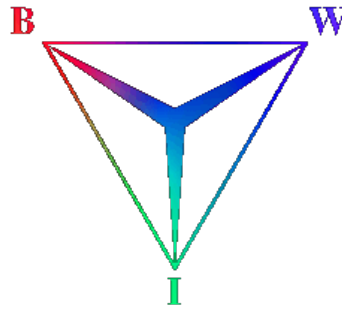


Credit Scoring

An overview of traditional methods and recent improvements.



C.A.M. Schoemaker

BMI Paper

Vrije Universiteit
Faculty of Exact Sciences
Business Mathematics and Informatics
De Boelelaan 1081a
1081 HV Amsterdam



December 2006

Preface

This paper is one of the final compulsory deliverables of the master studies Business Mathematics & Informatics. Its aim is to combine mathematics and informatics in a literature study that has a clear business focus and for the student to learn to write a thesis in a scientific way.

One can think of many possible topics, but it is not necessarily easy to find a topic that combines the three perspectives. Neural networks though is indeed such a topic: it belongs to the Artificial Intelligence department which is part of Computer Science, it has an often very complicated mathematical background, but its purposes are often very business related. I became interested in this subject during the master course Neural Networks and since at that time I already knew I was to do my internship at a bank, I wanted to combine those two fields. This is how I came across literature about credit scoring and relating this with neural networks seemed perfect for a literature study.

I enjoyed reading and writing about this subject and writing this thesis and I want to thank Elena Marchiori for her enthusiasm and her time and advice.

Kind regards,

Anne Schoemaker

Executive Summary

Credit risk evaluation is a very challenging and important problem in the domain of financial analysis. It is an opportunity but inadequate credit scoring can also cause great problems for a credit granting institution. This is why much research has been done for many years to find new ways of scoring credit applicants in a way that brings profit for the lender. Statistical methods have been used for many years, but their main drawback is the fact that they require a structure for the data. Neural networks deal with this shortcoming of statistical methods, as they do not require any pre-specified structure. A major disadvantage of neural networks though is their lack of explanatory power, which makes them hard to understand and to interpret for credit scorers. They are not able to explain to a client why he or she was granted or denied a loan.

This paper strives to give an overview of methods that are based on neural networks and hence have the same advantages, but deal with the explanatory disadvantage of neural nets. Support Vector Machines are a fairly new development and research showed that it has high classification accuracies and besides that it is not too hard to explain them mathematically. They also have the advantage that neural nets have too that they are robust. Then again, they do not provide a solution to the lack of explicability of neural network models. Rule-extraction techniques explicitly deal with opening the 'black box' of neural nets. Their classification accuracy is just as high or at least comes very close, but the rules the extract from the models are easy to interpret and easy to use in daily business. They do however need more time and effort to be implemented. Hybrid models can come in helpful, as they shorten the time it takes to train a neural network. Then again they are not a solution to the limitations of neural network models. For faster convergence of neural network models, the relative importance of input variables can be computed with the formulas that will be given in the respective chapter.

Management Samenvatting

Het evalueren van krediet risico is een belangrijk maar gecompliceerd probleem in het domein van de financiële analyse. Het verstrekken van een lening kan een kredietverstrekker winst opleveren, maar de keerzijde is dat onvoldoende credit scoring ook voor grote problemen kan zorgen. Vanwege het belang van dit onderwerp is er in de laatste decennia veel onderzoek gedaan naar het vinden van steeds betere methodes voor krediet risico bepaling. Jarenlang hebben statistische methodes de overhand gehad, maar een groot nadeel daarvan is dat een structuur opgelegd moet worden aan de data. Neurale netwerken zijn een relatief nieuwe ontwikkeling, die dit nadeel niet hebben. Een groot nadeel ervan is echter dat het moeilijk is een neuraal netwerk te doorgronden en te begrijpen waarom een beslissing is genomen. Dit maakt dat het lastig is voor een kredietverstrekker om uit te kunnen leggen aan een cliënt waarom hij of zij wel of niet een lening verstrekt krijgt.

Deze paper biedt een overzicht van methodes die gebaseerd zijn op neurale netwerken en daarmee de voordelen daarvan hebben, maar die wel een oplossing kunnen bieden voor de tekortkomingen van neurale netwerken. Support Vector Machines zijn een vrij recente ontwikkeling en onderzoek heeft aangetoond dat het hoge classificatie accuraatheid levert. Daarbij is het relatief eenvoudig deze methode wiskundig te begrijpen. Ook hebben ze het voordeel dat neurale netwerken ook hebben, namelijk dat ze robuust zijn. Daarentegen vormen ze nog geen oplossing voor het probleem van neurale netwerken dat ze moeilijk te doorgronden en begrijpen zijn. Rule-extraction technieken pakken dit probleem juist wel aan. De classificatie accuraatheid is haast zo hoog als die van neurale netwerken, maar de regels die deze technieken uit de netwerken halen zijn eenvoudig te interpreteren en te begrijpen. Het nadeel is dat het vrij veel tijd en expertise kost deze methode te implementeren. Hybride modellen zijn geen oplossing voor de tekortkomingen van neurale netwerken, maar kunnen wel handig zijn bij de implementatie, aangezien zij de tijd die het kost de netwerken te trainen aanzienlijk kunnen verkorten. Voor snellere convergentie van neurale netwerk modellen kan de relatieve importantie berekend worden van iedere input variabele met bepaalde formules.

Contents

1	Introduction	- 8 -
2	History of credit scoring	- 8 -
3	Relevant indicators in retail credit scoring models	- 9 -
4	Traditional methods for credit scoring	- 11 -
4.1	Linear regression	- 11 -
4.2	Linear Discriminate Analysis (LDA)	- 11 -
4.3	Logistic regression.....	- 12 -
4.4	Probit analysis.....	- 12 -
4.5	Linear programming	- 12 -
4.6	Classification trees	- 13 -
4.7	Nearest neighbours.....	- 14 -
5	Traditional methods compared.....	- 14 -
6	Neural network models	- 16 -
7	Advantages and limitations of neural networks	- 18 -
8	Improvements to neural network models.....	- 18 -
8.1	Rule-extraction techniques	- 19 -
8.1.1	Neurorule	- 19 -
8.1.2	Trepan	- 19 -
8.1.3	Nefclass	- 20 -
8.1.4	Outcomes.....	- 20 -
8.2	Support Vector Machines.....	- 22 -
8.3	Neuro-fuzzy systems.....	- 24 -
8.4	Hybrid Neural Discriminant Technique.....	- 26 -
8.5	Relative importance of input variables.....	- 26 -
9	Conclusion	- 27 -
10	Bibliography	- 28 -

1 Introduction

Credit risk evaluation is a very challenging and important problem in the domain of financial analysis. Companies put themselves at risk by lending money, but it also creates opportunities. If a loan is granted to an applicant and this applicant goes into default, the granting institution loses its money. On the other hand, if the institution fails to recognize an applicant as being possibly profitable, it will lose money by not having granted him that loan. Then again, a client that is never late with his payments, the institution does not receive interest, so a perfect applicant is not all that profitable either. It is therefore of vital importance to credit granting institutions to have proper methods for credit scoring. Then there is also the advent of Basle II that has put the spotlight on the need to be able to model the credit risk of portfolios of consumer loans not just the risk of each loan independently defaulting. The proposed new regulations require the lenders to provide equity capital as a function of the risk of the portfolio of the loans being funded where the portfolio can be split into appropriate risk segments.

For many years, credit scoring has been done using traditional, mostly statistical methods, but recently neural networks have received a lot of attention in this domain, because of their universal approximation property. However, a major drawback of neural networks for decision making is their lack of explanation capability. Even though they can achieve a high predictive accuracy rate, they are not always the best option since the reasoning behind how they reach their decisions is not readily available. Neural networks are usually complex, hard to understand and hard to explain. This is a feature that is very desirable in the credit granting domain though. Lately research has been done to find ways to make this possible and in this paper some of these methods developed and researched will be discussed: rule extraction, Support Vector Machines, neuro-fuzzy models, and hybrid learning.

This paper strives to give an overview of more traditional credit scoring methods, as well as neural network models for this purpose and innovations and improvements in this area. The first chapter describes shortly the history of credit scoring. Then a chapter is included that describes the indicators that are commonly used in credit scoring. Next, some of the most commonly used traditional credit scoring methods will be explained, followed by their advantages and shortcomings. Chapter 6 is about neural networks, the basics, and chapter 7 about their opportunities and shortcomings. The next section deals with some improvements recently made to neural network models when it comes to credit scoring. After that there is a short conclusion to it all.

2 History of credit scoring

It is said that consumer credit dates back to the time of the Babylonians, so about 3000 years ago. From these Babylonians, to the pawnbrokers and usurers in the Middle Ages, to consumer credit nowadays, the lending to the mass market of consumers is something from the last fifty years.

Credit scoring itself is basically a way of finding an underlying structure of groups in a population, where one cannot see this structure but only related characteristics. By finding this structure in a dataset of previous credit applicants, one tries to infer rules with which it is

possible to classify a new credit applicant as creditworthy or not. Fisher was in 1936 the first to come up with the idea of discriminating between groups in a population in statistics. He used this for completely different purposes and in 1941 David Durand was the first to apply the same techniques for discriminating between good and bad loans, but it had not yet been used for prediction.

At the same time some of the finance houses and mail order firms had problems managing their credit. Their credit analysts had been drafted into military service and the decisions on whether or not to grant a loan or send merchandise had always been made by these analysts. The firms had them write down their principles and rules of thumb, so they could continue their business. Not long after the war the possible usefulness of statistically derived methods for this purpose was recognized generally and it did not take long for the first consultancy to be formed.

The arrival of credit cards in the late sixties meant a big rise in the number of credit applicants, and both economic and in manpower terms it was no longer possible to do all this manually. The need to automate these decisions was expressed. Although many people felt that this kind of decisions, that had hitherto always been made by human experts, could not be made by machines, default rates dropped by 50%.

The passing of the *Equal Credit Opportunity Acts* (ECOA, 1975, 1976) made sure credit scoring could not be discriminative on for example race or sex, because it states that any discrimination can only be made based on statistical assessments. By this Act credit scoring was fully recognized in the USA.

In the 1980's banks started using credit scoring in other domains, like personal loans, for it's successes. Later they also started using it for mortgages and small business loans. Another important application is in the marketing domain. Since the growth of direct marketing in the 90's, scorecards are used to improve the response rate to advertising campaigns.

With the advance of computational skills, other techniques to develop scorecards were tried. In the 1980's two of the most important ones were introduced: logistic regression and linear programming. Recently, most of the emphasis is on artificial intelligence techniques like neural nets and expert systems.

Another recent development is the change in objectives: from minimizing the risk of clients defaulting, the emphasis now is more on maximizing profit made from a customer. This means companies will try to identify the customers that are most profitable.

Where credit scoring is a technique to decide whether or not to grant a loan to a new applicant, behavioural scoring are techniques that deals with existing customers. Should a firm agree to increase a customers credit limit or to grant him another loan? If a customer starts to fall behind in his repayments, what should the firm do? What kind of marketing should the firm aim at that customer? Behavioural scoring deals with this kind of issues. This topic nevertheless is not within the scope of this paper.

This chapter is a summary of papers [20], [27], [28], [29],

3 Relevant indicators in retail credit scoring models

To be able to say whether an applicant is credit worthy or not, one needs to design a model specification containing the right variables. Since the available data sets are usually very large, the risk of over fitting is quite small. Therefore it would theoretically be possible to choose a large number of characteristics to use. In practice, this does not work. Too lengthy a

procedure or questionnaire will deter clients. A standard statistical strategy also used in pattern recognition is to begin with a large number of characteristics and then identify an appropriate subset. In credit scoring the following three ways are most common:

- Using expert knowledge. An expert in this area has experience and feeling for it and this provides a good complement to formal statistical strategies. The latter are necessary to identify those characteristics that have predictive value. The former is needed if later on one is asked to justify the chosen selection of characteristics. There is a trade-off between simplicity and complexity: too complex is unacceptable, even though it may outperform simpler models, but too simple models have low performance.
- Using stepwise statistical procedures. For instance, forward stepwise methods sequentially add variables that improve the predictive accuracy.
- Third, individual characteristics can be selected using a measure of the difference between the distributions of the good and the bad risks on that characteristic. An example of such a measure that is commonly used is the *information value*:

$$\sum_j (p_{ij} - q_{ij})w_{ij}$$

w_{ij} are the weights of evidence, where, the j -th attribute of the i -th characteristic is given by $w_{ij} = \ln(p_{ij}/q_{ij})$ where p_{ij} is the number of those classified as ‘good’ in attribute j of characteristic i , divided by the total number of good risks and similarly q_{ij} is a proportion of bad risks in attribute j of characteristic i .

- Last, one can run ANOVA on the data set and eliminate those variables with a relatively high p -value.

These variables need to have the following two features: to be sound in helping to estimate an applicant’s probability of default; second, their explanatory power for analysis of a loan application. In retail credit scoring there are four main categories in which the variables that are most commonly used can be divided: demographic indicators, financial indicators, employment indicators and behavioural indicators. Table 1, drawn from [31], gives an overview of these indicators.

Demographic Indicators	Financial Indicators	Employment Indicators	Behavioural Indicators
1 Age of borrower	1 Total assets of borrower	1 Type of employment	1 Checking account (CA)
2 Sex of borrower	2 Gross income of borrower	2 Length of current employment	2 Average balance on checking account
3 Marital status of borrower	3 Gross income of household	3 Number of employments over the last x years	3 Loans outstanding
4 Number of dependants	4 Monthly costs of household		4 Loans defaulted or delinquent
5 Home status			5 Number of payments per year
6 District of address			6 Collateral / guarantee

Table 1. Indicators that are typically important in retail credit scoring models [30]

The first category is not the most important one, but it is used because it captures various regional, gender and other differences. In general, the older the borrower, the lower its default

probability and it is also lower for applicants that are married. Since homeowners have a house as collateral, this is less risky group too.

It is clear that the second category, financial indicators, is an important one. It says something about the possibility of repayment by the borrower, through considering the incomes and costs, the available resources for the household and such.

Employment indicators are the third set. Generally, people that are self-employed have a lower rating, as well as people who frequently change (low-skilled) jobs.

The last category, behavioural indicators, is the most important one. All these data are known by the bank, and because of their importance in estimating future possible default, they often share this information. A bank can easily check average balances of checking accounts, their in- and outflow, and whether or not an applicant has already been granted a loan before. The factor with the most influence is collateral. Real estate serves as the best guarantee, for the threat of losing one's house in case of default is a critical factor for a client and therefore has great influence on the client's decision the repay.

4 Traditional methods for credit scoring

The next part gives an overview of some of the most important traditional methods used for credit scoring.

4.1 Linear regression

The linear regression approach to linear discrimination relates p , the probability of default, with the application characteristics X_1, X_2, \dots, X_n in the following way:

$$p = w_0 + w_1 X_1 + w_2 X_2 + \dots + w_n X_n \quad (1)$$

where w_0, w_1, \dots, w_n are the weights.

An obvious problem with this approach is that the right hand side of the equation can have values between $-\infty$ and $+\infty$, whereas for the left hand side we want a probability, hence with values between 0 and 1. For coefficient estimation, ordinary least-square estimation can be used.

4.2 Linear Discriminate Analysis (LDA)

Discriminant analysis is based on linear regression. It involves the linear combination of the two (or more) independent variables that differentiate best between the (a priori defined) groups. For this the statistical decision rule of maximizing the between-group variance relative to the within-group variance. The linear combinations are derived from the following equation:

$$Z = w_1 X_1 + w_2 X_2 + \dots + w_n X_n$$

Where Z is the discriminant score, \mathbf{w}^T the discriminant coefficient and \mathbf{X} the independent variables.

When the dependent variable is categorical and the independent variables are metric, LDA is the appropriate technique to be used. By averaging the discriminant scores (Z) for all individuals within a particular group, one finds the group mean and this is referred to as *centroid*. In the case of credit scoring, where there are two groups –good and bad- there are two centroids. Statistical significance of this function is tested with a generalized measure of the distance between group centroids.

Discriminant analysis has been used in a wide range of problems. Application areas are for instance medicine, business, education, marketing research, finance, chemistry, biology, engineering and archaeology [29].

4.3 Logistic regression

Logistic regression deals with the problem of regression that the right hand side of the equation can have values between $-\infty$ and $+\infty$, whereas for the left hand side we want a probability, hence with values between 0 and 1, by matching the log of the probability odds with the right hand side of Equation 1.

$$\log\left(\frac{p}{1-p}\right) = w_0 + w_1X_1 + w_2X_2 + \dots + w_nX_n$$

For estimating the coefficients of corresponding parameters, maximization of the likelihood function is usually applied as a convergent criterion.

Logistic regression models are widely used in areas like social research, medical research, biological research, food science, design, control, bankruptcy prediction, market segmentation and customer behaviours.

Both linear regression and logistic regression are very sensitive to correlations between predictive variables, so it is important to eliminate strongly correlated variables from the data set on which regression will be performed. The classification results for both methods are very similar.

4.4 Probit analysis

Probit analysis is another non-linear regression approach [9]

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-y^2/2} dy$$

where the aim is to estimate $N^{-1}(p_i)$ as :

$$N^{-1}(p_i) = w_0 + w_1X_1 + w_2X_2 + \dots + w_nX_n,$$

the linear function of the applicant's characteristics.

4.5 Linear programming

Supposing that one has a sample of n_G goods and n_B bads, this method seeks to develop a scorecard where all the goods are above some threshold, or cut-off, and all the bads are below it. For it is not always possible to find such a scorecard with no errors, slack variables a are

introduced which allow for possible errors. Now, we want to find the weights w_1, w_2, \dots, w_m to be applied to the characteristics, that minimize the sum of the absolute values of the errors. This gives us the following linear program:

$$\begin{aligned}
 & \text{minimize} \\
 & a_1 + a_2 + \dots + a_{n_G+n_B} \\
 & \text{subject to} \\
 & w_1x_{i1} + w_2x_{i2} + \dots + w_mx_{im} \geq c - a_i, 1 \leq i \leq n_G \\
 & w_1x_{i1} + w_2x_{i2} + \dots + w_mx_{im} \leq c + a_i, n_G + 1 \leq i \leq n_G + n_B \\
 & a_i \geq 0 \\
 & 1 \leq i \leq n_G + n_B
 \end{aligned}$$

where c is the cut-off. [27]

4.6 Classification trees

Where in the methods above, weights are given to each answer and are added which ends up in a scorecard, classification trees divide the customer group into different subsets, each homogeneous in their default risk and different from the default risk of other groups. There are different ways for splitting the set into different subsets, but the main goal is to keep splitting until it is no longer possible to split into two new subgroups that are statistically significantly different. After splitting is done, the groups are classified as good or bad and every new applicant will be classified based on the group it ends up in when utilizing the tree. A classification tree is sometimes called an expert system, as it is an automated procedure that has the ability to learn. [27]. Figure 1 is an example of such a decision tree.

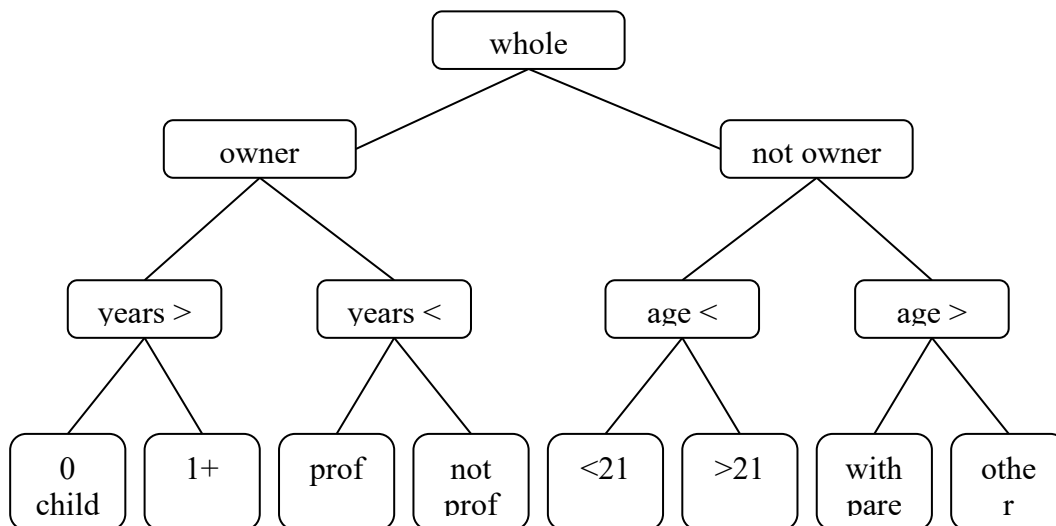


Figure 1 Example of a classification tree for credit scoring [27].

The criterion in the first layer is *residential status*, in the second it is *years at bank* and *age* respectively, in the third it is *number of children*, *employment*, *age*, *residential status* respectively.

4.7 Nearest neighbours

This is a non-parametric statistical approach. One chooses a metric on the space of application data to measure how far apart applicants are. Now all applicants in past data have a position on this metric space. Each new applicant is classified as good or bad according to which group is in the majority among the nearest neighbours of this new applicant.

Determining a suitable metric for this approach is quite similar to developing a linear scorecard.

When using this method, a very important step is the choice of the metric to be used. Henley and Hand [14] describe both the choice of this metric as well as the choice of the number of neighbours used. The *Euclidian Norm* is a commonly used metric, described by:

$$\rho_1(x, y) = \sqrt{(x - y)^T (x - y)}$$

where x and y are measurement vectors.

When the variables are in different units or when they are categorized¹, however, one needs a data- dependent version of the *Euclidian Norm* like:

$$\rho_2(x, y) = \sqrt{(x - y)^T A (x - y)}$$

where A is a $n \times n$ matrix with n number of variables. Two sorts of metrics are possible, as A can depend on x : local metrics where A depends on x and global metrics where it does not. Also, it is necessary to use some appropriate standardization of variables.

The choice of k , the number of nearest neighbours, determines the bias/variance trade-off in the estimator. A restriction is that k has to be smaller than the smallest class. Enas and Choi [6] suggest that $k \approx n^{2/8}$ or $n^{3/8}$ is reasonable. In empirical literature though, k is often found by crossvalidation on the misclassification rate. Here crossvalidation is done by dividing the training set into m subsets, where $m-1$ sets are used for training and the last one for testing. This is repeated for each subset. More details on how to choose k can be found in Hand and Vinciotti [11] and Holmes and Adams [16].

5 Traditional methods compared

A large number of studies make it clear that overall there is not much statistically significant difference between the prediction accuracies of the methods summarized above. An example of a study that acknowledges this is the one performed by Baesens [1]. He made a thorough study of different methods using 8 different consumer and small business data sets. Table 2 shows the average ranking in percentage correctly classified (PCC) of the following methods: Linear regression, logistic regression, linear programming, four variants of support vector machines, four variants of classification trees, two variants of nearest neighbours, neural net, naïve Bayes. Naïve Bayes assumes that the probability of being good is, in the basic case, the

¹ Variables may be categorized and measured on different scales. To overcome this problem, one can introduce dummy variables, or use so-called weights of evidence (w_{ij}). Here, the j -th attribute of the i -th characteristic is given by $w_{ij} = \ln(p_{ij}/q_{ij})$ where p_{ij} is the number of those classified as ‘good’ in attribute j of characteristic i , divided by the total number of good risks and similarly q_{ij} is a proportion of bad risks in attribute j of characteristic i .

independent product of the conditional probabilities of being good dependent on each characteristic value in turn.

Method	Lin regr	Lin progr	SVM	NN	Naïve Bayes	Class trees	Nearest neighbours
Average ranking in PCC	6.9	6.1	6.5	3.6	15.1	5.2	6.7

Table 2 Average ranking of methods

It can be seen that there is not much difference between the different methods, except that naïve Bayes performs significantly worse than the others.

But besides classification accuracy, the different methods can also be compared on a different level. Each has its own advantages and shortcomings.

LDA and linear regression have often been criticized because of their assumption of the categorical nature of the credit data and the fact that the covariance matrices of the good and bad credit classes are unlikely to be equal [24]. It was found that logistic regression, again, serves as a good alternative and is as effective as LDA. Harrel and Lee [12] on the other hand, found that logistic regression is just as efficient and accurate as LDA, even though the requirements for LDA are satisfied.

Another issue of LDA are the data requirements: the data should be independent and normally distributed while the covariance matrix is also required to comply with the variation homogeneity assumption. If the covariance matrices of the given populations are not equal, then the separation surface of the discriminant function is quadratic and hence in this case the quadratic discriminant analysis (QDA) needs to be used [17]. Despite LDA being a special case of QDA with stronger assumptions that should restrict the applications, LDA has reported to be a more robust method when the restrictions are violated [5] [25] [26]. On the other hand, Hand and Henley [10] say the following about this matter: “Certainly, if the variables follow a multivariate ellipsoidal distribution (of which the normal distribution is a special case), then the linear discriminant rule is optimal (ignoring sampling variation). However, if discriminant analysis is regarded as yielding that linear combination of the variables which maximizes a particular separability criterion, then clearly it is widely applicable. The normality assumption only becomes important if significance tests are to be undertaken.”

An advantage of LDA is that ordinary least-square estimation can be used to estimate the coefficients of the linear discriminant function, where logistic regression requires maximum likelihood methods of estimation of logistic regression models.

Regression approaches in general have the advantage that they allow performance of statistical tests to identify which questions/characteristics are important for the classification accuracy or if two questions/characteristics basically say the same thing and therefore get equivalent results. This way spurious questions can be eliminated which makes it possible to drop unnecessary questions and this makes the model a lot more robust.

Comparing linear programming and regression methods, linear programming can deal with a lot more variables than regression approaches. Another advantage is that it allows the scorecard designer to ensure that scores have properties that the lending organisation wants to

be included. It is hard to find the best scorecard with this property using statistical methods, but with linear programming it is quite easy.

Classification trees on the other hand are good at finding non-linear relationships within the data, which may cause errors in linear scorecards if they are not recognized. Both regression approaches and linear programming suffer from this limitation.

One of the attractive features of nearest neighbours is that it can be used to overcome population drift (a change in time in the distribution of a population). The method can be dynamically updated by adding applicants, of whom the true classes are known, to the design set and dropping old applicants. Henley and Hand [14] also identified the advantage of the method that it enables modelling of irregularities in the risk function over the feature space, due to its non-parametric nature. The k -NN method has been found to perform better than other non-parametric methods like kernel methods when the data are multidimensional. Another advantage is that this method is quite easy to understand. It is a fairly intuitive procedure.

A problem that yields for all statistical methods for credit scoring, is that they assume multivariate normality for independent variables and these assumptions are frequently violated in financial data sets, as described earlier in this section. This makes that these methods are theoretically invalid for finite samples. The major difference between statistical methods and machine learning techniques is that in the latter it is not necessary for investigators to impose structures to different models and to estimate parameters so that the model fits the data. Machine learning techniques on the other hand allow the model to learn the structures. Even though not all researchers agree on the topic of validity of statistical models when data sets do not comply all constraints, it is a clear advantage of machine learning models that they do not depend on this kind of restrictions. Therefore, neural network models (a special case of machine learning techniques) will be covered in the next sections.

6 Neural network models

Simply said, neural networks mimic the processing capabilities of the human brain. It is composed of neurons, which are the processing units, inspired by the human biological nervous system. A neuron receives input, processes this input and delivers output. A neural network model can contain none, one or several hidden layers, depending on its purpose. Previous studies have been shown though that every bounded, piecewise continuous function can be approximated with an arbitrarily small error by a network with one hidden layer; any continuous function can be approximated to an arbitrarily small accuracy by a network with two hidden layers. Therefore networks with one, or sometimes two, hidden layers are most commonly used. This design is called Multilayer Perceptron.

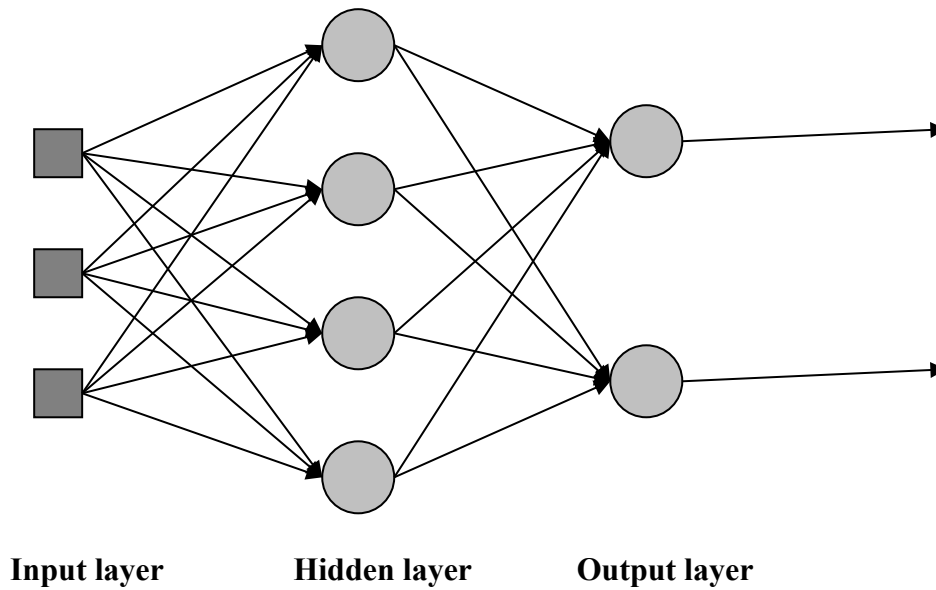


Figure 2 Example of a Multi Layer Feed Forward Neural Network (3-4-2).

Figure 2 is an example of a Feed Forward Neural Net, with one hidden layer. Each hidden neuron gets input from the input nodes, and each output node receives input from the hidden layer. A neuron can be described by:

1. a set of links that describe the neuron inputs, with weights w_1, w_2, \dots, w_m .
2. a linear combiner for computing the weighted sum of the inputs:

$$u = \sum_{j=1}^m w_j x_j$$

3. and an activation function (also called squashing function) φ for limiting the amplitude of the neuron output $y = \varphi(u+b)$, where b is the bias

Weights numerically represent the connection between two neurons. The higher the weight, the more important that specific connection.

The choice of the activation function (neuron model) depends on the data. Examples of these models are the hard-limiter, the piecewise linear model, the sigmoid function, the hyperbolic tangent, step function and Gaussian function. A typical activation function used in Feed Forward Neural Networks (FFNN) is the sigmoid function:

$$\varphi(v) = \frac{1}{1 + \exp(-av_j)} \quad \text{with } a > 0,$$

where $v_j = \sum_i w_{ji} y_i$, with w_{ji} the weight of the link from node i to node j and y_i the output of node i .

The classical learning algorithm of FFNN is based on the gradient descent method, and this method requires a function of the weights that is continuous and differentiable everywhere.

A neural network aims at minimizing the error, i.e. the difference between the real output and the output generated by the network². For this it computes the output and compares this with the desired output. As long as the error found does not meet the demands (which can be pre-specified), the network will continue learning by updating its weights. This updating can be done in several ways, depending on (amongst other parameters) the learning algorithm and the network architecture.

7 Advantages and limitations of neural networks

Neural networks are quite robust, for a number of reasons. First of all, as is not the case with for instance statistical methods, it is not necessary to make an analysis of a problem or of the internal structure of a system, based on in-depth knowledge. Second, they can handle great amounts of data, without any problem. Another important advantage is that neural nets do not have restrictions on the data, other than continuity and differentiability. It does not assume any probability distribution like normality or equal dispersion and covariance matrix requirements. The fact that neural networks quite easily adapt to changes in the ‘real world’, since they can modify the network’s weights and learning rate, also makes them robust.

Then, of course, in many studies neural network models have shown to have better prediction accuracies than statistical models in general.

An important drawback of neural nets is that development of a neural network requires quite a lot of expertise from the user, more than statistical models. The reason for this is that the network is relatively sensitive to changes in parameters like the learning rate or the transfer functions, which therefore need to be chosen with care.

Second, there is no formal way to find an optimal topology for a network. The number of nodes and the number of hidden layers need to be found by repetitious experimenting.

Where statistical models tend to under fit due to the human imposed structures and are often too simple, neural networks tend to be very complex and sometimes over fit the data. They are hard to understand and to explain. As Zan Huang et al. [33] state it: “This is in fact the trade-off between the explanatory power and parsimony of a model, where explanatory power leads to high prediction accuracy and parsimony usually assures generalizability and interpretability of the model.” One of the most important limitations of neural network models is their lack of explanatory power, also referred to as their ‘black box problem’.

From a computational point of view, another disadvantage of neural network models is the time it takes to train a net. This is, still, longer than the time needed for statistical methods.

8 Improvements to neural network models

As said in the previous chapter, an important limitation of neural networks is their lack of explanation capability. Neural networks themselves do not explain how they came to a certain decision. The net does not specify the structure it has found, nor does it for example explain why a certain applicant failed. Especially for credit scoring, this can be a major drawback,

² This only yields for so called ‘supervised learning’, where actual outputs are known. The opposite is ‘unsupervised learning’, where the desired output is not known and where one tries to find a certain hidden structure in the data.

since an applicant that is rejected might want to have an explanation, which the granting institution cannot give easily based on neural nets. Because, on the other hand, previous studies have shown that neural networks are a very promising method for credit scoring, ways to clarify decisions made by neural models are being sought.

8.1 Rule-extraction techniques

In the literature, it has been tried to so to speak ‘open up the neural network’s black box’ by rule extraction techniques that generate symbolic rules with (approximately) the same predictive power as the neural network itself. Baesens et al [2] propose three algorithms that they choose for the translucency of the rule extraction algorithm and the expressive power of the extracted rules. Translucency considers the technique’s perception of the neural network.

There are two approaches:

- The decompositional approach starts by analysing the activation values, weights and biases and with that extracts rules at the level of the individual hidden and output nodes.
- The pedagogical approach considers the neural net as a black box. It relates the inputs and outputs of the network, instead of looking at the internal structure of the network. These techniques classify examples with the trained network and then add ‘artificial’ examples, which are then used by a symbolic learning algorithm to infer rules.

The expressive power of the extracted rules depends on how these rules are expressed, what language is used. Examples are propositional IF-THEN rules, fuzzy classification rules (for instance, IF X is high AND Y is medium, THEN class=1) and *M-out-of-N* rules (for instance IF at least *M* out of the *N* conditions are satisfied, THEN class=1).

The three algorithms are Neurorule, Trepan and Nefclass. A summary of these methods as well as a comparison will be given here. For more in-depth information about the algorithms the reader is referred to Baesens et al [2].

8.1.1 Neurorule

Neurorule is a decompositional algorithm. It extracts propositional rules from trained three-layered feedforward neural networks by following these steps:

Step 1. Train a neural network to meet the pre-specified accuracy requirement.

Step 2. Remove the redundant connections in the network by pruning while maintaining its accuracy.

Step 3. Discretise the hidden unit activation values of the pruning network by clustering.

Step 4. Extract rules that describe the network outputs in terms of the discretised hidden unit activation values.

Step 5. Generate rules that describe the discretised hidden unit activation values in terms of the network inputs.

Step 6. Merge the two sets of rules generated in Steps 4 and 5 to obtain a set of rules that relates the inputs and outputs of the network.

8.1.2 Trepan

Trepan is a pedagogical algorithm that extracts decision trees from trained neural networks with arbitrary architecture by using a symbolic learning algorithm. It grows a tree by recursive partitioning. An important difference with decision-tree-induction algorithms is that the latter

have a limited amount of training observations available, whereas Trepan does not use the original training observations. It re-labels the observations according to the classification done by the network. This new set is used to initiate the tree-growing process. But it can also enrich the data with additional training instances, which are then labelled by the network itself. For this, Trepan takes the distribution of the data into account.

Trepan allows splits with *at least M-of-N* type of tests, where the splits are constructed by using a heuristic search procedure. Furthermore, it uses one local and one global stopping criterion for deciding when to stop growing the tree.

8.1.3 Nefclass

Nefclass is what is also referred to as a neurofuzzy system. It is a neural network fuzzy-rule extraction technique and these techniques use learning algorithms from neural networks to tune the parameters of a fuzzy system. Its architecture is that of a perceptron with one hidden layer, but the difference between this and a regular perceptron architecture is that the weights now represent fuzzy sets and that the activation functions are now fuzzy set operators. The fuzzy rules are represented by the hidden layer and the output neurons represent the different classes of the classification problem, with one neuron per class.

Nefclass allows a classifier to be created from scratch, but it also allows a user to model a priori domain knowledge before starting the learning. Either way, the user must start by specifying the fuzzy sets and membership function types for all inputs³.

Nefclass starts by creating nodes when needed and when these do not exist yet, if the specific input pattern requires this, for all inputs. Since this results in a large number of hidden neurons and fuzzy rules, one can specify a maximum number of nodes and either keep the first k or keep the best k . After the number of hidden nodes has been determined, the fuzzy sets between the input and the hidden layer are tuned to improve the classification accuracy. It is also possible to prune the network, to make it more comprehensible [22].

8.1.4 Outcomes

The main goal of Baesens et al [2] is to develop intelligent credit-risk evaluation systems that not only have high classification accuracies, but are also comprehensible and user friendly. Therefore they also take into account the complexity of the generated rules or trees as a performance measure. The fidelity of the models is tested by comparing their accuracies with that of the neural networks they model, since the extracted rule sets are supposed to model the behaviour of the neural net.

³ These membership functions can be trapezoidal, triangular, Gaussian or List.

Data set	Method	PCC _{train}	PCC _{test}	Complexity
A	C4.5	80.63	71.56	38 leaves, 54nodes
	C4.5rules	81.38	74.25	17 propositional rules
	Pruned NN	75.53	77.84	6 inputs
	Neurorule	75.83	77.25	4 propositional rules
	Trepan	75.37	73.95	11 leaves, 21 nodes
	Nefclass	73.57	73.65	14 fuzzy rules
B	C4.5	77.76	70.03	77 leaves, 114 nodes
	C4.5rules	76.70	70.12	17 propositional rules
	Pruned NN	73.05	71.85	7 inputs
	Neurorule	73.05	71.85	6 propositional rules
	Trepan	73.05	71.85	11 leaves, 21 nodes
	Nefclass	68.97	67.24	8 fuzzy rules
C	C4.5	82.80	73.09	438 leaves, 578 nodes
	C4.5rules	77.76	73.51	27 propositional rules
	Pruned NN	74.15	74.09	7 inputs
	Neurorule	74.27	74.13	7 propositional rules
	Trepan	74.15	74.01	9 leaves, 17 nodes
	Nefclass	70.06	69.80	4 fuzzy rules

Table 3 Accuracy and Complexity of Decision Trees, Neural networks, and Extraction Techniques [2]

Data set	Method	Fid _{train}	Fid _{test}
A	Neurorule	99.70	98.80
	Trepan	94.07	93.11
B	Neurorule	100	100
	Trepan	100	100
C	Neurorule	99.71	99.79
	Trepan	99.71	99.83

Table 4 Fidelity rates of extraction techniques [2]

Baesens et al [2] also compared the results of Table 3 with a logistic regression classifier, which obtained classification accuracies of 70.66%, 70.51% and 73.09% for the three data sets respectively. Neurorule has significantly better scores. Even though the difference might be small, on an absolute scale it can be significant. Henley and Hand [15] Say it this way: “Although the differences are small, they may be large enough to have commercial implications.”

Neurorule and Trepan have high classification accuracies, whereas Nefclass performs rather poor compared to those two. Both former methods have high fidelity rates (Table 4), which means they mimic well the decision process of the neural network.

These two methods also yield a very good classification accuracy when compared to C4.5 and logistic regression, both commonly used methods in credit scoring. Furthermore, both methods were able to extract very compact rule sets and trees for all the data sets.

A disadvantage of Trepan is that it is not very comprehensible, whereas Neurorule brings forth propositional rules that are easy to interpret and understand. Even though these rules are easy to interpret, though, they are not necessarily the most suitable representation when it comes to structure and efficiency of use for people who need them on an everyday basis for decision-making.

Since these methods have high accuracies and they are easier to interpret and understand than neural networks, Baesens et al [2] want to deal with the fact that the outcomes

of the models are not yet practical to deal with. They propose visualizing the extracted rule sets using decision tables, for which we refer to [2]. These decision tables are a convenient way to make the results of the classification easy to verify for the human expert and also easy to use in daily business.

8.2 Support Vector Machines

Recently research showed that statistical and machine learning models are not that different after all. In this perspective, Support Vector Machines (SVM) have attracted most interest in the last few years.

SVM is a new learning machine that was first introduced by Vapnik [30]. It is based on the *Structural Risk Minimization* principle from computational learning theory. Hearst et al [13] say the following: “it contains a large class of neural nets, radial basis function (RBF) nets, and polynomial classifiers as special cases. Yet it is simple enough to be analyzed mathematically, because it can be shown to correspond to a linear method in a high-dimensional feature space nonlinearly related to input space.” Seen this way, SVM’s could combine the strengths of more theory-driven and easily analysed conventional statistical methods and more data-driven, distribution-free and robust machine learning methods [33]. Much research has been done in recent years, and much progress has been made. SVM is used for a wide range of problems including bioinformatics, text categorization, image detection, but also for financial problems like the areas of time series prediction and classification and bankruptcy predictors. SVM is said to outperform other classifiers, including even neural networks and linear discriminant classifier. A compact explanation of support vector machines will be given here, for more detailed information please refer to [4] and [21].

In SVM, the classification problem can be represented as an optimization problem. Vapnik [30] shows that for pattern recognition training a SVM leads to a quadratic optimization problem with bound constraints and one linear equality constraint (2).

maximize

$$W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j \langle \phi(x_i), \phi(x_j) \rangle = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j K(x_i, x_j)$$

subject to

$$\sum_{i=1}^l y_i \alpha_i = 0$$

$$\alpha_i > 0$$

$$i = 1, \dots, l$$

(2)

Where a kernel function⁴ $K(x_i, x_j)$ is applied to allow all necessary computations to be performed directly in the input space. Conceptually, kernel functions map the original data, which usually is not linearly separable, into a higher-dimensional space and thereby make the input data set linearly separable in the transformed space (Figure 3). Choosing a kernel

⁴ A kernel function $K(x_i, x_j)$ is a function of the inner product between x_i and x_j , so it transforms the computation of the inner product $\langle \phi(x_i), \phi(x_j) \rangle$ to that of $\langle x_i, x_j \rangle$.

function is highly application-dependent and it is the most important factor in support vector machines.

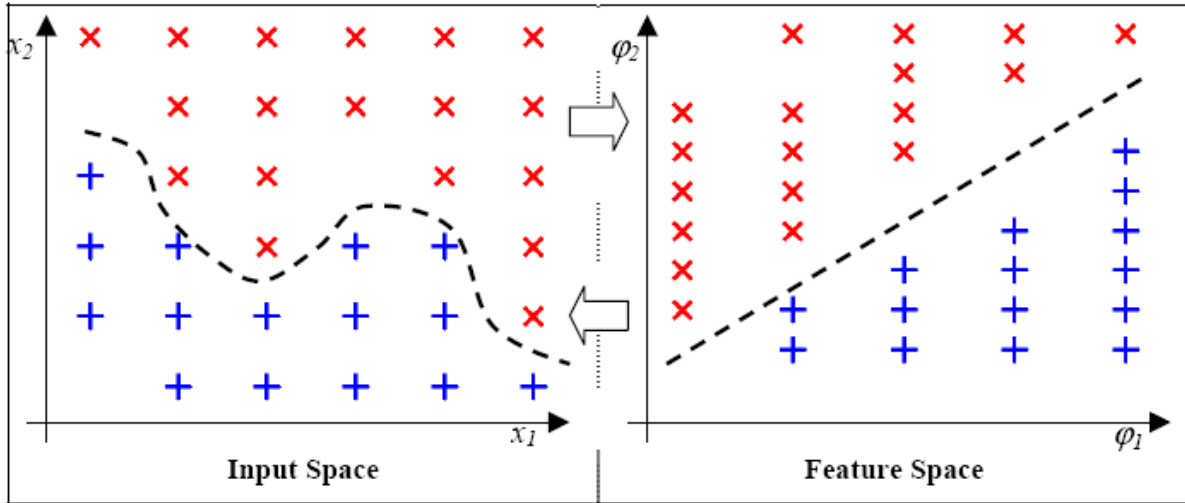


Figure 3 Example of mapping nonlinearly separable input into feature space where it becomes linearly separable [8]

We understand this kind of problem well, but because the number of training examples determines the size of the problem, using standard quadratic problem solvers will easily make the problem computationally impossible. Solutions given by Platt [23] include gradient ascend methods, chunking and decomposition and Platt's Sequential Minimal Optimization algorithm.

The formulation in Equation 1 is so to say the ideal situation, where the data is separable and hence the empirical error zero. In case of noisy data, slack variables can be introduced to relax the constraints ((3).

maximize

$$W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j \langle \phi(x_i), \phi(x_j) \rangle = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j K(x_i, x_j)$$

subject to

$$\sum_{i=1}^l y_i \alpha_i = 0$$

$$C \alpha_i \geq 0$$

$$i = 1, \dots, l$$

(3)

In their study Zan Huang et al [33] compared logistic regression, support vector machines and neural networks. They conducted both 10-fold crossvalidation and leave-one-out-crossvalidation on 4 different data sets. Table 5, as given in their paper, gives the results.

	10-fold crossvalidation			Leave-one-out crossvalidation		
	LogR (%)	SVM (%)	NN (%)	LogR (%)	SVM (%)	NN (%)
I	72.97	79.73	75.68	75.68	79.73	74.32
II	70.27	77.03	75.68	70.27	75.68	74.32
III	76.98	78.87	80.00	75.09	80.38	80.75
IV	75.47	80.00	79.25	75.47	80.00	75.68

Table 5 Prediction accuracies (LogR: logistic regression model, SVM: support vector machines, NN: neural networks)

From this table it can be deduced that support vector machines significantly outperform the other two methods. SVM's and neural networks both lead to better results than logistic regression does. The prediction accuracies obtained with the different crossvalidation procedures are comparable.

Neural networks have an important drawback, namely that of local minima. The existence of multiple local minima means that results may not be optimal and may not be reproducible. Support vector machines form a solution to this problem. The nonlinear mapping that is conducted is only implicitly defined in terms of a kernel function, also known in applied statistics. Therefore, the SVM's and kernel based algorithms in general can be understood as applying a linear technique in a feature space that is obtained via a nonlinear pre-processing step. The advantage of this, that is so important since it solves the neural network's reproducibility problem, is that the solution comes from a convex optimization problem that yields a unique solution [8].

This method seems very promising, but nevertheless a major drawback can be that it requires expert knowledge for implementation. SVM is quite easily explained mathematically and it is as robust as neural networks, but just like those it is not easily implemented.

8.3 Neuro-fuzzy systems

Fuzzy logic is a relatively new rule-based development in artificial intelligence. It is based on the way the human brain deals with imprecise information, whereas neural networks are based on the physical architecture of the brain. Fuzzy logic systems tolerate imprecision and can even use it to solve problems not solved before. Fuzzy logic systems can deal with issues like reasoning on a higher level than neural networks, but their disadvantage is that it does not have much learning capability. This makes it hard for the operator to tune the fuzzy rules and membership functions from the training data set.

Neuro-fuzzy systems address some of the shortcomings of both of these artificial intelligence techniques. They have the best of both worlds and merge neural networks and fuzzy logic into an integrated system.

Fuzzy systems relate classes with non-sharp boundaries and here membership is a measure of degree. The degree to which an object belongs to a fuzzy set is denoted by a membership function (a function that describes how points in an input space are mapped to a membership value, usually between 0 and 1) and fuzzy sets admit partial memberships. They use rules, which are basically IF-THEN statements. The input for the systems is information about the loan applicant and the output is whether or not this applicant is found to be creditworthy.

There are five steps in a fuzzy inference system [19]:

1. Fuzzify the input. In this step the degree to which each of the inputs belong to each of the appropriate fuzzy sets via membership functions is determined. The input values

are always crisp numerical independent variables and these measure the operation of the fuzzy system.

2. Apply fuzzy indicator. Here those crisp variables are fuzzified via a membership function. This function computes the measure of belief in the fuzzy indicator.
3. Apply implication method. The system combines the fuzzy indicators in ways described by the fuzzy rules in the system
4. Aggregate all outputs. Each fuzzy rule now produces an output value that indicates the measure of certainty in the inferred value.
5. Defuzzify the output. For all rules that produce outputs that relate to the same measure, the centroid of the clipped, fuzzy-membership functions is determined and used to defuzzify the output. This process is done for all outputs generated by the system.

For a more thorough exposition about fuzzy logic the reader is referred to [19].

These inference systems can be used for modelling situations where the membership functions are known in advance and where the users might predetermine the rule structure. When this is not the case, when a predetermined model is not available and when we cannot discern a pattern, neural networks come to aid. They can help fuzzy logic systems learn information about the data set and compute the membership functions, in stead of choosing the parameters associated with the membership function arbitrarily. ANFIS – adaptive network-based fuzzy inference system / adaptive neuro-fuzzy inference system- is a type of adaptive network that is functionally equivalent to fuzzy inference systems. It uses a given input-output data set and constructs a fuzzy inference system whose membership function parameters are tuned using a hybrid-learning algorithm (which is a combination of backpropagation and the least squares method) or a backpropagation algorithm. Using ANFIS fuzzy systems are able to learn.

ANFIS is a network-type structure that is similar to that of a neural network. To interpret the input-output map, it maps inputs through input membership functions and associated parameters, and after that through output membership functions and associated parameters to output. For more details, see [19].

In their paper Malhotra et al [19] come to the conclusion that ANFIS has a performance that is comparable to that of multiple discriminant analysis regarding identifying good loans. It performs statistically significantly better on identifying bad loans, and with that on minimizing Type I errors⁵. This error is more important than Type II errors in consumer credit modelling because it reduces the costs of misclassification. Moreover, ANFIS is a better technique, because it does not require the restrictive assumption of multivariate normality of the distribution and unknown but equal dispersion and covariance structures for the groups where MDA does require this. Because of the easily interpretable IF-THEN rules neuro-fuzzy methods use it is easy to clarify why a loan was accepted or rejected. This opposed to the black box problem neural networks present. Furthermore, it is possible to adapt the model by embedding extra rules in the model if needed according to the expertise of the loan officer. Therefore they say that neuro-fuzzy models are promising methods for consumer credit risk modelling.

There are on the other hand some clear limitations to this method. For example, there is no formal theory about how to determine an optimal network topology. Furthermore, training a neuro-fuzzy network can be very computationally intensive.

⁵ Type I errors are those instances that are bad but are misclassified as good. Type II errors are the opposite: good instances that are misclassified as bad. Type I errors yield high costs, since this applicant is most likely to default. Whereas Type I errors also yield costs, namely those of missed opportunities, but they are not as big and therefore do not pose the biggest risk.

8.4 Hybrid Neural Discriminant Technique

This method aims to overcome the drawback of neural networks that it takes a long time training the network in designing the optimal network's topology. Tian-Shyug Lee et al [29] propose a two-stage hybrid modelling procedure in integrating the linear discriminant approach with neural networks techniques. This means that they first use LDA in modelling the credit scoring problem and then the significant predictor variables are served as the input variables of the designed neural networks model. Then, the credit scoring result of discriminant analysis is also included in the input layer as extra information trying to give a better initial solution and increasing the credit scoring accuracy. This method provides a better initial solution and therefore it converges much faster than the conventional neural network model. Moreover, using discriminant analysis as a supporting tool for designing the topology of neural networks is valuable because one can learn more of their inner workings. Tabel 6 shows the results of their experiment.

Credit scoring models	[1-1] ⁶	[2-2] ⁷	Average correct classification rate
Discriminant analysis	74.57	68.29	71.40
Logistic regression	76.19	70.76	73.45
Backpropagation NN	85.27	62.34	73.70
Hybrid neural discriminant model	81.53	72.55	77.00

Tabel 6 Classification results

This too is a promising method in that it improves on original neural network models, but it does however still deal with the same disadvantages as neural networks do.

8.5 Relative importance of input variables

This part is not so much a new method based on neural networks, as it is merely a way to improve neural nets. To find a network that works as well as possible, and also for later study of the outcomes, it would be very convenient to know what contribution the input variables have with in a neural network. In an attempt to clarify this contribution, Garson [7] proposes a contribution measure to evaluate the relative importance of input variables (Equation 4). Yoon et al [32] propose the measure in (5). Both methods are based on a typical three-layer backpropagation neural network. We consider a neural network with I input units, J hidden units and K output units. The connections between input, hidden and output layers are denoted by w_{ij} and v_{jk} , where $i=1, \dots, I, j=1, \dots, J$ and $k=1, \dots, K$.

$$Con_{ik} = \frac{\sum_{j=1}^J \frac{|w_{ji}| |v_{jk}|}{\sum_{i=1}^I |w_{ji}|}}{\sum_{i=1}^I \sum_{j=1}^J \frac{|w_{ji}| |v_{jk}|}{\sum_{i=1}^I |w_{ji}|}} \quad (4)$$

⁶ Good credit applications classified as good.

⁷ Bad credit application classified as bad.

$$Con_{ik} = \frac{\sum_{j=1}^J w_{ji} v_{jk}}{\sum_{i=1}^I \left| \sum_{j=1}^j w_{ji} v_{jk} \right|} \quad (5)$$

Both methods measure the relative contribution of each input variable to each of the output variable. If there is quite a large number of classes towards which the model classifies, Yoon's method becomes very difficult to interpret and it does not improve the understanding of the model. The results from the contribution analysis with Garson's method on the other hand, do allow understanding the relative importance of the input variables.

9 Conclusion

This paper gives an overview of traditional and more recent methods on the topic of credit scoring. Statistical methods seem limited and therefore much research has been done to find methods that are more accurate. Neural network models have received a lot of attention, but these methods suffer from the disadvantage of a lack of explanation of their outcomes.

Support Vector Machines are a fairly new development and research showed that it has high classification accuracies and besides that it is not too hard to explain them mathematically. They also have the advantage that neural nets have too that they are robust, which is important if population drift is an issue. Then again, they do not provide a solution to the lack of explicability of neural network models.

Rule-extraction techniques explicitly deal with opening the 'black box' of neural nets. Their classification accuracy is just as high or at least comes very close, but the rules the extract from the models are easy to interpret and easy to use in daily business. They do however need more time and effort to be implemented.

Hybrid models can come in helpful, as they shorten the time it takes to train a neural network. Then again they are not a solution to the limitations of neural network models, so best would be to investigate what opportunities they can bring in combination with other methods.

For faster convergence of neural network models, the relative importance of input variables can be computed with the formulas the respective chapter. Like with hybrid models, they are not so much a solution themselves, so again it would be interesting to see in what way they can improve other methods when they are included.

As a final word, the choice of a method and the use of a technique depend on the complexity of the institution, the size and the type of the loan. For a small institution with small loans that yield relative small risks, it is not worth it to invest a lot of time and money in implementing that newest, state-of-the-art method. Then again for an institution that depends on those risks, it is of vital importance to use a method that predicts as well as possible.

As far as known, no studies that compare the methods of chapter 8 are available. An interesting topic for further research would be comparing these, and possibly other, methods on their classification accuracies and applicability. Also further investigation of combinations of the proposed methods could lead to interesting and promising results.

10 Bibliography

1. *Baesens B.*, Developing Intelligent Systems for Credit scoring using Machine Learning Techniques, PhD thesis, Katholieke Universiteit Leuven. 2003.
2. *Baesens B., Setiono R., Mues C., Vanthienen J.* Using Neural Network Rule Extraction and Decision Tables for Credit Risk Evaluation, Management Science. 2003.
3. *Cortes C., Vapnik V.N.* Support Vector networks, Machine Learning 20 273-297. 1995.
4. *Cristianini N., Shawe-Taylor J.* An introduction to Support Vector Machines, Cambridge University Press. 2000.
5. *Dillon W.R., Goldstein M.* Multivariate analysis methods and applications, John Wiley & Sons, Inc. New York. 1984.
6. *Enas G.G., Choi S.C.* Choice of the smoothing parameter and efficiency of k -nearest neighbour classification. 1986.
7. *Garson D.* Interpreting neural network connection strengths, AI Expert 47-51. 1991.
8. *Gestel van T., Baesens B. Garcia J., Dijcke van P.* A Support Vector Machine Approach to Credit Scoring, Bank en Financiewezen. 2003.
9. *Grablowsky B.J., Talley W.K.* Probit and discriminant functions for classifying credit applicants; a comparison, Journal of Economics and Business. 1981.
10. *Hand D.J., Henley W.E.* Statistical Classification Methods in Consumer Credit Scoring: a Review, Journal of the Royal Statistical Society. Series A (Statistics in Society), Vol. 160, No. 3, pp. 523-541. 1997.
11. *Hand D.J., Vinciotty V.* Choosing k for Two-Class Nearest Neighbour Classifiers with Unbalanced Classes, Pattern Recognition Letters 24, 1555-1562. 2003.
12. *Harrel F.E., Lee K.L.* A comparison of the discrimination of discriminant analysis and logistic regression. 1985.
13. *Hearst M.A., Dumais S.T., Osman E., Platt J., Schölkopf B.* Support Vector Machines, IEEE Intelligent Systems, vol.13 no.4 pp.18-28. 1998.
14. *Henley W.E., Hand D.J.* A k -nearest Neighbour Classifier for Assessing Consumer Credit Risk, The Statistician. 1996.
15. *Henley W.E., Hand D.J.* Construction of a k -nearest neighbour credit scoring system, IMA Journal of Management Mathematics, 8:305-321. 1997.
16. *Holmes C.C., Adams, N.M.* A Probabilistic Nearest Neighbour Method for Statistical Pattern Recognition, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64(2) 295-306. 2002.
17. *Johnson R.A., Wichern D.W.* Applied multivariate statistical analysis, Prentice Hall. 1998.
18. *Kin Keung L., Lean Y., Shouyang W., Ligang Z.* Credit Risk Analysis Using a Reliability-Based Neural Network Ensemble Model. 2006.
19. *Malhotra R., Malhotra D.K.* Differentiating between good credits and bad credits using neuro-fuzzy systems, European Journal of Operational Research, vol.136 no.1 pp.190-211(22). 1999.
20. *Malhotra R., Malhotra D.K.* Evaluating consumer loans using neural networks, EFMA 2002 London Meetings. 2001.
21. *Müller K.R., Mika S., Ratsch G., Tsuda K., Schölkopf B.* An introduction to kernel-based learning algorithms, Neural Networks IEEE Transactions on vol.12(2) pp.181-201. 2001.
22. *Nauck D.* Data analysis with neuro-fuzzy methods, International ISSEK Workshop on Data Fusion and Perception, Udine. 2000.

23. *Platt J.C.* Fast training of support vector machines using sequential minimum optimization, in: Schölkopf B., Burges C., Smola A. (Eds), *Advances in Kernel Methods-Support Vector Learning*, pp 185-208. 1999.
24. *Reichert C.C., Cho, Wagner G.M.* An examination of the conceptual issues involved in developing credit scoring models, *Journal of Business & Economic Statistics* vol.1 no.2 pp. 101-114. 1983.
25. *Sanchez M.S., Sarabia L.A.* Efficiency of multi-layered feedforward neural networks on classification in relation to linear discriminant analysis, quadratic discriminant analysis and regularized discriminant analysis. 1995.
26. *Sharma S.*, *Applied multivariate techniques*, John Wiley & Sons, Inc. New York. 1996.
27. *Thomas L.C.* A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers, *International Journal of Forecasting*, 16:149-172. 2000.
28. *Thomas L.C., Oliver R.W., Hand D.J.* A Survey of issues in Consumer Credit Modelling Research. *Journal of the Operational Research Society*, 1006-1015. 2005.
29. *Tian-Shyug Lee, Chih-Chou Chiu, Chi-Jie Lu, I-Fei Chen.* Credit scoring using the hybrid neural discriminant technique, *Expert Systems with Applications* vol. 23(3) pp. 245-254. 2002.
30. *Vapnik V.* *The Nature of Statistical Learning Theory*, Springer Hong Kong. 2000.
31. *Vojtek M., Kocenda E.* Credit scoring methods. DATUM.
32. *Yoon Y., Guimares T., Swales G.* Integrating artificial neural networks with rule-based expert systems, *Decision Support Systems* vol. 11(5) pp. 497-507. 1994.
33. *Zan H., Hsinchun C., Chia-Jung H., Wun-Hwa C., Soushan W.* Credit rating analysis with Support Vector Machines and Neural Networks: a market comparative study, *Decision Support Systems*. 2004.