The univariate functions $f^{(*)}$ are typically each set to be:

$$f(x) = \frac{1}{1+e^{-x}} \quad , \quad (4.5\text{-}3)$$

which varies smoothly from 0 to 1 (when x varies from $-\infty$ to ∞).

Back propagation is an algorithm for training a multi-layer perceptron, i. e. for modifying the weights of a MLP network.

Suppose a training data set with k input variables and l classes consists of a set of n cases. The i_{th} case has input $X_i = (x_{i1}, x_{i2}, ..., x_{ik})$ and output $Y_i = (Y_{i1}, Y_{i2}, ..., Y_{il})$, if the case belongs to the j_{th} class, the j_{th} of $(Y_{i1}, Y_{i2}, ..., Y_{il})$ is equal to1, others are equal to 0.

Back propagation algorithm adjusts the weight matrix parameters $\mathbf{w}$ in order to minimize an error measure:

$$E = \frac{1}{2}\sum_i \sum_j (y_{ij} - Y_{ij})^2 , \qquad (4.5\text{-}4)$$

where Y_{ij} is the actual value of j_{th} output as given by i_{th} case. y_{ij} is the output value from the model calculated by (4.5-1) and (4.5-2).

The minimization is accomplished by means of gradient decent. The basic strategy in gradient descent is to compute the gradient and adjust the weights in the opposite direction. The gradient $\nabla E(\mathbf{w})$ of $E(\mathbf{w})$ is the vector field of derivatives of E:

$$\nabla E(\mathbf{w}) = (\frac{dE(\mathbf{w})}{dw_1}, \frac{dE(\mathbf{w})}{dw_2}, ...) \qquad (4.5\text{-}5)$$

The vector $\nabla E(\mathbf{w})$ points in the direction of fastest increase of E. Consequently an adjustment of $\mathbf{w}$ in the direction of $-\nabla E(\mathbf{w})$ provides the maximum possible decrease in E. The weights are repeatedly adjusted by:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla E(\mathbf{w}) + \alpha \Delta \mathbf{w} \qquad (4.5\text{-}6)$$

where $\Delta \mathbf{w}$ is the most recent weight change. Here two parameters are added: η is called 'learn rate', α is called 'momentum', which are used to adjust the step size and speed of the change of $\mathbf{w}$.

The training process ends until the error E does not descend or some other stopping condition appears.

4.6 Comparisons of the introduced algorithms

Input requirements:

Linear discriminant analysis (LDA), Logistic regression (LR) and neural networks (NN) require that input variables are numerically-valued. Categorical variables must be transformed into numeric variables.

The applications of LDA have some more requirements (cf. Leke93, P. 176). One main requirement of LDA, also one main criticism of LDA, is the assumption that variables are multivariate normally distributed and the covariance matrices are equal for the groups. Some statistic methods were used by model builders to test these assumptions (cf. Feid92, P. 74; P. 173). Some other model builders tried to transform input variables so that their marginal

density is approximately normal, usually by applying a monotonic transformation of the power law type. For significant portion of credit information the normal assumption cannot be true, for example, many categorical variables in credit information are certainly not normally distributed. The linear discriminant rule may be not optimal if the assumptions are not satisfied. However, if LDA is regarded as yielding the linear combination of the variables which maximizes a particular separability criterion, then clearly it is widely applicable. Some empirical observation of credit scoring also showed that the fact of the violation of normal assumption in LDA did not prevent its successful application (cf. HaHe97, P. 532; Desa97, P. 327).

The classification functions produced by LDA and LR are linear functions of input variables. The multicollinearity among the input variables has negative effects on the results (cf. Feid92, P. 111; Nieh87, P. 109). Multicollinearity means one variable is the linear combination of other variables. One should pre-examine the input variables and ensure there is no strongly correlated input variables left in the data set.

Other disadvantage of LDA and LR is that they cannot handle interactions. Interaction occurs when the correlation between a variable and the dependent variable depends on the value of other variables. To solve this problem, interaction variable, which is the product of two or more variables can be included in the model (cf. RoGl94, P. 601).

One requirement for NN is the normalization of input variables. The input variable may have different scales of values, their influences on the output would be unequally considered, which leads to the biased models. Therefore, values of each input variable need to be transformed into the same scale (e.g. between 0 and 1) (cf. Krau93, P. 144-145).

Claims by practitioners imply that the performance of neural networks, DT (decision trees), and instance-based learning methods (IBL) in comparison to the conventional methods would depend upon the proportion of bad loans in the data set (cf. Desa97, P. 339; Mich94, P. 133). Usually, same prior probabilities of 'bad' and 'good' cases in the train sample can improve model performances. In contrast, statistical algorithms can cope, to a greater or less extent, with the different class proportions in the train sample (cf. Mich94, P. 133).

From the comparison of input requirement of the five algorithms (cf. Table 4.6/1) it can be seen that the restricted requirements by traditional methods (LDA and LR) limit their applicability and additional preprocessing tasks should be undertaken. In the contrary, modern methods (NN, IBL, and DT) can learn knowledge from the data with little assumptions or requirements of the input data.

Model outputs:

The outputs of LDA and LR are in the form of scores. These two techniques have been most widely used for building scorecards. Typically the coefficients and the numerical values of the

attributes are combined to give single contributions which are added to give an overall score (cf. HaHe97, P. 531).

The outputs of NN can be either continuous scores or class memberships. One differentiates them as classification networks and scoring networks (cf. FrHo98, P. 188).

The outputs of DT are usually the class memberships. The tree methods with class outputs classify the consumers into groups, each group has similar default risk and as different from the default risks of other groups as is possible (cf. Thom00, P. 158). However, there are also tree methods that can give numeric predictions. For example, the tree method M5 gives continuous predictions with linear regression functions at the leaves (cf. Fran98).
The output of instance-based learning methods can be scores or classes. k-nearest-neighbors method produces class memberships, while the numeric prediction instance-based method locally weighted regression outputs continuous scores.

For the methods with scores output, in case of two classes a cutoff value can be determined to choose the desired tradeoff between good and bad risks. This increases flexibility of decision making. When the risk concept of the credit grantors changes, what they need to do is only changing the cutoff score. In contrary, the methods that output definite class memberships lack this flexibility although they are straightforward. When risk definition changes, they should rebuild their models.

		Classification methods				
		LDA	LR	NN	DT	IBL
Input requirements	numeric variables	X	X	X		
	normally distributed variables, equal covariance matrices	X				
	problem of interaction	X	X			
	problem of multicollinearity	X	X			
	normalization of variables			X		
	sensitive to class proportions			X	X	X
Output form	score	X	X	X	X	X
	class			X	X	X
Performance	classification accuracy	no unified results in the past research				

Table 4.6/1: Comparisons of five classification methods

Performances in past applications

The fact that new techniques like NN can discover non-linearity which is not possible in the traditional techniques should correspond to a superior performance. However, the small number of publicly available comparisons between these techniques in the credit scoring context does not clearly support this expectation of new techniques' performance. The published literature suggests mixed results: some find that traditional techniques perform better than new techniques, others the reverse (cf. Yoba00, P. 111). Some author even concluded that there is only small difference between classification accuracy of each method (cf. Thom00, P. 160; HaHe97, P. 536).

For example, one research found that NN might outperform somewhat better than LDA for classifying the poor performance customers of a consumer credit product (cf. Desa97, P. 344; Desa96, P. 36). However, another research found that 'bad' cases can be identified better with discriminant analysis, while 'good' cases can be classified better with NN (cf. Erxl91, P. 23).

When a new algorithm is presented, the author always proves its superiority with experiments on some actual data sets. However, whether the superiority can be proved in other data sets is not sure. The comparison research showed that any method can not perform always better than others. For a particular data set, there exists a particular optimal algorithm.

The performance of a classification algorithm depends on how the model builder treat the data set, such as selecting of input variables, treating missing values, etc. and how the model builder choose the model parameters. To select an optimal method for a particular data set, these factors should be considered.

5 A framework of the data mining application process for credit scoring

5.1 Reasons for a process framework

Credit scoring is a classification problem of data mining. Like other data mining applications the process of credit scoring tends to be iterative and computing extensive that cannot yet guarantee an optimal model solution. Credit scoring has been always based on a pragmatic approach: when a model works then use it. The best and standard solution for every circumstance does not exist.

Providing a general framework for data analysis process of credit scoring, which includes necessary pre- and post-processing of the real-world large data sets in order to support a standard model building process will be of great utility. The framework contributes itself to the academic research as well as the practical application.

Many new data mining techniques were continuously presented and improved by statisticians and computer scientists. Their practical uses in business applications need to be validated. The framework for credit scoring serves as the tool to validate the effect of new data mining techniques in practical applications. Researchers can validate their new model algorithms in a systematic way by adopting the standard scoring model building process.

The general framework presents a reference to the practical credit scoring model building. Not only many of the techniques under the general framework can be applied to model building, but the process and the process controlling strategies in the general framework can support the model building in a consistent and systematic way.

It should be pointed out that whether the models built with this framework are the best solutions is hard to judge. Famous credit rating firms like Experian or Fair-Isaac have invested huge resources in the development of models for credit scoring. These models are propriety, closely-held secrets and are typically specialized to the problem at hand (cf. FoSt01, P. 36). It is impossible to apply their methodology to the data analyzed in this research and compare their performance. It is also not the aim of this research. The goal of the framework is not to reach the only best solution, which is impossible for various real world credit decision problems; but rather to identify the process of building credit scoring models and techniques that can help to get solutions approaching to the optimum.

Although the framework provided here is trying to be generalized in order to represent the most circumstances of credit scoring applications, the absolute standard solution still does not exist. The real world problem of credit scoring cannot be expected to be controlled completely like an experiment in laboratory. There are so many decisions that must be made

to tackle expected and unexpected variations in practice. This framework should be understood as only a template that can facilitate the control of the complex real world data mining process.

5.2 The presentation of the general framework

The overall process of data mining includes three stages (cf. chapter 2.2, Figure 2.2/1): 1). problem definition and data preparation, 2). data analysis and model building, and 3). models application and validation. As mentioned in chapter 2.2, the first stage and the last stage are specifically related with application domains. In the following sections the practical problems in the first and the third stages are outlined, and a detailed framework for the second stage is presented.

5.2.1 The stage of the problem definition and data preparation

In this stage following tasks should be finished.

Task 1. Define the business problem.

Problem definition decides the business objectives of scoring models and the practical uses of the built models.

- Business Objectives: Scoring models can be built for various objectives, for example, the scoring model for screening credit applicants, the scoring model to evaluate existing credit takers (cf. Liuy01, chapter 3.2).
- Practical uses: The built models can be used in different ways, for example, the scoring model designed for automatic decision making, the scoring model used as an assistant tool to support the decision making by experts (cf. chapter 5.2.3).

The objectives and the uses of scoring models drive the entire data mining process. They are the basis on which the data mining project is established and decide the criterions by which the final model will be judged.

Task 2. Collect relevant data

The data that are relevant to the problem should be collected from the available sources (cf. Liuy01, chapter 4.1). Scoring models are based on past cases, a time period should be determined for the selecting of relevant data (cf. Liuy01, chapter 4.2).

Task 3. Merge and clean the data.

The data from different resources should be transformed, cleaned and merged. This step may be not easy and take much time in practice, since the data are usually not in the same format, some data are even not in the electronic format. A well established

business database, especially a data warehouse, would facilitate the tasks of data collection and preparation.

Task 4. Define the classes.

The final task is the determination of the class memberships of the cases (cf. Liuy01, Chapter 4.4). The collected cases may have definite classes, for example, the bankrupt and non-bankrupt firms. In other situations, the cases have uncertain classes, for example, it should be decided whether the bad customers are defined as those that are behind in payment for 90 days or 30 days. Sometimes, there are not available past cases with known payment behaviors. The collected cases are analyzed by experts manually and divided into different risk classes.

The tasks in this stage consist of several steps which are illustrated in Figure 5.2.1/1. These steps have not a definite order in practical process. For example, the classes may be firstly defined, after then the data are collected. In addition, iterations may occur in every step.

The result of first stage is a data set with the standard format: one dependent variable represents the class of cases and multiple independent variables represent the features of the cases.

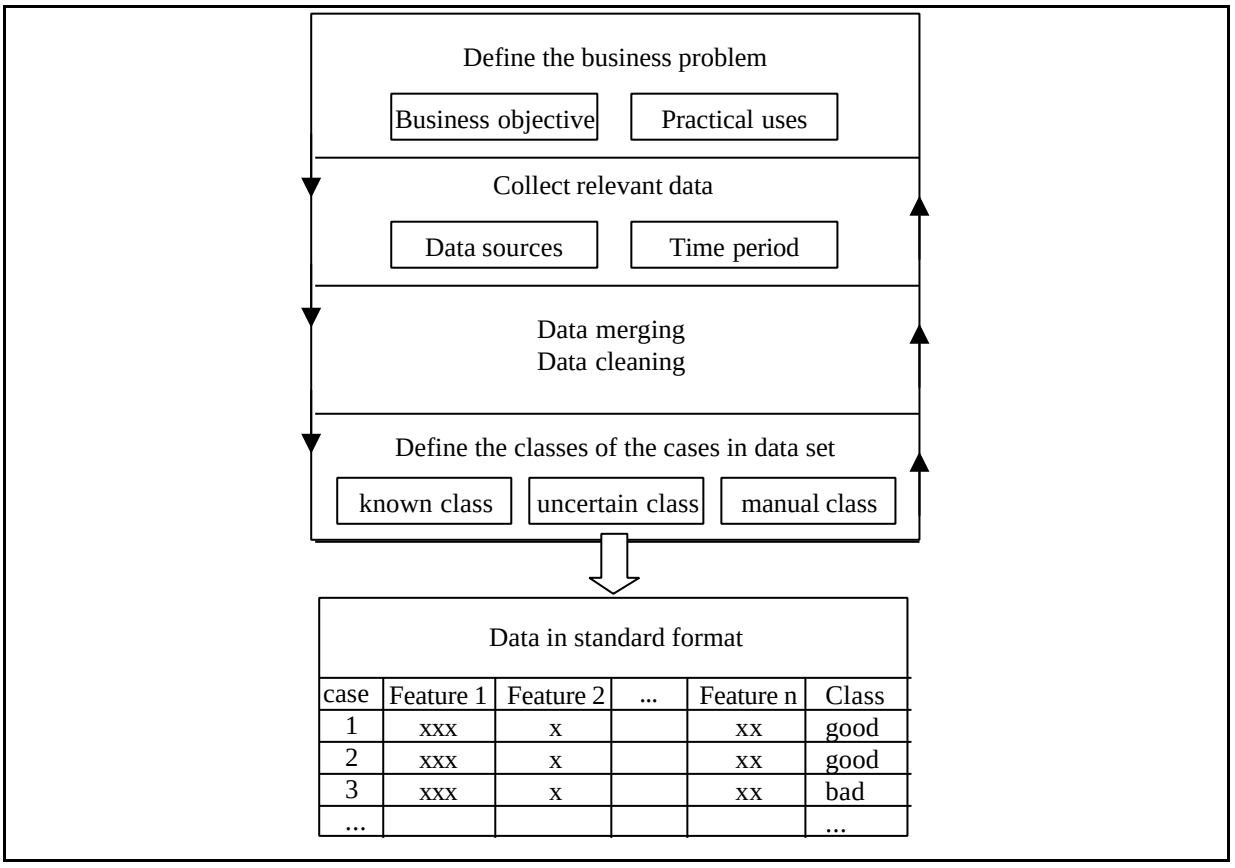


Figure 5.2.1/1: The first stage: problem definition and data preparation

5.2.2 The stage of the data analysis and model building

Having the available data set with the standard format we start the second stage --- the data analysis and model building stage. The objective of this stage is to build scoring model from the available data. Various data mining techniques can be applied to this stage. In order to validate the effectiveness of some new data mining techniques, a framework for the data analysis and model building stage is created.

The overall framework is illustrated in Figure 5.2.2/1. The framework contains four parts: the "input-relevant subprocess" preprocesses the input data; the preprocessed data are input into the "core algorithms", which finish the basic model building; the basic models are combined in the "model-relevant subprocess" to get improved models; all the models, generated either from a single model algorithm or from a model combination technique, are evaluated in the "evaluation subprocess".

One of the reasons of this framework is to evaluate some data mining core algorithms and peripheral techniques. Therefore, various techniques are included in the framework. The practical scoring model building process may not use all the techniques in this framework.

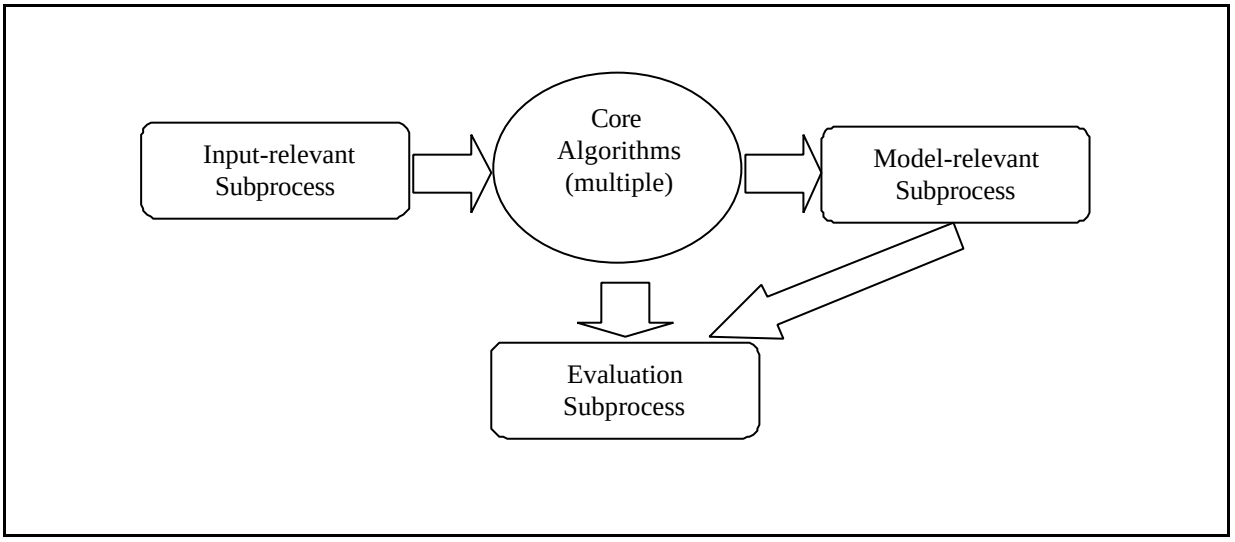


Figure 5.2.2/1: The general framework for data analysis and model building

In order to explain how subprocesses are related with each other and how they work, the detailed diagrams of each subprocesse are given below. The legends used in the diagrams are shown in Figure 5.2.2/2.

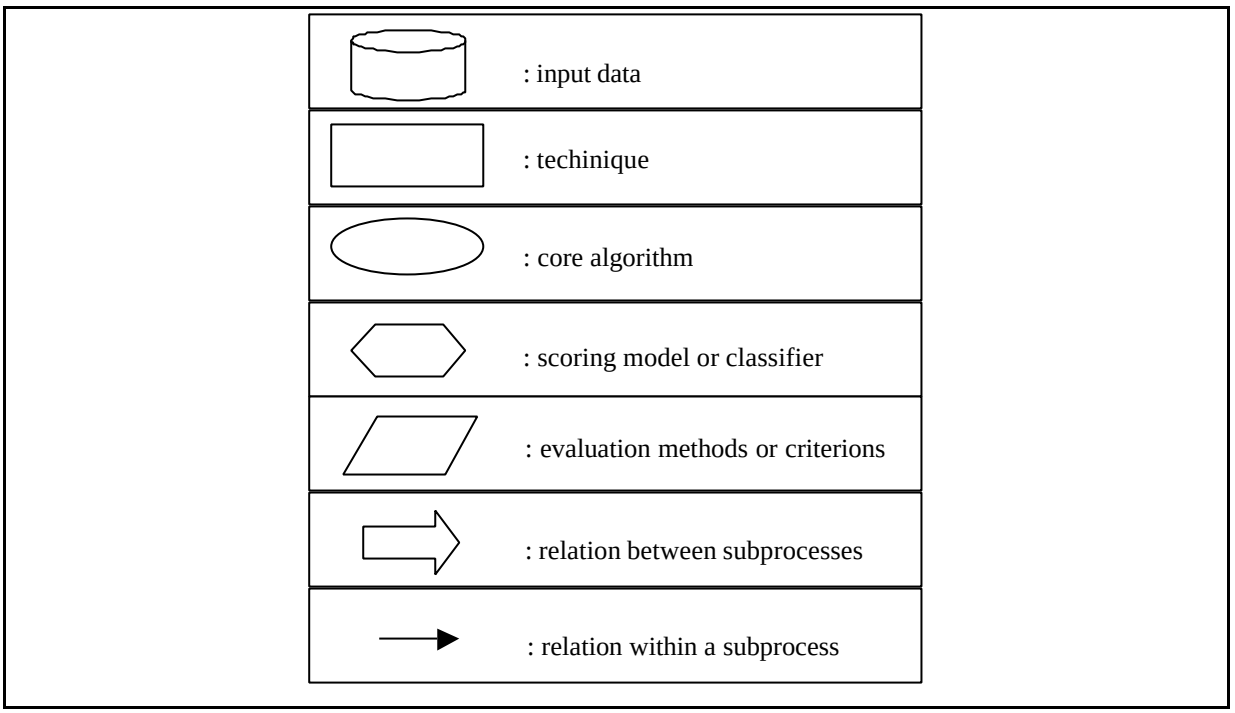


Figure 5.2.2/2: The legends in diagrams

- Input-relevant subprocess: In this part some preprocessing techniques help to improve the quality of input data (see Figure 5.2.2/3). For example, since some model algorithms might require input data with certain measurement level, it is sometimes necessary to change the measurement level of some variables. The techniques for changing continuous variables into categorical are called discretization techniques. Another example, relevant features in the original data set need to be selected. Some feature selection techniques are available but still need to be validated in practice. Other preprocessing might also be necessary and can be added to this framework.

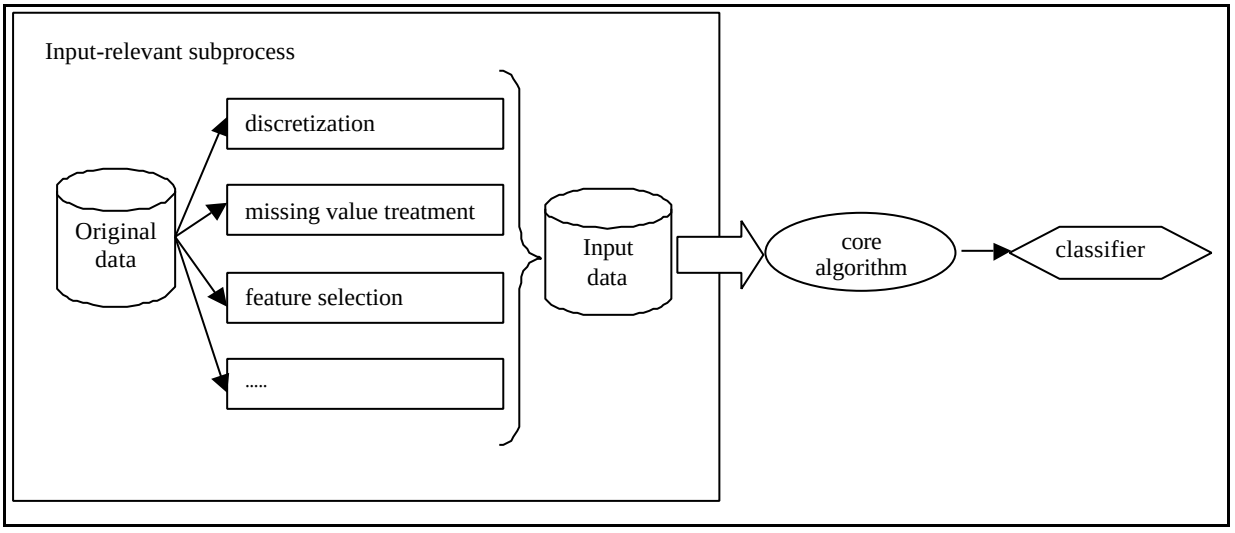


Figure 5.2.2/3: The input-relevant subprocess

- Model-relevant subprocess: In this part models are combined to improve the performance of individual model (see Figure 5.2.2/4). Some model combination techniques are like

bagging, boosting and stacking. There are two steps in model combination techniques: 1). generate component models by repeatedly building different classifiers using one base learning algorithm or by building multiple classifiers using different learning algorithms, 2). combine prediction results of component models through the methods like voting or training a 2nd-level model to estimate the weights of component models.

- Evaluation subprocess: Models are compared and evaluated in this subprocess (see Figure 5.2.2/5). The criterions and methods for evaluating credit scoring models are used in this subprocess. Criterions of evaluation include the classification accuracy, the speed and the interpretability of models, etc. Methods of evaluation include confusion matrix, ROC curve, cost function and learning curve, etc. The final model or models can be chosen according to these evaluation criterions and methods.

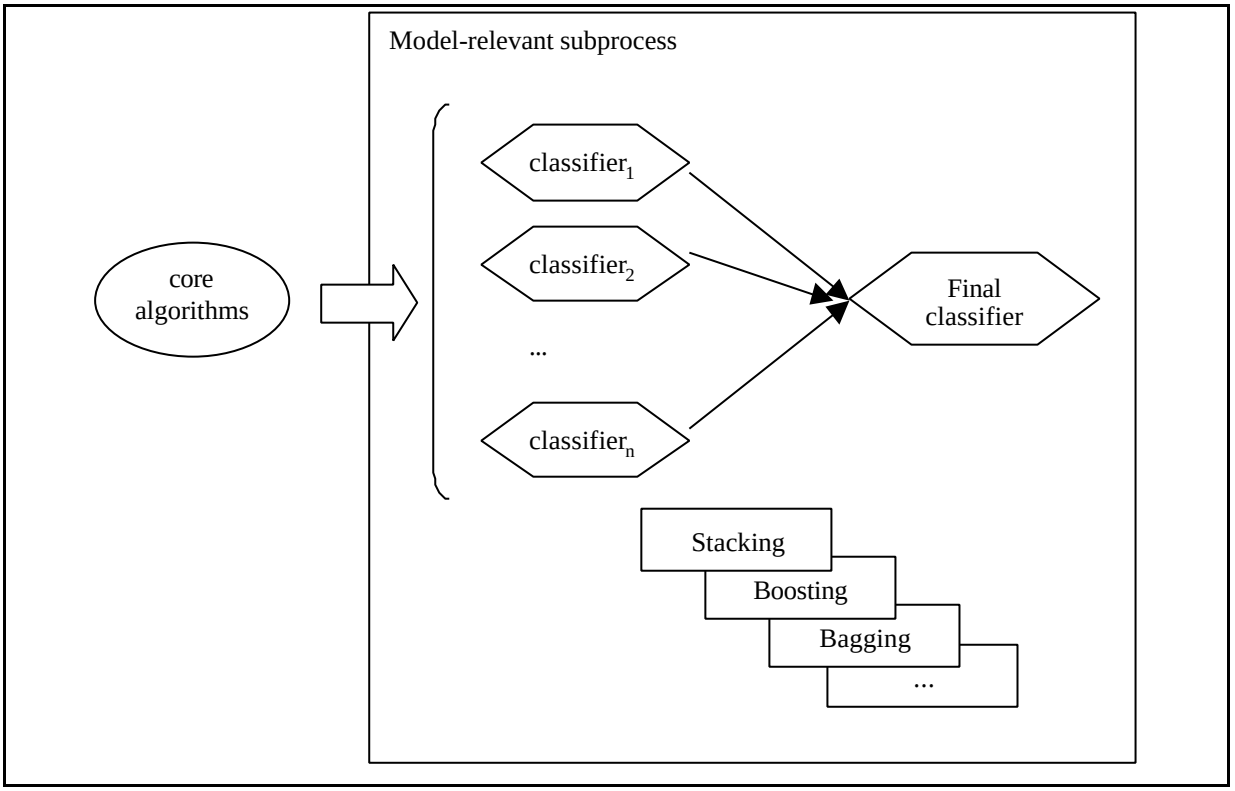


Figure 5.2.2/4: The model-relevant subprocess

The aim of the evaluation is either to select the proper, if possible, optimal model among a number of models or to compare and validate the effectiveness of some input-relevant techniques or model combination techniques. The models to be evaluated and compared can be created by using different data mining techniques in any step of the process: the classifiers generated either from the core algorithms (input data may be preprocessed through the input-relevant subprocess) in Figure 5.2.2/3 or from the model-relevant subprocess in Figure 5.2.2/4 can be the objectives to be evaluated. For example, in input-relevant subprocess, different feature selection techniques may result different input

data sets, through the evaluation of the models generated with these input data sets, the most suitable feature selection technique can be found.

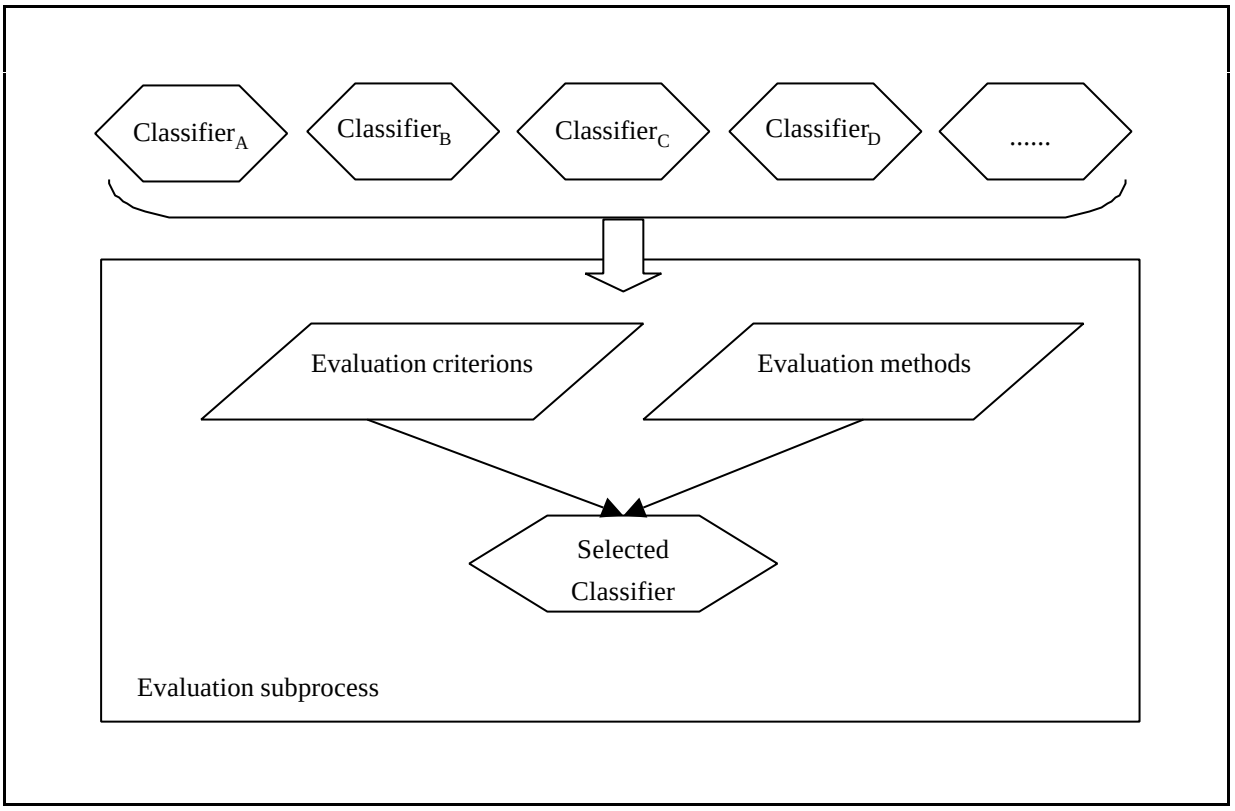


Figure 5.2.2/5: The evaluation subprocess

5.2.3 The stage of the model application and validation

In the third stage, the built scoring models should be applied and validated in the practical credit decisions. The uses of scoring models are different in practice. The possible uses of scoring models can be generalized as followings:

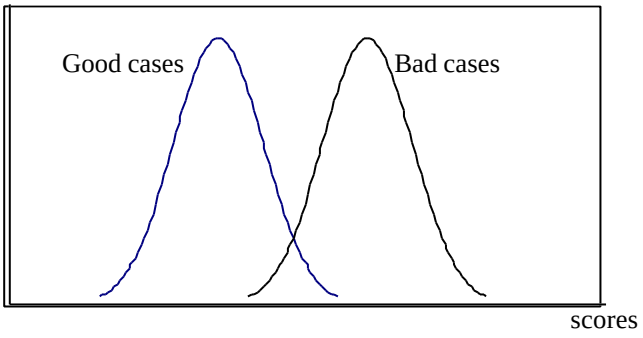
- Scoring models are used as the only analysis method for an automatic credit decision. This is applied usually to the credit products granted in larger numbers and with small amount of money, such as credit card. Human interventions on credit decisions are only necessary with individual difficult and important cases.
- Scoring models are used as one of the analysis tools and integrated in a credit decision process or in a credit decision system. The credit decision system may consist of scoring models and other analysis tools as well as experienced experts. A simple example is a two-step decision making process of credit granting (cf. Baet98, P.16). The credit applicants is pre-evaluated firstly by scoring models, the cases classified as 'good' are accepted, the cases classified as 'bad' are reexamined individually by experienced credit analysts.

- The results of scoring models are used as inputs of a rating system or a portfolio model. Sometimes, the behavior scoring models are not used only to make direct credit decisions, but also used as ingredients of other systems or models for the credit portfolio management. For example, the results of scoring models are used to establish the ratings for credit risk. Companies are classified into several risk levels according to their scores, perhaps adjusted by experts according to other factors when necessary (cf. Baet94, P.2). Sometimes, a portfolio model aggregates the default probability of each individual risk predicted by scoring models to compute the overall risk of the portfolio. A simple way to aggregate the default risk of the credit portfolio is to multiply the probability of default by the amount of capital at risk for each loan and then sum over all loans. One may use a simple definition of capital at risk, such as the total debt minus the value of the guarantee. This single measure exhibits some information about the aggregate default risk in the portfolio, which can also be used to estimate the amount of provisional reserves required for the portfolio (cf. GaTa97, P. 31).

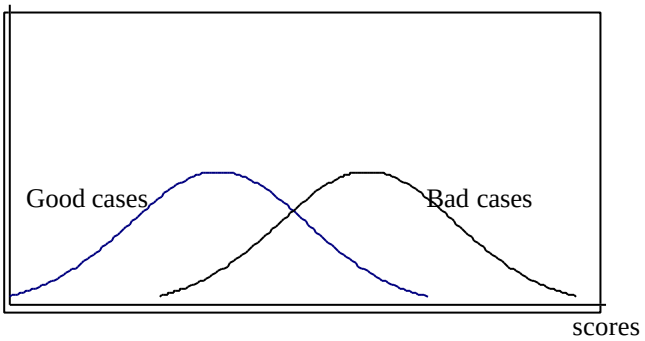
Another important issue with model application is the rebuilding of models. The scoring models based on stationary data will be not up-to-date due to the changing economic and market conditions. In order to make models robust to the population drift, it is necessary to dynamically rebuild models after a period of time. The frequency of model rebuilding depends on specific application problems (consumer or business credit) and the variations in economic conditions (economic fluctuation or stability).

It is necessary to judge whether scoring models are still up-to-date. If enough data are available, the effectiveness of scoring models can be observed through the distribution of the produced scores. Usually, a model can produce scores for 'bad' cases and 'good' cases. The distributions of the scores for 'good' and 'bad' cases show a form (see Figure 5.2.3/1-a) at the time when the model is newly built. After a period of time, the distributions may change so as to the model cannot distinct between 'good' and 'bad' cases as well as before (see Figure 5.2.3/1-b). This is a signal which shows that the model is not up-to-date and should be rebuilt.

In the model application stage, the actual payment behaviors of credit cases should be systematically recorded. These data, on the one hand, are used for validating scoring models to ensure their effectiveness: whether and to what extent the credit decisions are agreed with the real credit behaviors. The validation results generate feedback information for the revision of the scoring system. On the other hand, the real credit examples are new samples used to rebuild scoring models.



a. The distributions of scores at first



b. The distributions of scores after a period of time

Figure 5.2.3/1: The changing distributions of scores

A newly built scoring system is usually based on limited past data, while in the rebuilding process, more past cases are available, maybe with more new information. Therefore, with the increasing of the accumulated past cases and their relevant information, the credit scoring system will become more reliable and more robust. The establishment of a credit scoring system is an iterative process (cf. Figure 5.2.3/2). Maybe it will take years to collect enough data to establish a mature system, which still needs to be revised regularly to adapt to new variations in environments.

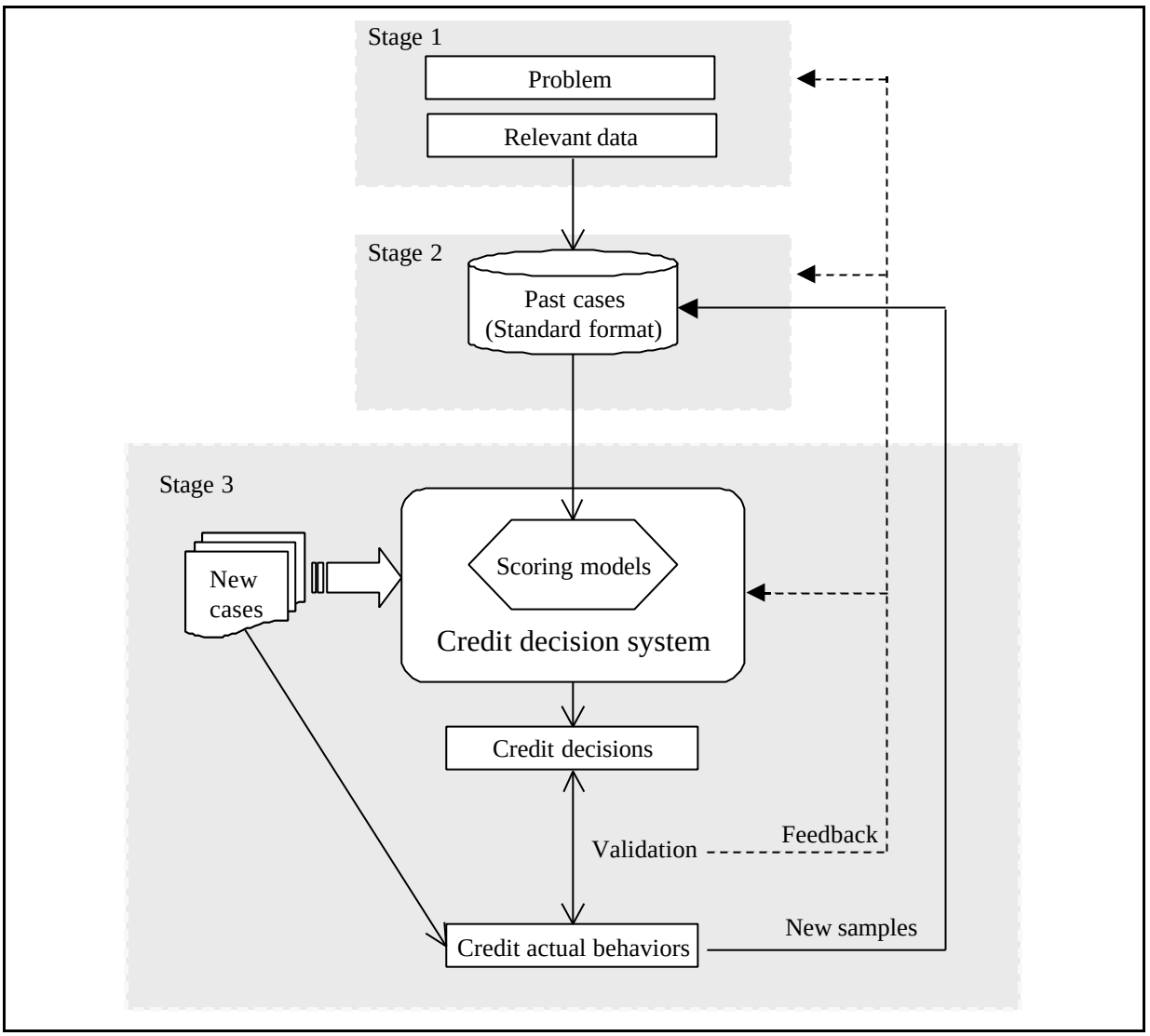


Figure 5.2.3/2: The iterative process of the scoring model building

6 Summary and conclusion

As one of the application areas of data mining, the credit scoring problem is related to the classification decision. The problem of classification decision has three professional backgrounds: statistics, machine learning, and artificial neural networks. There are many categories of classification algorithms that using two kinds of model generation approaches. Among them five are commonly used in the credit scoring model building and representative for different categories of classification techniques: Bayesian linear discriminant analysis, instance-based learning, Logistic regression, model trees M5, and Multi-layer propagation networks. They have different input requirements, different output forms and have been used for the credit scoring problem both practically and academically.

The performances of these five model techniques have been compared by many researchers. Not any of them is always superior or inferior to others. The process of credit scoring model building involves much iteration in computing in order to search for the best solution.

Based on the general process of data mining approach a general framework of credit scoring model building is presented. The framework contains the overall process of credit scoring. The practical issues in the problem definition and data preparation stage as well as the model application and validation stage are discussed.

The focus of the framework is mainly on the stage of data analysis and model building. Multiple core algorithms, some new data mining techniques such like feature selection methods and model combination techniques are incorparated into the framework. The framework serves the validation of these core algorithms and new techniques. Therefore, the evaluation subprocess is in the middle point of the framework. The criterions and methods of evaluation are used for the validation of different algorithms and techniques.

Further research would be: 1). the fine identification of detail components under the framework, especially the model evaluation criterions and methods. 2). the application of the framework on a real world credit scoring problem. The process of credit scoring model building may be best illustrated by a real world example. 3). the validation of some feature selection and model combination techniques using the framework.

Literature

AgYu99 Aggarwal, C. C./Yu, P. S.: Data Mining Techniques for Associations, Clustering and Classification. In: Zhong, N./Zhou L. Z., (eds.), Methodologies for Knowledge Discovery and Data Mining, PAKDD-99, Springer, Berlin, 1999, P. 13-23.

Baet94 Baetge, Jörg: Rating von Unternehmen anhand von Bilanzen. In: Wirtschaftsprüfung, 47(1994), 1. P. 1-10.

Baet98 Baetge, Jörg: Empirische Methoden zur Früherkennung von Unternehmenskrisen. Nordrhein-Westfälische Akademie der Wissenschaften, Natur-, Ingenieur- und Wirtschaftswissenschaften, Vorträge, N 432, Westdeutscher Verlag, 1998.

Boeh97 Boehrer, D.: Data Warehouse und Data Mining: wie Banken strategische Informationen aus ihren Daten entdecken können. Publikation der Swiss Banking School; 150: 9. Lehrgang, 1997.

Brei84 Breiman, L,/Friedenman, J. H./Olshen, R. A./Stone, C. J.: Classification and regression trees. Monterey, CA: Wadsworth, 1984.

Cabe98 Cabena, P./Hadjinian, P./Stadler, R./Verhees, J./Zanasi, A.: Discovering Data Mining: From Concept to Implementation. Prentice Hall PTR, Upper Saddle River, New Jersey, 1998.

Cios98 Cios, K. J./Pedrycz, W./Swiniarski, R. W.: Data Mining Methods for Knowledge Discovery. Kluwer Academic Publishers, Boston, 1998.

Desa96 Desai, V. S./Crook, J. N./Overstreet, G. A.: A comparison of neural networks and linear scoring models in the credit union environment. In: European Journal of Operational Research 95 (1996), P. 24-37.

Desa97 Desai, V. S./Conway, D. G./Crook, J. N./Overstreet, G. A.: Credit Scoring models in the credit-union environment using neural networks and genetic algorithms. In: IMA Journal of Mathematics Applied in Business and Industry (1997) 8, P. 323-346.

FaUt95 Fayyad, U. M./Uthurusamy, R.: Preface. In: Fayyad, U. M./Uthurusamy, R., (eds.), Proceedings of the first International Conference on Knowledge Discovery & Data Mining, AAAI Press, Menlo Park, California, 1995.

Fayy96 Fayyad, U. M./Piatetsky-Shapiro, G. /Smyth, P.: From Data Mining to Knowledge Discovery: An overview. In: 1994 KDD workshop, Fayyad, U. M./ Piatetsky-Shapiro, G./Smyth, P./Uthurusamy, R., (eds.), Advances in Knowledge Discovery and Data Mining, AAAI Press/MIT Press, Menlo Park, California, 1996, P. 1-34.

Feid92 Feidicker, Markus: Kreditwürdigkeitsprüfung - Entwicklung eines Bonitätsindikators, dargestellt am Beispiel von Kreditversicherungsunternehmen. IDW- Verlag GmbH, Düsseldorf 1992.

FoSt01 Foster, D. P./Stine, R. A.: Variable Selection in data mining: Building a predictive model for bankruptcy. Working papers 01-05, Wharton Financial Institutions center. Wharton school of university of Pennsylvania. February 26, 2001. Downloaded from: http://fic.wharton.upenn.edu/fic/papers/01/0105.pdf, 14.11.2001.

Fran98 Frank E./Wang Y./Inglis S./Holmes G./Witten I.H.: Using model trees for classification. Machine Learning, 32(1) 1998, P. 63-76.

Fraw91 Frawley, W. J./Piatetsky-Shapiro, G./Matheus, C. J.: Knowledge Discovery in Databases: An Overview. In: 1989 KDD workshop, Piatetsky-Shapiro, G./ Frawley, W. J., (eds.), Knowledge discovery in Databases, Menlo Park, California, AAAI Press, 1991, P. 1-27.

FrHo98 Fritz, S./Hosemann, D.: Behaviour Scoring for Deutsche Bank's German Corporates. In: Gholamreza Nakhaeizadeh, et al. (eds.), Application of machine learning and data mining in finance, 10th European Conference on Machine Learning, Workshop notes; 24. April 1998, TU Chemnitz, P. 179-194.

FrSc01 Frank, Ulrich/Schauer, Hanno: Software für das Wissensmanagement. In: WISU - Das Wirtschaftsstudium 30 (2001), 5, P. 817-726.

GaTa97 Galindo, J./Tamayo, P.: Credit Risk Assessment using Statistical and Machine Learning Basic Methodology and Risk Modeling Application. In: Proceedings of Computational Economics'97 conference, 1997. Downloaded from: http://otn.oracle.com/products/datamining/pdf/credit_risk.pdf, 4.12.2001

HaHe97 Hand, D. J./Henley, W. E.: Statistical Classification Methods in Consumer Credit Scoring : a Review. In: J. R. Statist. Soc. A (1997), 160, Part 3, P. 523-542.

HeHa97 Henley, W. E./Hand, D. J.: Construction of a k-nearest-neighbour credit-scoring system. In: IMA Journal of Mathematics Applied in Business and Industry (1997)8, P. 305-321.

Hofm90 Hofmann, V. H. J.: Die Anwendung des CART-Verfahrens zur statistischen Bonitätsanalyse von Konsumentenkrediten. In: Zeitschrift für Betriebwirtschaft, 60(1990), 9, 941-962.

Joos98 Joos, P./Vanhoof, K./ Ooghe, H./Sierens, N.: Credit classification: a comparison of logit models and decision trees. In: Gholamreza Nakhaeizadeh et al. (eds.), Application of machine learning and data mining in finance: European Conference on Machine Learning (ECML'98), Workshop notes; 24. April 1998, TU Chemnitz, P. 59-70.

KaCo97 Kattan, MW./Cooper, RB.: The predictive accuracy of computer based classification decision techniques, a review and research directions. In: Omega, the International Journal of Management Science, 1997, Vol.26, No.4, P. 467-482.

Kenn98 Kennedy, R. L./Lee,Y. C./Van Roy, B./Reed, C. D./Lippmann, R. P.: Solving Data Mining Problems through Pattern Recognition. Prentice Hall, PTR, 1998.

Krau93 Krause, C.: Kreditwürdigkeitsprüfung mit Neuronalen Netzen. IDW- Verlag GmbH, Düsseldorf, 1993.

Kron98 Kronborg, D./Tjur, T./Vincents, B.: Credit scoring: Discussion of methods and a case study. Research report, CBS, Copenhagen Business School, Department of Management Science and Statistics, 1998.

Leke93 Leker, Jens: Fraktionierende Frühdiagnose von Unternehmenskrisen - Bilanzanalysen in unterschiedlichen Krisenstadien. Verlag Dr. Otto Schmidt, Köln, 1993.

Liuy01 Liu, Y.: New Issues in Credit Scoring Application. Research paper, Institute of Information Systems, University of Goettingen, Nr. 16/2001, Göttingen.

Mich94 Michie, D./Spiegelhalter, D. J./Taylor, C. C.: Machine Learning, Neural and Statistical Classification. Horwood, New York, 1994.

Moxo96 Moxon, B.: Defining Data Mining. In: DBMS ONLINE, DBMS Data Warehouse Supplement, August 1996. Downloaded from: http://www.dbmsmag.com/9608d53.html, 05.12.2000.

Müll96 Müller, J.: DV-gestützte Systeme zur Kreditwürdigkeitsprüfung bei Kreditversicherungen. Göttinger Wirtschaftsinformatik, Band 23, Unitext Verlag Göttingen, 1997.

Nieh87 Hiehaus, H.: Früherkennng von Unternehmenskrisen. IDW-Verlag GmbH, Düsseldorf, 1987.

Pira99b Piramuthu, Selwyn: Feature selection for financial credit-risk evaluation decisions. In: INFORMS Journal on computing, Vol. 11, No. 3, Summer 1999.

Quin93 Quinlan, J. R.: C4.5, Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, California, 1993.

RoGl94 Rosenberg, E./Gleit, A.: Quantitative methods in credit management: A Survey. In: Operations Research, Vol. 42, No. 4, 1994, P. 589-613.

Thom00 Thomas, L. C.: A survey of credit and behavioural scoring: forcasting financial risk of lending to consumers. In: International Journal of Forecasting 16 (2000), P. 149-172.

WeIn98 Weiss, S. M./Indurkhya, N.: Predictive Data Mining: A Practical Guide. Morgan Kaufmann Publishers, Inc. San Francisco, California, 1998.

WeKu91 Weiss, S. M./Kulikowski, C. A.: Computer systems that learn: Classification and prediction methods from statistics, neural nets, machine learning, and Expert Systems. Morgan Kaufmann Publishers, Inc. San Mateo, California, 1991.

West00 West, D.: Neural network credit scoring models. In: Computer & Operations research 27(2000), P. 1131-1152.

WiFr00 Witten, I. H./Frank, E.: Data Mining: Practical machine learning tools and techniques with Java implementations. Morgen Kaufmann Publishers, San Francisco, 2000.

Yoba00 Yobas, M. B./Crook, J. N./Ross, P.: Credit scoring using neural and evolutionary techniques. In: IMA Journal of Mathematics Applied in Business and Industry (2000)11, P. 111-125.