Support vector machines in ordinal classification

A revision of the ABN AMRO corporate credit rating system

Harmen J. Dikkers

August 2005

TU Delft
Delft University of Technology

ABN·AMRO

Support vector machines in ordinal classification

A revision of the ABN AMRO corporate credit rating system

THESIS

in partial fulfilment of the requirements for the degree of
Master of Science
presented at Delft University of Technology,
Faculty of Electrical Engineering, Mathematics, and Computer Science,
Department of Man-Machine Interaction

by
Harmen J. Dikkers
August 2005

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands
E-mail: dikkers@gmail.com
Student number: 9183257

Research was done at ABN AMRO
Group Risk Management
Quantitative Consultancy
Gustav Mahlerlaan 10
PO Box 283 – PAC HQ9060
1000 EA Amsterdam

Members of the Supervising Committee:
drs.dr. L.J.M. Rothkrantz
dr.ir C.A.P.G. van der Mast
dr. K. van der Meer
dr. D.R. Fokkema (ABN AMRO)

Keywords: artificial intelligence, data mining, credit rating analysis, bond rating prediction, ordinal classification, support vector machines, least squares support vector machines, Basel 2

Abstract

Risk assessment of credit portfolios is of pivotal importance in the banking industry. The bank that has the most accurate view of its credit risk will be the most profitable. One of the main pillars in assessing credit risk is the estimated *probability of default* of each counterparty, i.e., the probability that a counterparty cannot meet its payment obligations in the horizon of one year. A *credit rating system* takes several characteristics of a counterparty as inputs, and assigns this counterparty to a rating class. In essence, this system is a classifier whose classes lie on an ordinal scale.

This thesis provides an extensive assessment of the ABN AMRO credit rating system. The current rating tool, an expert system, is carefully reviewed. We show that this system has several drawbacks in both its mathematical fundamentals and its implementation. We propose a new credit rating framework, which incorporates an improved version of the current model.

Aside from this expert system, we applied linear regression, ordinal logistic regression, and support vector machine techniques to the credit rating problem. The latter technique is a relatively new machine learning technique that was originally designed for the two-class problem. We propose two new techniques that incorporate the ordinal character of the credit rating problem into support vector machines. We show that the current rating model used at ABN AMRO performs in line with statistical and support vector machine techniques. The results of our newly introduced techniques are promising.

Contents

Acknowledgements

This master's thesis is the result of my graduation project at the Delft University of Technology, at the Man-Machine Interaction group. Most of the work has been performed at ABN AMRO. Dr. Diederik Fokkema offered me the opportunity to work on my graduation project at the quants' department of ABN AMRO: Quantitative Consultancy. I have really enjoyed to be a part of this inspiring group, and would like to thank him for giving me this opportunity and for being my supervisor. I had never expected that a manager at ABN AMRO could give me so many useful comments on object-oriented programming, C++, and nearly anything I did.

Meetings with my other supervisor, dr.drs. Leon Rothkrantz, were always both constructive and enjoyable. Not only did he give me many useful suggestions, there was always some time for anecdotes on the university and study association Christiaan Huygens.

It has been an extreme struggle to finally get the data after more than nine months. I would like to thank the team that joined me in the fight for data: Iwan Meister, Siemone Griffioen, Roel van Leyen, Huib Ypma, and Michael Voorn.

Invaluable comments on my report and employed techniques have come from Marijn, Joris, Adriaan, and David Tax. My girl-friend Ariënne has given me the mental strength during the past months and would bring me coffee at three o' clock in the morning during deadlines. Last but not least my colleagues at ABN AMRO and my friends at the university deserve to be mentioned in this section.

Harmen Dikkers
July 29, 2005

Notation

Boldface variables like $\mathbf{x}$ and $\mathbf{A}$ represent vectors and matrices that consist of elements x_i and A_{ij} respectively.

$\|\cdot\|$	2-norm of a vector
$\boldsymbol{\alpha}$	Lagrange multiplier vector
b	Bias
c	Number of rating classes
C	Regularisation constant
Cov	Covariance
$E(\cdot\|\cdot)$	Expectation
ϵ	Residual
$\mathcal{F}$	Function class
$\boldsymbol{\varphi}(\cdot)$	Mapping to feature space
$\mathbf{G}$	Gram matrix with elements $G_{ij} = y_i y_j \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j) = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$
$\mathcal{H}$	Feature space
$\mathbf{I}_i$	Identity matrix of size i
$K(\cdot,\cdot)$	Kernel function
ℓ	Number of training samples
$l(\cdot,\cdot)$	Loss function
$\mathcal{L}(\cdot)$	Lagrange function
μ	Sample mean
$\mathcal{N}(\mu,\sigma^2)$	Normal distribution with mean μ and variance σ^2
$\mathbb{N}$	Set of natural numbers, i.e., all integers
ω_i	Rating class, $i = 1,\ldots,c$
$\boldsymbol{\nu}$	Lagrange multiplier vector
$P(\cdot), P(\cdot,\cdot)$	Probability density function
π_i	Prior probability of class ω_i, $i = 1,\ldots,c$
Q_1, Q_2, Q_3	75^{th}, 50^{th} and 25^{th} percentile value respectively
r	Pearson's correlation coefficient
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of positive real numbers
S	Set of ℓ patterns $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}$
$S(\cdot)$	Soft saturation function
σ	Sample standard deviation, or bandwidth of the RBF kernel function
Σ	Covariance matrix
sign$[i]$	Sign function: +1 if $i > 0$, 0 if $i = 0$, and –1 if $i < 0$
$.^T$	Transpose of a vector or matrix
$\mathcal{U}(a,b)$	Uniform distribution on the interval $[a,b]$
$v_i(\cdot)$	Voting function
$\mathbf{w}$	Weight vector
$\mathbf{x} \in \mathcal{X}$	Input vector and input space
$\boldsymbol{\xi}$	Slack variable vector
X_i, Y	Variates with corresponding probability distribution functions $P(x_i)$ and $P(y)$
$y \in \mathcal{Y}$	Output scalar and output space
z	Latent variable

Definitions

ANOVA	Analysis of variance
BPNN	Backpropagation neural network
Capital structure	The use of debt financing
Default	The failure to meet a financial obligation
EC	Economic capital
EAD	Exposure at default
EL	Expected loss
Exposure	The maximum loss suffered from a default by a counterparty
GRACE	Generic Rating ABN AMRO Counterparty Engine; a wrapper system for all rating tools of ABN AMRO (except MRA)
IRD	Internal Ratings Database; the historical database that contains all counterparties and ratings
LGD	Loss given default
Leverage	*see* capital structure
LINREG	(Ordinary least squares) linear regression
LOGREG	(Ordinal) logistic regression
LS-SVM	Least-squares support vector machines
MDA	Multiple discriminant analysis
MFA	Moody's Financial Analyst[1]; the present spreading tool
MRA	Moody's KMV Risk Advisor[1]; the present rating tool
Notch	Distance between two subsequent rating classes
PCC	Percentage correctly classified
PCC-1	Percentage correctly classified within one notch
PD	Probability of default
QC-MRA	Quantitative Consultancy's MRA
RAPID	ABN AMRO's credit proposal system
RBF	Radial basis function
SVM	Support vector machine
UCR	Uniform Counterparty Rating
VBA	Visual Basic for Applications
Volatility	A measure of the uncertainty or risk in the future price of an asset.

[1] Trademark of Moody's KMV

Chapter 1

Introduction

A large corporation that wants to apply for a loan, obviously shops around to negotiate the best possible price and terms. The provider of such a loan, often a bank, wants to make a decent profit from this loan. Basically, this profit consists of two parts. Interest payments form the basis of the profit. Even more important, however, is whether the corporation will eventually be able to pay off the loan itself. Not surprisingly, these two aspects are tightly coupled. The larger the risk that a corporation might not be able to meet its payment obligations in the future, the more interest it needs to pay.

A bank therefore assesses the risk it takes when providing a loan to a corporation (or any other customer). If the bank overestimates this risk, the price of the loan will be too high compared to competitor banks, and the corporation will apply for a loan elsewhere. On the other hand, if the bank underestimates the risk, the bank will issue loans to dubious debtors for a price that is too low.

Careful risk assessment of credit portfolios is therefore of pivotal importance in the banking industry. The bank that has the most accurate view of its credit risk will be the most profitable. Moreover, since banks are the cornerstones of a country's economy, inaccurate credit risk assessment can have a tremendous impact.

1.1 Minimum capital requirements

In the 1970s, as banking grew more global, institutions looked for a common way to manage risk across countries. A committee of bank regulators, among which the Dutch Central Bank (DNB), created a treaty that would encourage management, control and regulation of banking systems: the 1988 Basel Capital Accord (Basel-I). All Dutch banks are bound by this treaty.

Basel-I sets the minimum capital requirements for banks. It requires banks to divide their exposures up into broad classes reflecting similar types of borrowers. Exposures to the same kind of borrowers are subject to the same capital requirements. Ordinary corporate loans for instance pose 100% risk to the bank, whereas mortgage loans only account for 50%.

Basel-I requires banks to keep their capital ratios above 8%, where the capital ratio is defined as

$$\text{Capital ratio} = \frac{\text{Total capital}}{\text{Credit risk}} = \frac{\text{Total capital}}{\sum_i \text{Risk}_i \times \text{Exposure}_i} \tag{1.1}$$

A bank that only has exposures to corporate borrowers, for instance, will thus be allowed to issue loans for a maximum of 12.5 times its total capital.

The Basel-I framework does not distinguish between the relative degrees of creditworthiness among individual borrowers. A loan issued to for instance a AAA-rated[1] company poses the same

[1] The highest possible rating on Standard and Poor's scale of credit rating. Appendix A.3 describes different credit rating scales.

amount of risk to the bank as an immature C-rated company, whereas the underlying actual risk is clearly different.

Aside from this aspect, many other extensions to the Basel-I framework were requested throughout the financial world. This resulted in the development of a second version of the accord: the New Basel Capital Accord.

1.2 The New Basel Capital Accord

The New Basel Capital Accord (Basel-II) defines a refined and renewed set of rules regarding proper banking practice (BIS 2004). One of its goals is to align the Basel-I guidelines more closely to each bank's actual risk of economic loss. Aside from credit risk two other types of risk have been introduced: market risk and operational risk.

$$\text{Capital ratio} = \frac{\text{Total capital}}{\text{Credit risk} + \text{Market risk} + \text{Operational risk}}$$

Market risk and operational risk have been left out of the scope of this thesis. More information on these topics can for instance be found in BIS (2004). The calculation of credit risk is similar to formula 1.1, but is now directly related to the creditworthiness of the borrower. Basel-II describes, aside from this standardised approach to measure credit risk, the Internal Risk Based (IRB) approach. According to the IRB approach, banks will be allowed to use their own measures of a borrower's credit risk to determine their capital requirements, subject to strict methodological and disclosure standards.

Basel-II prescribes that credit risk should directly be related to *expected loss* (EL). The expected loss of a borrower is calculated by

$$\text{EL} = \text{PD} \cdot \text{EAD} \cdot \text{LGD}$$

where the *probability of default* (PD) is the probability that a borrower will default in the horizon of one year, *exposure at default* (EAD) is the expected total exposure or outstanding loans to the borrower at the time of default; and *loss given default* (LGD) is the expected percentage of the exposure which the bank will be unable to recover.

Basel-II does not prescribe a calculation method for either of these elements. Therefore, banks are permitted to use their own best practice methods.

1.3 Economic capital

ABN AMRO prefers an even more sophisticated and safer methodology than prescribed by Basel-II. The bank wants to quantify the probability that it defaults itself. Moreover, ABN AMRO wishes to introduce risk-adjusted rates on loans. Loan pricing (i.e., the interest a borrower pays) will depend on expected loss per borrower and worst-case expected loss among all borrowers. A final important aspect is the correlation among defaults: if a company defaults, it might drag other companies into default as well, which on their turn can cause other companies to default. This way a vicious circle is created.

Therefore, the focus has shifted to *economic capital* (EC) rather than capital ratio. Economic capital is a percentile (e.g., 99.95%) of the loss distribution curve, which is generated using a Monte Carlo simulation. This percentile is the amount of capital the bank needs to cover its unexpected losses in the best 9995 out of 10,000 possible scenarios. The bank will only default in the five other scenarios, or, statistically, once every 2000 years. More information on Monto Carlo simulations for economic capital calculations can be found in Caouette et al. (1998).

Figure 1.1 shows a typical loss distribution curve. In this example, the (mean) expected loss is 1% of its total exposure. ABN AMRO wants to be able to cover unexpected loss in 99.95% of all scenarios. The 99.95^{th} percentile corresponds to an economic capital of approximately 2.8% in this example, which means that the bank's capital should be 2.8% of its total exposure. An

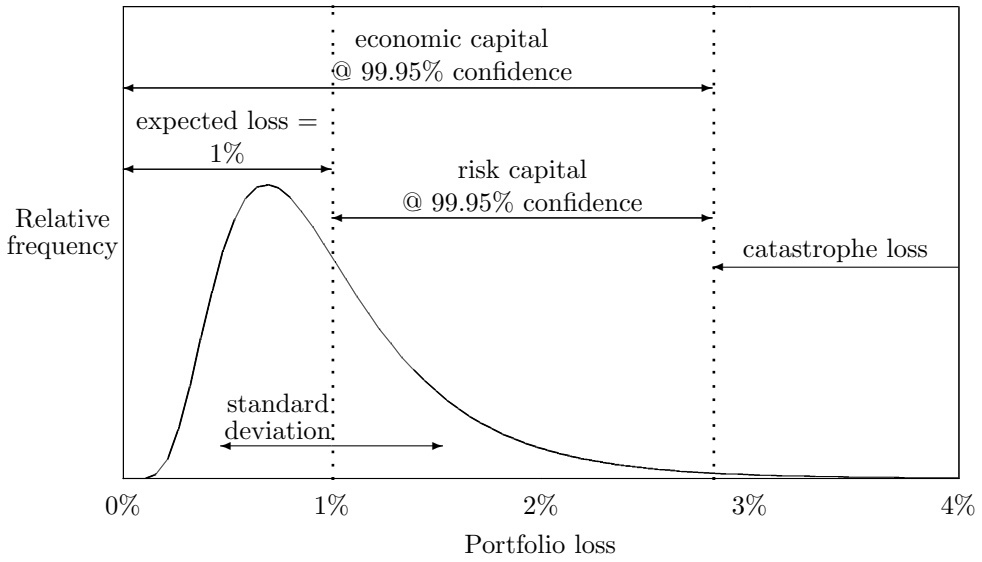


Figure 1.1 — Economic capital

economic capital of 99.95% corresponds to a AA+ company according to Standard and Poor's (cf. appendix A.3).

To calculate and manage the economic capital, we need to know the three elements of expected loss PD, EAD, and LGD, and the correlations among them. One model for each of these elements does not suffice, as distinct types of borrowers show very different loss characteristics. Therefore, separate models exist for each borrower type, such as models for insurance companies, banks, non-financial corporate clients, and consumers.

1.4 Uniform counterparty rating

In credit rating it is common to assign a rating, or risk bucket, to a counterparty instead of formulating a PD directly. Counterparties of the same rating are supposed to be of equal risk. Within ABN AMRO the ratings are called Uniform Counterparty Ratings, or UCRs. These range from the highest rating 1, 2+, 2, 2–, 3+ all the way to 6+[2]. UCRs 6, 7, and 8 indicate forms of financial distress. A comparison of UCRs and credit ratings from Moody's and Standard and Poor's can be found in appendix A.3.

The most important aspect of the UCRs is that they are ordered according to their corresponding probability of default, but that the actual value of the PD is irrelevant for this ranking. The credit rating problem hence reduces to finding a function $f(\cdot)$ that maps a counterparty that is described by certain characteristics to a class on the ordinal scale:

$$f(\cdot) : \mathcal{X} \subseteq \mathbb{R}^n \longrightarrow \mathcal{Y} \subseteq \{1, \ldots, c\},$$

where $\mathcal{X}$ is the input space of n features, and $\mathcal{Y}$ is the ordinal output of c rating classes in total. The ordinality implies that we expect the classes to be ordered, as is depicted in figure 1.2.

Although we are working with ordered UCRs, there is still an underlying distribution of probabilities of default. We can assign an expected PD to a UCR class, or calculate this value based on historical defaults. The default estimates for the different PDs have not been listed in this report for confidentiality reasons.

[2]Note that 2+ is a higher rating class than 2–, which might be counter-intuitive.

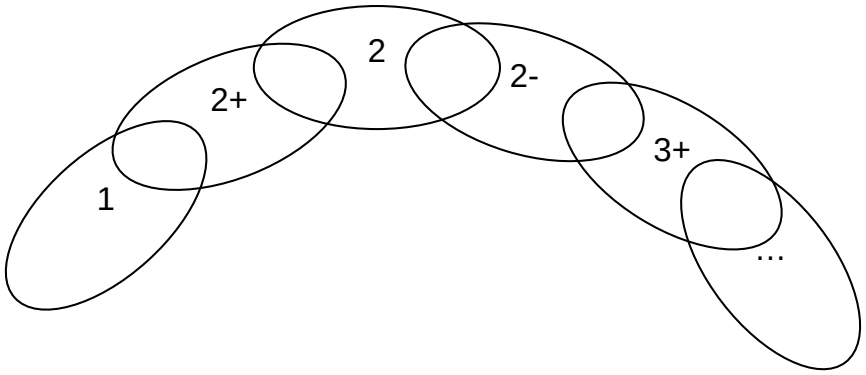


Figure 1.2 — Ordinality among the rating classes

1.5 Credit rating process

We will now describe the credit rating process as depicted in figure 1.3. When a corporate customer (or counterparty) of ABN AMRO applies for a loan, he contacts his account manager. The account manager will then create a credit proposal, which includes an estimate of the counterparty's creditworthiness.

The counterparty provides annual reports including financial statements to ABN AMRO, which are entered into a spreading system named Moody's Financial Analyst (MFA). The account manager now opens the rating system that is currently in place: Moody's Risk Advisor (MRA). Financial ratios are calculated from the statements in MFA and are automatically provided to this rating system. Several subjective questions regarding the counterparty need to be answered. The counterparty's country of residence, main sales areas, and industry have to be provided as well. These three main areas are inputs for the credit rating model.

The credit rating system determines the counterparty's creditworthiness and quantifies it by means of a UCR: the *initial* UCR. The account manager can correct this initial UCR according to his insights and will come to a *proposed* UCR. This concludes his assessment of the customer's creditworthiness. A summary called the *Corporate Rating Sheet* will be created, which is basically a PDF-document stating the key elements of the proposed UCR.

The credit proposal itself has to be created in RAPID, ABN AMRO's credit proposal system. The proposal will contain information about the counterparty, the proposed UCR, and details concerning the requested loan. The Corporate Rating Sheet is attached to this proposal. After a first review by a credit analyst, it will be propagated to the credit risk committee. This committee of credit experts is allowed to make final amendments to the proposed UCR before they approve it. The *approved* UCR is entered into the RAPID system and the account manager receives a notification message. He can now negotiate the terms and conditions of the loan with the customer.

1.6 Problem definition

We have seen that the rating tool is an important part of the credit rating process. It involves both risk management and loan pricing; aspects that are central pillars of the banking industry.

Our research goal is to *assess the performance of the present corporate credit rating model*. 'Corporate' indicates this model is used for the corporate or wholesale portfolio, containing non-financial and non-governmental borrowers that have a turnover exceeding 50 million euros. The wholesale portfolio contains approximately 8,000 counterparties with an exposure ceiling of 269 billion euros. Its main aspect is that all counterparties are bound by similar accounting rules, which makes them mutually comparable.

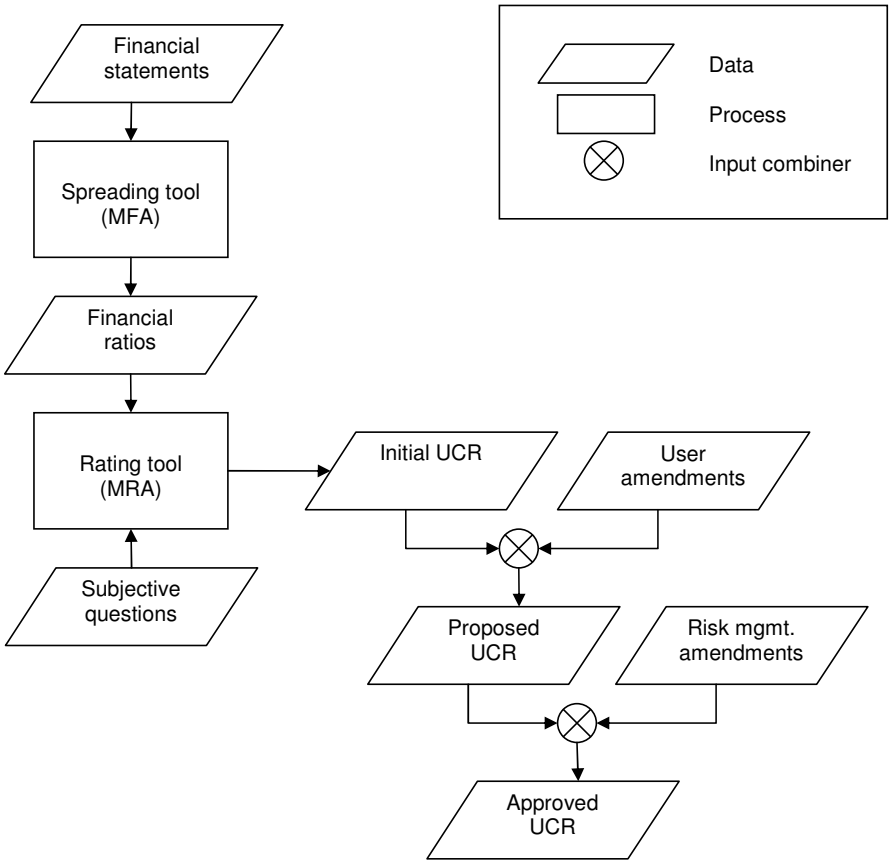


Figure 1.3 — Credit rating process

The research goal will be tackled in a three-tier approach:

- Review of the status quo
- Benchmark the current situation with proven mathematical techniques
- Research state-of-the-art credit rating techniques

First we will thoroughly assess how well the current situation works and what its main shortcomings are. This will show that the model that is currently in place does not meet today's requirements. We will design and implement a new credit rating program, and incorporate an implementation of the present model.

Credit rating data has been recorded starting in January 2004. It has thus become possible to apply statistical approaches to the credit rating problem. A model based on statistics can be maintained more easily since it learns from examples. The quality of prediction will become measurable as well. Research peers have suggested several different techniques to tackle credit rating. A state-of-the-art technique called support vector machines has only recently been applied, but has proven to be very successful. We will therefore benchmark the present rating model with both ordinary statistical techniques and support vector machines.

At first sight one would think that a credit rating tool should be validated based on actual defaults. The number of defaults in the ABN AMRO wholesale portfolio is, however, very low. It is infeasible to design a model based upon this default information. This problem applies to the other players in the financial world as well, and it is therefore common practice to learn from examples based on external agency's ratings or credit committee's opinions (Standard and Poor's

2004). In our project we will benchmark the systems under research against the approved UCR, i.e., the final opinion of ABN AMRO regarding the creditworthiness of a borrower.

Evidently, these models can and must be validated using actual default information. In fact, it is the opinion of the credit committee that will have to be validated, since all developed models will be based on their decisions. Validation belongs to the world of rare event statistics and is outside the scope of this project.

1.7 Project assignment

The general layout of the project is as follows:

1. Literature survey
2. Review of the status quo
   - Gather the data
   - Review the present model
3. New model
   - Develop and implement a new rating system
   - Incorporate techniques used in the present rating system
   - Evaluate this system
4. Mathematical and artificial intelligence techniques
   - Select variables
   - Research possible approaches
   - Implement techniques
5. Results
   - Select performance measures
   - Validate all systems and analyse results
   - Conclusions and recommendations

Chapter 2 describes techniques that are commonly used in credit rating. A survey of the work in which these techniques have actually been applied is presented in chapter 3. Next comes the assessment of the present situation. Chapter 4 gives an extensive description of the data and its characteristics. Based on an analysis of the present tool, a new credit rating framework is proposed in chapter 5, outlining its development, implementation, and evaluation. The goal of chapter 6 is two-fold. First aspect is the revelation of characteristics of the currently used model that have previously been unknown. The second part describes an improved version of the existing model that is incorporated in our framework. Several support vector machines methods are presented next, including two new approaches that take the ordinal character of credit rating into account. A comparison of the results, including that of regression techniques, is presented in chapter 8. We will end this report with our conclusions and recommendations in chapter 9.

Chapter 2

Preliminaries

This chapter discusses techniques suitable to solve the credit rating problem, i.e., find a function $f : \mathcal{X} \subseteq \mathbb{R}^n \longrightarrow \mathcal{Y} \subseteq \{1, \ldots, c\}$ that correctly assigns an unseen pattern to one of c rating classes. We will learn this function from a training set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}$ of ℓ examples. Both the training set and the unseen patterns are assumed to be independently and identically distributed according to the same unknown probability distribution $P(\mathbf{x}, y)$.

The best function that can be obtained is the one minimising the expected risk, or error (Müller et al. 2001),

$$R[f] = \int l(f(\mathbf{x}), y) dP(\mathbf{x}, y),$$

where $l(f(\mathbf{x}), y)$ is a suitably chosen cost function. For the binary problem the 0/1 loss function $l(f(\mathbf{x}), y) = \frac{1}{2}|(f(\mathbf{x}) - y)|$ is an obvious choice. A common loss function for ordinal and regression problems is the squared error: $l(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$.

The risk, however, cannot be minimised directly, due to the fact that the underlying probability distribution $P(\mathbf{x}, y)$ is unknown. Therefore, we will have to estimate a function that is optimal with respect to the available information: the training set and the properties of the function class $\mathcal{F}$ from which the solution f is chosen. In empirical risk minimisation, one tries to find the function $f(\cdot)$ that minimises the average risk on the training set:

$$R_{\text{emp}}[f] = \frac{1}{\ell} \sum_{i=1}^{\ell} l(f(\mathbf{x}_i), y_i)$$

The empirical risk will asymptotically converge to the expected risk when $\ell \to \infty$. However, for smaller ℓ, the training set might give large deviations with respect to the underlying unknown probability distribution. One can therefore not guarantee that minimisation of the empirical risk will result in a minimal expected risk as well, and *overfitting* might occur.

We can avoid overfitting by restricting the complexity of function class $\mathcal{F}$ from which $f(\cdot)$ is chosen. This is also intuitive: we prefer the simplest model that explains the data (Occam's razor). Other techniques involve cross-validation and early stopping.

In the next sections we will discuss the following techniques:

- Ordinary least squares regression
- Logit and probit regression
- Multiple discriminant analysis
- Backpropagation neural networks
- Support vector machines

In the first three techniques, we make assumptions on the distribution of the data to avoid overfitting. Neural networks usually utilise early stopping techniques, whereas overfitting in support vector machines is avoided using structural risk minimisation.

2.1 Ordinary least squares regression

The earliest studies in credit rating focused on the *ordinary least squares regression*. Ordinary least squares regression fits a linear multiple regression (LINREG) model

$$Y = \mathbf{w}^T\mathbf{x} + b + \epsilon,$$

where $\mathbf{x}$ represents a vector of observed characteristics of a counterparty, $\mathbf{w}$ is a vector of coefficients, b is the offset, and ϵ is the residual. The Gauss-Markov theorem states that the ordinary least squares estimators are the best linear unbiased estimators in the class of all linear unbiased estimators of $\mathbf{w}$ and b when a number of assumptions are met. If all assumptions are met, the estimators are unbiased (i.e., expected to be equal to the true value) and efficient (i.e., estimated with smallest variances). We will list the most important ones.

1. The relationship between the dependent and independent variables is linear.
2. The expected value of the residuals equals zero: $E(\epsilon|\mathbf{x}) = 0$. Failure of this assumption results in a biased estimate of the offset.
3. The residuals are homoscedastic: $E(\epsilon^2|\mathbf{x}) = \sigma^2 =$ constant. Homoscedasticity means that for every $\mathbf{x}$, the spread of ϵ around $\mathbf{x}$ will have the same range. Failure of this assumption results in inefficient estimates and biased tests of hypotheses. Outliers are one of the possible causes for heteroscedasticity.
4. There is no correlation between the independent variables and the residuals (no simultaneity): $\mathrm{Cov}(x_i, \epsilon) = 0$. Failure of this assumption results in biased estimates of the coefficients of the independent variables.
5. The independent variables are not perfectly correlated (no multicollinearity): $r_{x_i x_j} \neq \pm 1$, where r is Pearson's correlation coefficient (cf. section 4.4.2). Failure of this assumption results in inefficient estimates and biased tests of hypotheses.

Sometimes stronger assumptions are relied on. The 'no multicollinearity' assumption then requires the explanatory variables to be statistically independent. Second, the residuals are assumed to be normally distributed. Hair et al. (1987) provides further reading on regression.

Ordinality

The ordinal nature of the credit rating problem causes difficulties for the definition of Y. A common solution is to assign $Y = i$ if the rating class is ω_i, where ω_i are the c ordered rating classes and i indicates the rank. Class ω_1 for instance represents AAA, ω_2 AA, etcetera. Reversely, a counterparty is assigned to class ω_i when $i - 0.5 < E(Y|\mathbf{x}) \leq i + 0.5$. If $E(Y|\mathbf{x}) < 0.5$ or $E(Y|\mathbf{x}) > c + 0.5$, it is assigned to ω_1 and ω_c respectively.

When historical probabilities of default are available, these probabilities can serve as independent variables as well. Each class rank is thus replaced by the estimated or historical probability of default associated to its rating class ω_i: $Y = p_i$, where p_i is the PD of class ω_i. Another possibility is to replace the rank with the *logit* of these probabilities: $\mathrm{logit}(p_i) = \log((p_i)/(1-p_i))$. This logit transformation ensures that $E(Y|\mathbf{x})$ remains on the scale $[0, 1]$.

The use of (continuous) linear regression for an ordinal problem might lead to violations of several assumptions. The primary problem is that this model defines Y as an interval scale with equal intervals attached to each rating class, e.g., the risk difference between class AAA and class AA is assumed to be the same as that between BB and B. Linear regression further assumes that both the errors and the dependent variable follow the normal distribution and have constant variance over the complete range. These two assumptions are violated by definition due to the discrete character of the ratings.

2.2 Logit and probit regression

Ordered logit[1] (LOGREG) and *ordered probit* models are considered to be more appropriate for ordinal dependent variables; they allow the dependent variable to be non-continuous. Both models are examples of generalised linear models (GLMs). GLMs assume a *link linear* relationship between the independent and the dependent variables:

$$g(Y) = \mathbf{w}^T\mathbf{x} + b,$$

where $g(Y)$ is the link function. The maximum likelihood estimates of the regression parameters $\mathbf{w}$ and b are calculated using the iteratively reweighted least-squares algorithm.

In the remainder of this subsection we will discuss the GLM with a logit link function. The function used in probit regression is the inverse of the standard normal cumulative distribution function.

LOGREG attempts to model the dependent variable by means of maximum likelihood estimation. Suppose $\mathbf{x}$ is a vector of explanatory variables and $\mathcal{P}_i(\mathbf{x}) = \sum_{j=1}^{i} p(\omega_i|\mathbf{x})$ is the cumulative probability that a counterparty belongs to rating class ω_i or below. The linear logistic model has the form:

$$\text{logit } \mathcal{P}_i(\mathbf{x}) = \log\left(\frac{\mathcal{P}_i(\mathbf{x})}{1-\mathcal{P}_i(\mathbf{x})}\right) = \mathbf{w}^T\mathbf{x} + b_i, \qquad i = 1, \ldots, c-1$$

where b_i is a class-specific intercept and $\mathbf{w}$ is the *class-independent* vector of slope parameters. The observation is then classified into the class with the highest probability $\mathcal{P}_{i+1}(\mathbf{x}) - \mathcal{P}_i(\mathbf{x})$. Compared to MDA and LINREG, LOGREG is less strict in its statistical assumptions:

1. The relationship between the dependent and independent variables is assumed to follow the logistic distribution.
2. The separate equations for each class differ only in their intercepts (parallel slopes or proportional odds).
3. There is no multicollinearity among the independent variables.

2.3 Multiple discriminant analysis

Multiple discriminant analysis (MDA) tries to differentiate between groups by identifying the variables that discriminate most. It is based on the Bayesian method to minimise the expected misclassification costs. A counterparty characterised by vector $\mathbf{x}$ is assigned to class ω_i if

$$\frac{f_i(\mathbf{x})}{f_j(\mathbf{x})} > \frac{\pi_j l(i|j)}{\pi_i l(j|i)}, \qquad \forall i \neq j, i, j = 1, \ldots, c$$

where f_i and f_j are the multivariate probability density functions for class ω_i and ω_j, respectively. π_i and π_j are the prior probabilities of ω_i and ω_j. $l(i|j)$ and $l(j|i)$ are the misclassification costs of assigning a class ω_i counterparty as ω_j and vice versa.

The vast majority of research on MDA has focused on normal density functions for f_i and f_j. This model applies when the counterparties $\mathbf{x}$ of class ω_i are continuous-valued, randomly corrupted versions of a prototype counterparty $\boldsymbol{\mu}_i$. Another key element in normal discriminant functions is the covariance matrix among the independent variables.

If the covariance matrices for all the classes are identical, the discriminant functions are linear. The single covariance matrix can be estimated from the complete set of counterparties. We will refer to this type of functions as *linear discriminant analysis* (LDA).

When the covariance matrices are unequal over the different classes, the discriminant function that maximises the likelihood of correct classification is in a quadratic form: *quadratic discriminant analysis* (QDA). The covariance matrices will have to be estimated based on the available

[1] Ordered logit regression is in result similar to ordered/ordinal logistic regression and proportional odds logistic regression.

counterparties in each class separately. In practice, this method is inefficient for high-dimensional input spaces since it requires the estimation of many parameters.

The assumptions of normal multiple discriminant analysis largely coincide with those of linear regression:

1. The independent variables are multivariate normally distributed: $\mathbf{x} \sim (\boldsymbol{\mu}, \Sigma)$.
2. In LDA, the variance/covariance matrix is homogeneous across all classes: $\Sigma_i = \Sigma_j$. Outliers are a possible cause for violation of this assumption.
3. There is no multicollinearity among the independent variables.
4. The relationship is linear in its independent variables.

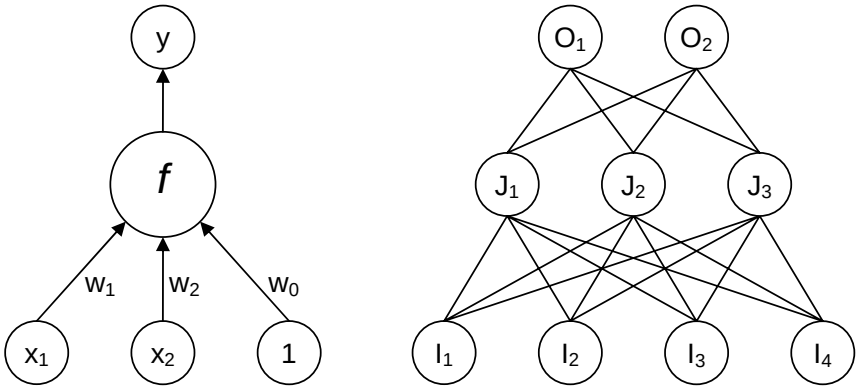
Both Duda et al. (2001) and Webb (2002) extensively discuss discriminant analysis as well as other classification techniques.

2.4 Backpropagation neural networks

A *neural network* is used to map a data set using a recursive mathematical expression. It consists of a set of input nodes and output nodes, which are connected through intermediate-layer nodes named hidden-layer nodes. When the training data is fed into the input layer, it is propagated forward through the layers. The output y_j of node j is derived using the following formula:

$$y_j = f\left(\sum_{i=1}^{k} w_{ji}x_i + w_{j0}\right) = f(\mathbf{w}_j^T\mathbf{x}),$$

where k is the number of input nodes, x_i the value of input node i, and w_{ji} is its corresponding weight. The bias is given by w_{j0}, and $f(\cdot)$ is the *activation function*, which is usually an S-shaped function. Figure 2.1 shows a node and a typical neural network.



(a) *One node*

(b) *Typical neural network with one hidden layer and two outputs*

Figure 2.1 — Neural network

The final outputs that come from the output layers are compared with the desired values by means of squared errors:

$$\mathcal{J}(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(d_i - o_i)^2 = \frac{1}{2}\|\mathbf{d} - \mathbf{o}\|^2,$$

where n is the number of output nodes, o_i is the computed output, and d_i the desired output. The learning in a neural network is performed by minimising $\mathcal{J}(\mathbf{w})$. The optimal weights $\mathbf{w}$ are found

by means of the *gradient descent* algorithm. The weights are initialised with random values, and they are changed in the direction that will reduce the error:

$$\Delta \mathbf{w} = -\eta \frac{\delta \mathcal{J}(\cdot)}{\delta \mathbf{w}},$$

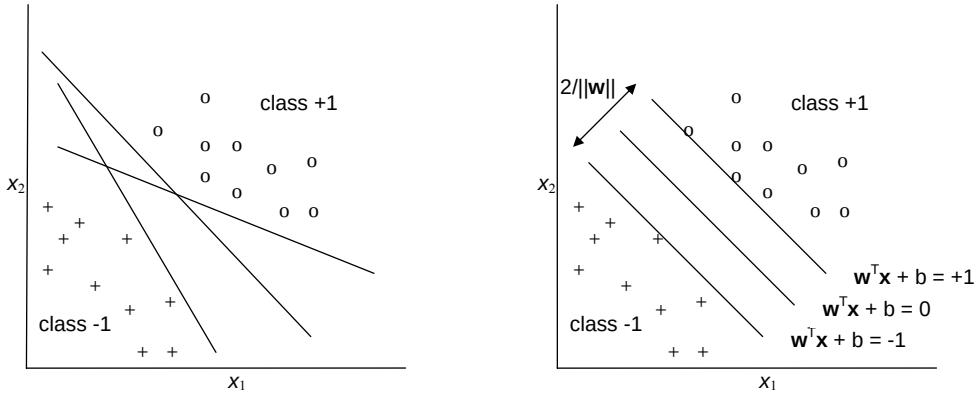
where η is the learning rate. After the complete training set has been fed to the network and the errors are accumulated, the weights are adjusted backward from the output nodes to the input nodes. This forward propagation and backward adjustment continues until $\mathcal{J}(\mathbf{w})$ converges. This specific network is called the *backpropagation neural network*.

The neural network has several advantages over regression techniques. Most important importantly it does not make any assumptions on the distribution of the data. It is tolerant to incomplete and noisy data, and is able to approximate any complex non-linear mapping if enough data is provided.

There are many disadvantages as well. The neural network operates as a black box; the hidden layer nodes have no particular meaning. The design of the network, like the choice of activation function and the number of hidden layers, is still based on heuristics. Large amounts of data are required, and the net might converge to a local minimum. Last, neural networks lack statistical properties. Hypothesis testing is hence impossible.

2.5 Support vector machines

The basic *support vector machine classifiers* (SVMs) try to find a hyperplane that separates two classes. The class of hyperplanes that is considered is $\mathbf{w}^T\mathbf{x} + b = 0$ with weight vector $\mathbf{w}$ and bias b, i.e., the linear hyperplanes. There may exist many separating hyperplanes, or decision boundaries, that separate the data of the two classes (cf. figure 2.2a). We can define a unique separating hyperplane using the *margin*: the minimal Euclidean distance of a pattern to the decision boundary. The hyperplane that maximises this margin is called the *maximal margin hyperplane*, cf. figure 2.2b.



(a) *Separating hyperplanes* (b) *Maximal margin hyperplane*

Figure 2.2 — Binary classification problem

We have rescaled $\mathbf{w}$ and b in such a way that all data points obey the following rules:

$$\begin{cases} \mathbf{w}^T\mathbf{x}_i + b \geq +1, & \text{if } y_i = +1 \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1, & \text{if } y_i = -1 \end{cases} \tag{2.1}$$

The margin of the classifier in the *canonical formulation* is given by $2/\|\mathbf{w}\|$. The classifier itself that identifies new patterns now becomes

$$y = \text{sign}\left[\mathbf{w}^T\mathbf{x} + b\right], \tag{2.2}$$

but as we will see later will never be evaluated in this form.

Maximising the margin is equivalent to minimising $\|\mathbf{w}\|$ in the canonical representation:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \mathcal{J}(\mathbf{w},b) = \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \qquad i = 1,\ldots,\ell \end{aligned}$$

Note that the single constraint is equivalent to the two decision rules from equation 2.1.

Before we explain the solution to this optimisation problem, we will add two enhancements to the SVM classifier: non-linearity and non-separability. The non-linearity is introduced by replacing $\mathbf{x}$ with $\boldsymbol{\varphi}(\mathbf{x})$, where $\boldsymbol{\varphi}(\cdot)$ is a non-linear function that maps $\mathbf{x}$ from input space into *feature space*: $\boldsymbol{\varphi}(\cdot) : \mathcal{X} \subseteq \mathbb{R}^n \longrightarrow \mathcal{H}$. This is depicted in figure 2.3: a problem that is not linearly separable in $\mathbb{R}^n$ might be separable in $\mathcal{H}$.

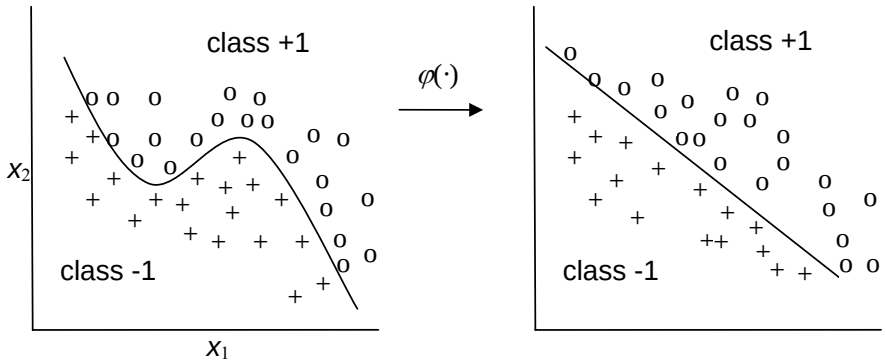


Figure 2.3 — Mapping from input space $\mathbb{R}^n$ to feature space $\mathcal{H}$

Non-separable data sets can be made separable by allowing for misclassifications. *Slack variables* ξ_i are added that measure the degree of violation for the i^{th} constraint. The optimisation problem is now given by:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \mathcal{J}(\mathbf{w},b,\boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{\ell}\xi_i, \qquad C \in \mathbb{R}^+ \\ \text{subject to} \quad & \begin{cases} y_i(\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_i) + b) \geq 1 - \xi_i, & i = 1,\ldots,\ell \\ \xi_i \geq 0, & i = 1,\ldots,\ell \end{cases} \end{aligned}$$

Minimising $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ maximises the margin between both classes in feature space, whereas minimising $\sum_{i=1}^{\ell}\xi_i$ minimises the misclassification costs. The positive regularisation constant C determines the trade-off between regularisation (C small) and empirical risk minimisation (C large).

Dual formulation

We will now use concepts from optimisation theory. Minimising a quadratic function under linear inequality constraints is solved using Lagrange theory. The Lagrangian to the maximal margin problem is given by:

$$\mathcal{L}(\mathbf{w},b,\boldsymbol{\xi};\boldsymbol{\alpha},\boldsymbol{\nu}) = \mathcal{J}(\mathbf{w},b,\boldsymbol{\xi}) - \sum_{i=1}^{\ell}\alpha_i[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{\ell}\nu_i\xi_i,$$

where $\alpha_i \geq 0$ and $\nu_i \geq 0$ are the Lagrange multipliers. The solution to this problem is given by the saddle point of the Lagrangian, i.e., by minimising $\mathcal{L}(\mathbf{w},b,\boldsymbol{\xi};\boldsymbol{\alpha},\boldsymbol{\nu})$ with respect to $\mathbf{w}$, b, and

$\boldsymbol{\xi}$, and maximising it with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\nu}$:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \max_{\boldsymbol{\alpha},\boldsymbol{\nu}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\nu})$$

$$\begin{cases} \dfrac{\delta \mathcal{L}(\cdot)}{\delta \mathbf{w}} = \mathbf{0} & \rightarrow \quad \mathbf{w} = \displaystyle\sum_{i=1}^{\ell} \alpha_i y_i \boldsymbol{\varphi}(\mathbf{x}_i) \\ \dfrac{\delta \mathcal{L}(\cdot)}{\delta b} = 0 & \rightarrow \quad \displaystyle\sum_{i=1}^{\ell} \alpha_i y_i = 0 \\ \dfrac{\delta \mathcal{L}(\cdot)}{\delta \xi_i} = 0 & \rightarrow \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, \ell \end{cases}$$

If we substitute the first expression into equation 2.2, the classifier is given by:

$$y = \text{sign}\left[\sum_{i=1}^{\ell} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right], \tag{2.3}$$

where $K(\mathbf{x}_i, \mathbf{x}) = \boldsymbol{\varphi}(\mathbf{x}_i)^T \boldsymbol{\varphi}(\mathbf{x})$ is a positive definite *kernel function* satisfying the Mercer theorem (Cristianini and Shawe-Taylor 2000). This is called the *dual formulation* of the SVM classifier. The Lagrange multipliers $\boldsymbol{\alpha}$ are computed as the solution to the following convex quadratic programming problem:

$$\max_{\alpha_i} \quad -\frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \alpha_i \alpha_j + \sum_{i=1}^{\ell} \alpha_i$$

$$\text{subject to} \quad \begin{cases} \displaystyle\sum_{i=1}^{\ell} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, \ell \end{cases}$$

The classifier construction problem has simplified to determining the α_i. These α_i are referred to as the *support values* and give the relative weight of their corresponding *support vector* $\mathbf{x}_i$.

Neither $\mathbf{w}$ nor $\boldsymbol{\varphi}(\mathbf{x})$ has to be calculated to find the separating hyperplane. No explicit construction of the non-linear mapping $\boldsymbol{\varphi}(\mathbf{x})$ is thus required. We will use the kernel function $K(\cdot, \cdot)$ instead. This means the feature vector $\boldsymbol{\varphi}(\mathbf{x})$ is only implicitly known, and it may even become infinite dimensional. The most commonly used kernel functions are:

- Linear $\qquad K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)$
- Polynomial $\qquad K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + c)^d, \qquad c \in \mathbb{R}, d \in \mathbb{N}, \gamma \in \mathbb{R}^+$
- Gaussian RBF $\qquad K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\dfrac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right), \qquad \sigma \in \mathbb{R}^+$

The SVM technique has several interesting properties. Compared to neural nets, there are two major advantages. First there is the global and unique solution of the SVM classifier, whereas neural nets might return local minima. Second, the support vectors have a geometrical meaning, and the support values α_i give the relative importance of the support vectors. Another advantage is the sparseness of $\boldsymbol{\alpha}$: the classifier is influenced only by the support vectors, i.e., the vectors that have $\alpha_i > 0$. Finally, the size of the classifier is dependent on the number of data points ℓ, not on the dimension of the feature space. This way we can use kernel functions like the radial basis function (RBF) that implicitly maps the data onto an infinite-dimensional feature space.

The kernel function is, on the other hand, one of the disadvantages as well. The user has to select this kernel function and set its parameter(s). This part is still based on heuristics. The second major limitation is speed and size. The training of the classifier is a quadratic programming problem with a size of number of training samples.

Cristianini and Shawe-Taylor (2000) and Vapnik (1998) explain support vector machines and the aforementioned derivations in detail.

Multiple classes

So far we have discussed the binary SVM classifier. There are two possible approaches to extend a binary to a multi-class classifier: combine several binary classifiers, or use a single machine scheme that considers all classes at once. Both Hsu and Lin (2002) and Rifkin and Klautau (2004) have compared several methods for multi-class SVMs. They both concluded that there is little difference in classification accuracy between the multi-class schemes[2]. Rifkin and Klautau pose that simple schemes like one-against-all and all-pairs are preferable to more complex error-correcting coding schemes or single-machine schemes. We will shortly introduce the former two coding schemes.

The *one-against-all* strategy constructs c binary SVM classifiers with training size ℓ in case of c classes. Each i^{th} classifier is trained with data where the i^{th} class has positive labels and all other classes have negative labels. The resulting class is given by the index i of the classifier that has the highest value for *latent variable* z that is given by the signed distance to the separating hyperplane:

$$z = \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) + b$$

Note that this equation is simply the SVM classifier of equation 2.2 without taking the sign of the function.

In *all-pairs* a classifier between each pair of classes is constructed, which leads to $\frac{1}{2}c(c-1)$ classifiers for the c-class problem (Hastie et al. 1998). The training set is thus only formed by these two classes, and will have a size of $2\ell/c$ on average. In the testing phase, all the classifiers are applied. There are several ways to combine the outputs to determine the preferred class. The most commonly used technique lets each classifier vote for its preferred class, and picks the class with the most votes. This technique is known as *majority voting* or *max-wins*. Other techniques involve estimation of the posterior probability (Hastie et al. 1998).

SVM techniques that take ordinality into account are still rare. In chapter 7 we will discuss how to address ordinality and give a more complete overview of multi-class SVMs.

Least squares support vector machines

Suykens et al. (2002) propose a modification of the SVM classifier: *least squares support vector machines* (LS-SVMs). The LS-SVM optimisation problem differs from the ordinary SVM classifier in two aspects. It uses a least squares cost function for the slack variables, i.e., ξ_i is replaced with ξ_i^2. The second difference is the replacement of the inequality constraints by equality constraints. The optimisation function is thus becomes as follows:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \mathcal{J}(\mathbf{w},b,\boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{2}C\sum_{i=1}^{\ell}\xi_i^2, && C \in \mathbb{R}^+ \\ \text{subject to} \quad & y_i(\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_i) + b) = 1 - \xi_i, && i = 1,\ldots,\ell \end{aligned}$$

Following the same reasoning as with standard SVMs, it can be shown that the LS-SVM classifier is obtained as the solution to the following linear system of equations:

$$\begin{pmatrix} 0 & \mathbf{y}^T \\ \mathbf{y} & \mathbf{G} + \frac{1}{C}\mathbf{I}_\ell \end{pmatrix} \begin{pmatrix} b \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{1} \end{pmatrix},$$

where $\mathbf{y} = [y_1 \ldots y_\ell]^T$, $\mathbf{1} = [1 \ldots 1]^T$, $\boldsymbol{\alpha} = [\alpha_1 \ldots \alpha_\ell]^T$, $\mathbf{I}_\ell$ a unity matrix, and *Gram* matrix $\mathbf{G}$ has elements $G_{ij} = y_i y_j \boldsymbol{\varphi}(\mathbf{x}_i)^T \boldsymbol{\varphi}(\mathbf{x}_j) = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ and satisfies the Mercer condition.

The LS-SVM classifier shares most of its properties with the standard SVM formulation, like the global and unique solution, the requirements it poses to kernel functions, and the equation of the classifier (equation 2.3). The major advantage of LS-SVMs is that the problem simplifies

[2] Hsu and Lin conclude that all-pairs and the DAG approach outperform other techniques, but their tables do not support this conclusion.

from a quadratic optimisation problem to a linear system of equations. This approach significantly reduces the complexity and computation time for solving the problem. The disadvantage, however, is the lack of the theoretical background. The generalisation error of the original SVM formulation is bound (Vapnik 1998), but this property is lost in LS-SVMs. Furthermore, the classifier is no longer sparse in $\boldsymbol{\alpha}$. It can be shown that $\alpha_i = C\xi_i$, and since ξ_i will practically never equal zero, $\alpha_i \neq 0$. Suykens proposes a pruning technique to achieve sparseness and reports good generalisation results. Van Gestel et al. (2004) benchmark LS-SVMs against ordinary SVMs and conclude that both have equal prediction accuracy. It should be said, however, that Van Gestel is part of the research group that developed LS-SVMs.

Chapter 3

Related work

Modern credit analysis dates back from the late nineteenth century, when credit markets began to issue and trade bonds. At the time, a rating depended on the judgement of experts, who looked at both the quantitative and the qualitative characteristics of a counterparty. The first works using formal statistical models date back to the 1960s, when Horrigan (1966) used standard ordinary least squares regression on multiple ratios to predict potential financial distress.

More recently, artificial intelligence techniques have been proposed as alternatives for the statistical methods, among which rule-based expert systems and machine learning techniques such as neural networks. Machine learning techniques are free of the parametric and structural assumptions that underlie statistical methods. These techniques learn the particular structure of a model from the data in an inductive manner. This section describes previous work in the field of credit rating chronologically starting from 1988. Table 3.1 provides a schematic overview of the discussed articles.

3.1 The neural network era

Dutta and Shekhar (1988) were the first to apply backpropagation neural networks (BPNNs) to the rating replication problem. They compared neural networks with multiple regression to discern AA-rated from non-AA counterparties. Both two- and three-layered neural networks outperformed the regression model. A two-layered neural network with ten financial variables correctly classified 88.3%, as compared to 64.7% for the regression model. The data consisted of thirty training samples and seventeen test samples in only two rating categories, clearly not enough to be able to compare these results with other techniques.

Surkan and Singleton (1990) compared the neural networks with multiple discriminant analysis. They used a population of 126 bond patterns with seven financial variables. The number of rating classes was reduced to two: one consisting of Aaa bonds, and the other consisting of A1, A2, and A3. They selected sixteen prototype bonds for training, and tested it on twenty samples from each of the two classes. The result was that neural networks outperformed MDA with accuracies of 88% compared to 39%.

Utans and Moody (1991) and Moody and Utans (1995) obtained a 30.6% accuracy in neural networks using fifteen rating categories. The complete data set of 196 firms was used to initialise the weights in the neural network, which implies a possible bias in the model. Neural networks significantly outperformed linear regression, but the classification accuracy for the latter have not been given. Moody and Utans assessed neural nets in the case of five and three distinct rating classes as well, resulting in performances of 63.8% and 85.2% respectively.

Kim et al. (1993) evaluated neural networks, linear regression, ordinal logistic regression, discriminant analysis, and an ID3 rule-based system. They collected 168 data points in six rating classes ranging from AAA to B. The population was almost evenly divided over the rating classes in both the training and the test set. Neural networks were the best performers with an accuracy

ratio of 55.2% on an out-of-sample test set of 58 counterparties.

Maher and Sen (1997) compared the neural network approach with ordinal logistic regression. Overfitting was avoided using a separate train (60%), test (20%), and validation set (20%). A maximum performance of 70% was obtained for neural networks on six rating classes and a population of 299.

Kwon et al. (1997) improved neural networks to better represent the ordinal nature of rating classes by introducing ordinal pairwise partitioning (OPP). The main idea of this approach was to partition the data set in an ordinal and pairwise manner according to the output classes. Separate neural networks were then trained using the different partitioned data sets. Their OPP model predicted the ratings of 71–73% in five rating classes correctly, whereas conventional neural networks scored 66–67% and multiple discriminant analysis 58–61%.

Chaveesuk et al. (1999) researched two additional supervised neural networks paradigms: radial basis function (RBF) and learning vector quantisation (LVQ). The RBF network is similar to the backpropagation neural network, except for the behaviour of the hidden layer. The sigmoidal/sum activation function of the BPNN is typically replaced by a Gaussian kernel, where each of the hidden neurons calculates the 'closeness' between the input vector and its centre. An interesting fact is that this design is conceptually similar to a support vector machine classifier with an RBF kernel function. The LVQ network is merely a supervised variant of Kohonen's self-organising maps. The two methods were benchmarked against conventional neural nets, linear regression and logistic regression, on a population of sixty training samples and thirty test samples. The prediction accuracy for RBF (23.3–38.3%) and LVQ networks (36.7%) was significantly worse than the LINREG (48.3%), LOGREG (53.3%) and BPNN implementations (51.9–56.7%).

3.2 Alternative artificial intelligence approaches

Case-based reasoning (CBR) is a problem solving technique that re-uses past cases and experiences to find a solution to a problem. The key issue of CBR lies in the case indexing process. There are three possible approaches to indexing: nearest-neighbour (NN), inductive, and knowledge-guided. Shin and Han (1999, 2001) applied all three CBR variants to the credit rating problem, and included a hybrid NN-knowledge variant based on genetic algorithms (GA). They concluded that both this hybrid variant (75.5%) and the inductive learning variants (62.8–70.0%) had a higher prediction accuracy than multiple discriminant analysis (60%), ID3 (59%), and other CBR variants (61–62%).

Rating agency Standard and Poor's uses support vector machines (SVMs) in their CreditModel rating application. CreditModel produces quantitatively derived estimates of Standard and Poor's credit ratings for both public and private counterparties. Friedman (2002) discloses a few details of the methodology in a technical white paper. The key variables mainly consist of financial data, and the number of rating categories is nineteen (AAA to CCC- including pluses and minuses). Friedman benchmarks the proximal support vector machine (PSVM) technique (Fung and Mangasarian 2001) with neural networks and Nadaraya-Watson regression, which is a type of non-linear regression. The PSVM technique gives a 30.8% performance, compared to 23.6 and 29.5% for neural networks and Nadaraya-Watson regression respectively. These results are averages taken from thirteen different training and out-of-sample test sets.

Van Gestel et al. (2003) applied another SVM approximation to the credit rating replication problem: least squares support vector machines (LS-SVMs). The binary LS-SVM classifier is extended to a multi-class classifier using a all-versus-all coding technique with majority voting decoding. They compared this approach to linear regression, logistic regression, and neural networks. The data set consisted of 3599 observations of banks and 79 candidate inputs, where 30% of the data set was used for out-of-sample testing. They concluded that the LS-SVM methodology yields significantly better results (54.5%) than linear regression (28.9%), logistic regression (36.6%), and neural nets (27.5%). The article does not provide any details on the choice of the kernel function and the hyperparameter selection method.

Huang et al. (2004) describe their support vector machine approach in more detail. They experimented with different kernel functions, hyperparameter settings and multi-class approaches

using the BSVM software package. Their best performing set-up consisted of a radial basis kernel function with Crammer and Singer's formulation for multi-class SVM classification (Crammer and Singer 2000), although comparable results were derived using a third-order polynomial kernel function. They compared SVMs to both logistic regression and neural networks, performing ten-fold cross-validation on a data set of 265 American companies. The support vector machines and neural networks consistently outperformed the logistic regression model.

3.3 Discussion

Table 3.1 shows that in early work, backpropagation neural networks outperform both MDA and LINREG by large. These results are, however, based on rather small data sets, which makes the results doubtful. More carefully conducted work shows less dramatic but still prominent advantages of neural nets over the standard techniques. Apparently the linear relationship assumption that underlies both linear regression and multiple discriminant analysis does not hold.

Not surprisingly, the competition between logistic regression and BPNNs proved to be more challenging. The prediction accuracy of both techniques is roughly similar in all cited work. The logistic model is better suited for the credit rating problem than MDA and LINREG; it is indeed most often used in cases where latent probability values are estimated from binary or ordinal multi-class observations.

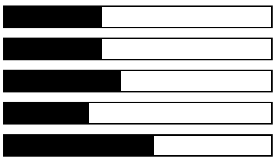
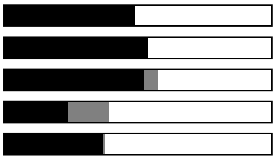
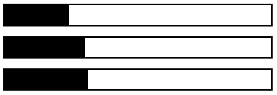
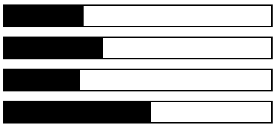
Several new approaches have been presented, of which support vector machines seem the most promising. The three most recent papers all claim that the SVM approach leads to the best results. The results presented by Van Gestel et al., however, are rather remarkable; all previous research indicated that neural nets performed equal to or better than logistic regression. Moreover, the performance of SVMs is nearly two times higher that that of neural networks. The neural network scheme has apparently not been designed carefully. The most surprising fact is a classification accuracy of over 50% in the fifteen-class problem. Although it is very difficult to compare the results of different authors due to different numbers of rating classes and variables, we can easily conclude that this value outperforms any other research. One cause is the distribution of the data set: seven out of fifteen classes contain radically less observations than the other eight. In practice this simplifies the problem to an eight-class problem, since new objects will hardly be classified to these rare classes, at low misclassification costs.

Another possible cause of this implausible good result can be found in the description of the data set. The 3599 observation are based on data from 831 companies over eight years. Since many companies hardly in change in terms of their ratings and financial ratios, many similar patterns occur in both the training and the test set, with similar associated rating classes. The classifier will thus be heavily biased to companies that are already present in the data set, and its real out-of-sample accuracy might be limited. This conclusion even further stresses the question why neural nets underperform, since this technique is based on correctly classifying patterns.

This latter cause hold for the results of Huang et al. as well. Their data consists of 265 observations from ten years of data of 36 companies, resulting in a similar biased 'out-of-sample' test set.

Although the SVM results might be somewhat questionable in absolute sense, we still regard them as promising for SVMs do outperform other techniques. This might indicate that the relation between the input and the rating class is more complex than a linear or logistic relation. Logistic regression, however, remains a competing technique. We will therefore research both techniques on the ABN AMRO data set. The next chapter describes the data that serves as input for any technique. Next chapter 5 introduces a new framework for classification, in which any of the techniques can be incorporated. Chapter 7 describes the experimental set-up of our SVM classifier, and chapter 8 compares the results with regression techniques and the existing credit rating system.

Table 3.1 — Prior research to artificial intelligence techniques

Study	Year	Method	Accuracy[1]	#variables	#samples	#classes
Dutta and Shekhar	1988	LINREG BPNN		6, 10	47	2
Surkan and Singleton	1990	MDA BPNN		7	126	2
Utans and Moody	1991, 1995	BPNN		10	196	5
Kim et al.	1993	LINREG MDA LOGREG ID3 BPNN		8	168	6
Maher and Sen	1997	LOGREG BPNN		5, 6, 7	299	6
Kwon et al.	1997	MDA BPNN BPNN w/OPP		24	3085	5
Chaveesuk et al.	1999	LINREG LOGREG BPNN RBF LVQ		8	90	6
Shin and Han	1999, 2001	MDA ID3 CBR CBR w/GA		12	3886	5
Friedman	2002	BPNN NWREG SVM		N/A	N/A	19
Van Gestel et al.	2003	LINREG LOGREG BPNN SVM		79	3599	15
Huang et al.	2004	LOGREG BPNN SVM		5, 14	265	5

[1] The black bars indicate the classification accuracy for the worst performing architecture; the grey bars give the accuracy of the best performing architecture.

LINREG: linear regression
MDA: multiple discriminant analysis
LOGREG: ordinal logistic regression
BPNN: backpropagation neural network
ID3: inductive decision tree
OPP: ordinal pairwise partitioning
RBF: radial basis function network
LVQ: learning vector quantisation network
CBR: case-based reasoning
GA: genetic algorithms
NWREG: Nadaraya-Watson regression
SVM: support vector machines

Chapter 4

Data description

This chapter describes the variable construction and selection process. An extensive variable selection process was performed in 1998, when ABN AMRO was about to move to a more sophisticated default risk model than the one used at the time. One of the candidates was an off-the-shelf program called Moody's Risk Advisor (MRA).[1] Two independent expert panels were formed to decide on which ratios and other factors should serve as inputs for the model, using the MRA default choices as a starting point. Zondag (1998a, 1998b) describes the outcome of the panel meetings.

In this chapter we will first examine the variables that are pointed out as useful in the credit rating literature. Then we will review the ABN AMRO expert panels' decisions, focusing on all variables that have been under discussion. Next comes our own research. The third section discusses data preprocessing, such as outlier removal and transformations. An exploratory analysis on the data is given next, of which the conclusions are presented in the fifth section.

The data collection and cleaning process was a very time-consuming task which resulted in several important achievements. The added value for readers interested in rating models and data analysis is however limited, so we have decided to leave it out of the main portion of the report. More information on this topic can be found in appendix B.

4.1 Theory

Credit rating agencies have been rating corporations since the nineteenth century. A company's rating depends on a variety of different inputs. The main source for these variables is the company's annual report. All large companies have an obligation to both their shareholders and the government to publish an annual report, including the company's financial statement.

To determine the financial position of a company, it is common in economics to calculate ratios from the different sheets of their financial statement. The financial statement consists of a balance sheet, a profit and loss account, and a cash flow statement. The balance sheet reflects the value of the company's assets, liabilities, and the equity at a specific point in time. The profit and loss account is compiled at the end of the fiscal year (or another accounting period) to show gross and net profit or loss. The cash flow statement shows the borrower's 'free' operational cash flow, which will be available for debt service. Appendix A.1 gives templates for the different parts of the financial statement.

Financial data are only one of the many possible information sources. In fact, large credit rating firms evaluate a whole array of characteristics to assess the probability of default of borrowers. Other aspects can include the analysis of country risk and industry risk, or the vulnerability to competitive pressures and unexpected events. Table 4.1 gives the main characteristics used by Standard and Poor's (2004).

The large rating firms have informal meetings with the company's top executives to evaluate its prospects and access confidential information (e.g., plans for new products to be launched in

[1] At that time the program's name was Lending Advisor Encore, developed by Crowe Chizek.

Table 4.1 — Main risk factors in the analysis of corporate credit risk

Risk source	Profile
Business risk	Industry characteristics Competitive position, e.g., - marketing - regulation - technology
Financial risk	Financial characteristics Financial policy Profitability Capital structure Cashflow protection Financial flexibility

the next year). Researchers will usually not be able to retrieve this information from a company. Credit rating research has therefore mainly focused on the data that is available to them, i.e., financial data.

There are many ratios that may possibly characterise the financial situation of a company. Extensive research has been performed by Chen and Shimerda (1981), who conclude that there are seven factors that are crucial to the prediction of defaults. These are given in table 4.2.

Table 4.2 — Financial drivers of default risk

Factor	Description	Example
Volatility	A measure for agility. A higher equity volatility implies a higher probability of a firm's asset value falling below its level of debt, which implies insolvency.	Historical volatility
Size	Larger companies are generally more diversified in their exposure to geographies, products, and people, and this lowers their prospective volatility. Note that this greatly overlaps with volatility.	Turnover
Profitability	Higher profit lowers default probabilities. Combining profitability with interest expense makes it a combination of leverage and profitability	Gross Profit Margin, Total Debt to EBITDA, Interest Coverage Ratio
Leverage	Higher leverage implies higher default probabilities.	Gearing Ratio, Equity Ratio
Liquidity	Lower liquidity implies higher default probabilities.	Current Ratio, Quick Ratio
Growth	Both high and low growth rates are associated with higher default probabilities	Annual Turnover Growth
Inventories	Higher inventory levels imply higher default probabilities.	Stock Days

The use of ratios has some side-effects as well. We will have to take into account that a negative ratio is the result of either a negative numerator or a negative denominator. Both ratios, that appear to be the same, might in fact have a completely different impact on the UCR. A denominator can have a value close to zero, leading to an extremely high or low value (outlier).

In general, financial ratios will not follow the Gaussian distribution. In fact, the y-axis will be a semi-asymptote. The shape of the ratio versus the UCR is roughly known from theory. For example, the *Current Ratio* is known to have a logarithmic shape, whereas the *Annual Turnover Growth* has a U-shape. We will have to perform some sort of transformation to avoid overfitting on for instance extreme values and fat tails. Falkenstein (2002) proposes four different methods:

- Replace the ratio with its percentile
- Transform the ratio into a standardised Gaussian variable (e.g., $\frac{x-\mu}{\sigma}$)
- Apply a sigmoidal function (e.g., $\frac{1}{1+e^{-x}}$) to the ratio or its standardised Gaussian variable
- Use the non-parametric univariate default estimate generated by each variable

4.2 Variable candidates

The risk posed by a counterparty consists of the financial risk and the business risk. The financial risk will be measured by financial ratios, their derivatives trend and volatility, and answers to subjective questions. The business risk consists of three macro-economic scores. The following subsections treat these elements in detail.

4.2.1 Raw ratio value

Several financial ratios calculated from the latest financial statement will serve as inputs. The experts selected fourteen financial ratios that they felt were most predictive. They are divided into four categories: operations, liquidity, capital structure, and cash flow and debt service. Table 4.3 gives the chosen ratios and their categories. Appendix A.2 shows how to calculate the different ratio values for the standard annual case. When we take non-annuals into account, we need to annualise all profit and loss account elements.

Table 4.3 — Financial ratios

Category	Financial ratio
Operations	Gross Profit Margin
	Operating Profit Margin
	Annual Turnover Growth
	Return on Capital Employed
Liquidity	Current Ratio
	Quick Ratio
	Debtor Days
	Stock Days
	Creditor Days
Capital structure	Gearing Ratio
	Equity Ratio
Cash flow and debt service	Interest Coverage Ratio
	Total Debt to EBITDA
	NOFF to Financing Charges

When compared to Chen and Shimerda, operations correspond to profitability and growth; liquidity is liquidity including inventories; capital structure and leverage are synonyms; and cash flow and debt service combines leverage and profitability. A remarkable missing aspect with respect to Chen and Shimerda's findings is the size of the counterparty. The panels found that this variable would largely be covered by the subjective questions. In our research we have added the log of two different size variables (*Total Assets* and *Turnover*), to ensure that all the suggested aspects will be covered.

Many ratios are based on two consecutive statements. *Debtor Days*, for instance, gives an average value of two statements. When financial statements are available for the past n years, only one statement is available for calculations in year n. Most ratios will now be based on this statement only. *Annual Turnover Growth* and NOFF *to Financing Charges* however require a second statement; without this statement, the ratio is omitted, resulting in missing values. Thus, when n financial statements are available, $n-1$ of the *Annual Turnover Growth* and NOFF *to Financing Charges* ratios are available, and n of the other ratios.

Some ratios are not relevant for particular industry sectors. The transportation sector, for instance, will not have any *Cost of Goods Sold* and thus not have any *Creditor Days* or *Stock Days*. For all five ratios where this is the case (*Gross Profit Margin*, *Operating Profit Margin*, *Creditor Days*, *Stock Days*, and *Debtor Days*), calculating the raw value will lead to missing values.

4.2.2 Ratio trend

The trend itself over these years is regarded as a useful variable as well, although strictly speaking the trend is not one of the aspects pointed out by Chen and Shimerda. The trend is equal to the slope of the ordinary least squares regression line:

$$trend(\mathbf{x},\mathbf{y}) = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

where y_i is the ratio, x_i the date of the ratio, and n the number of financial statements. x_1 is set to 0 and refers to the date of the oldest statement. The other x_i equal the date of statement i less the date of the oldest statement, calculated in months.

At least three financial statements have to be available to be able to calculate the trend. In theory two will suffice, but we have chosen to set three as a minimum to have a more reliable estimation of the trend. In many cases less than three statements are available, resulting in missing values. The maximum has been set to four.

4.2.3 Ratio volatility

The volatility is comparable to the standard deviation of the trend. Just as with the trend, at least three statements must be available, which is the theoretical minimum as well. The value of the volatility is always a non-negative value, and will have a log-shaped distribution.

The volatility is included for each of the financial ratios using the latest n financial statements, up to a maximum of five. Among the several ways to calculate the volatility, we have chosen to use the following formula:

$$\overline{\Delta y} = \frac{\sum_{i=1}^{n-1}(y_{i+1} - y_i)}{n-1}$$

$$volatility(\mathbf{y}) = \frac{n}{\sum_{i=1}^{n} y_i}\sqrt{\frac{\sum_{i=1}^{n-1}(y_{i+1} - y_i - \overline{\Delta y})^2}{n-1}},$$

where y_i is the ratio, $\overline{\Delta y}$ is the average change in y, and n is the number of financial statements.

This formula only holds when the distance between the elements of $\mathbf{x}$ are all equal. This happens for instance when the financial statements of 2001–2005 are all composed in December, resulting in twelve months difference between subsequent statements. The formula that takes unequal distances into account can be found in Dikkers (2005b).

4.2.4 Subjective questions

Not only quantitative information is taken into account. The account manager has to answer several subjective questions. Eight questions were found to be relevant for determining the prob-

ability of default, all of which are listed in table 4.4. The number of possible answers may differ, but they are always ordinal, ordered from negative to positive.

Table 4.4 — Subjective questions and possible answers

Subjective question	Possible answers
Competitive leadership	Weak, moderate, strong, dominant
Trading area	Local, national, regional, global
Market conditions	Negative growth, flat market, average growth, strong growth, high growth
Supplier risk	Very high, high, average, low, very low
Accounting risk	High, moderate, low
Customer concentration	Very high, high, average, low, very low
Stock liquidity	Very low, low, average, high, very high
Access to capital	Low, average, high, very high

4.2.5 Macro-economic factors

Although industry differences are covered by peer group percentiles, this might be counterproductive if, e.g., some industries have higher default rates because of their higher-than-average leverage (and thus risk). The ABN AMRO Industrial Sector Research department determines a score on a scale $[0, 100]$ for each of the 68 different industries, which combines risk and prospects. When a counterparty is active in multiple industries, a weighted average of the scores is calculated to come to the final *Industry Score*.

An additional risk is posed by the country that a counterparty operates in. Similar to the *Industry Score*, the Country Risk department calculates a score that reflects the risk of having sales in a country: the *Country of Sales Score*. Again this might be a weighted average. Finally, the residence risk is assessed to form the *Country of Residence Score*. This score is also provided by the Country Risk Management department.

4.3 Preprocessing

In this section we will discuss three aspects of preprocessing: outlier detection and removal, missing values, and the possible transformations.

4.3.1 Outliers

Outliers have a large influence on most regression and classification techniques. Fortunately they only appear in the financial ratios and their derivatives (trend and volatility); subjective questions and macro-economic scores always have a 'clean' value. Our first task is to identify the cause of these outliers. A robust method to remove outliers is presented next.

Causes

The outliers in the trend and volatility of financial ratios are always caused by one or more extreme values of the ratios themselves. Focusing on these raw variables will explain all outliers.

We can identify two types of outlier in the raw ratio values. Outliers with very high and low values are regularly seen; these were mostly caused by their denominators being close to zero. Values might even be $\pm\infty$ when the denominator equals zero.

Another form of outlier is caused by numerators or denominators having an unexpected sign. Most of the ratios in table 4.3 have both their numerator and denominator non-negative, or only one of these can be negative. For *Interest Coverage Ratio* and NOFF *to Financing Charges*, however, difficulties might arise. The numerator can take any value and the denominator is

expected to be non-negative. However, the denominator might in practice be negative as well, resulting in unexpected values for these two ratios. These values will be overridden to $-\infty$ or $+\infty$ based on expert knowledge.

Solutions

There are many ways to handle outliers. The most common way is to replace an outlier by the sample mean plus or minus two times the sample standard deviation, assuming that 5% of the values are outliers. However, since infinite values appear regularly, the sample standard deviation cannot be calculated. When we replace infinite values, with for instance ± 999, the sample standard deviation will largely depend on the choice of the value this replacement value, as will the outlier replacement value.

A more robust method is to replace the values better than the $100(1-i)^{\text{th}}$ percentile and the values worse than the $100(i)^{\text{th}}$ percentile with their respective values. The sample mean and standard deviation will now correspond to the Winsorised mean and standard deviation. The mean is given by:

$$\mu_{W(i)} = \mu_{W(j/\ell)} = \frac{1}{\ell}\left(jx_{j+1} + \sum_{i=j+1}^{\ell-j} x_i + jx_{\ell-j} \right), \tag{4.1}$$

where j is the number of observations to skip, the total of ℓ observations are ordered, and $j/\ell = i$. The standard deviation is calculated similarly.

We have set i to 2.5 in order to obey the 95%-rule. While examining the results of this outlier removal method, it appeared that for many ratios valuable information was deleted from the data set, whereas for other ratios too many outliers were still present. The underlying assumption that 95% of the values are correct evidently does not hold, nor would any other value.

This implies that the number of outliers has to be estimated first, which might create a bias in the dataset. Resti (2002) circumvents this problem by having experts set maximum and minimum acceptable values for each variable. No assumption is needed regarding the percentage of outliers.

Minimum (maximum) values have been extracted from the current MRA system. These values indicate the extremes where performing worse (better) does no longer affect the probability of default, according to ABN AMRO experts. The range of allowed values is very small compared to research peers and would lead to extremely high percentages of outliers (Resti 2002). Therefore we have tripled the range for each variable, e.g., an original range of $\langle 1, 3\rangle$ would now become $\langle -1, 5\rangle$. This proved to be the most robust outlier correction method.

The results are given in table 4.5. The minimum and maximum accepted value are given in column three and four, and the next two columns state the percentage of observations that needed to be replaced by either the minimum or the maximum. The final column shows the percentage of missing values, which will be discussed in the next subsection.

4.3.2 Missing values

Just as in the previous subsection, we will first try to identify the cause, and then provide a solution to the missing values problem.

Causes

Two types of missing value are present in the data: expected and unexpected missing data. Expected missing data only appears in ratios and their derivatives. The most common cause is an insufficient number of financial statements, such that the trend and volatility cannot be calculated. Another cause is the inapplicability of certain ratios to particular sectors of industry, like the *Gross Profit Margin* for the services industry. Unexpected missing data is caused by the incomplete data storage procedures that were used before August 2004. The values of *Gross Profit Margin*, *Operating Profit Margin*, and *Annual Turnover Growth* were not stored up to that date.

Table 4.5 — All available variables

	Variable	Min value	Max value	Replaced by min	Replaced by max	Missing values
Raw values						
R01	Gross Profit Margin	-0.9	1.8	0.1%	0.0%	7.1%
R02	Operating Profit Margin	-0.8	1.5	0.9%	0.1%	1.2%
R03	Annual Turnover Growth	-0.4	0.5	1.2%	10.7%	8.5%
R04	Return on Capital Employed	-0.2	0.4	3.1%	8.7%	0.0%
R05	Current Ratio	-	5.3	-	3.6%	0.1%
R06	Quick Ratio	-	3.5	-	4.4%	0.1%
R07	Debtor Days	-	270.8	-	1.0%	4.4%
R08	Stock Days	-	397.9	-	1.0%	12.1%
R09	Creditor Days	-	516.0	-	1.9%	7.1%
R10	Gearing Ratio	-2.5	7.3	3.7%	2.3%	0.0%
R11	Equity Ratio	-0.4	0.9	2.5%	3.5%	0.1%
R12	Interest Coverage Ratio	-5.0	11.5	4.7%	33.0%	0.6%
R13	Total Debt to EBITDA	-3.0	7.5	3.4%	7.8%	0.0%
R14	NOFF to Financing Charges	-4.5	11.3	5.2%	16.1%	0.9%
Percentiles						
P01	Gross Profit Margin	-	-	-	-	15.8%
P02	Operating Profit Margin	-	-	-	-	1.2%
P03	Annual Turnover Growth	-	-	-	-	8.5%
P04	Return on Capital Employed	-	-	-	-	0.0%
P05	Current Ratio	-	-	-	-	0.1%
P06	Quick Ratio	-	-	-	-	0.1%
P07	Debtor Days	-	-	-	-	4.4%
P08	Stock Days	-	-	-	-	20.0%
P09	Creditor Days	-	-	-	-	16.3%
P10	Gearing Ratio	-	-	-	-	0.0%
P11	Equity Ratio	-	-	-	-	0.1%
P12	Interest Coverage Ratio	-	-	-	-	0.6%
P13	Total Debt to EBITDA	-	-	-	-	0.0%
P14	NOFF to Financing Charges	-	-	-	-	0.9%
Trends						
T01	Gross Profit Margin	-0.2	0.2	0.8%	1.0%	12.9%
T02	Operating Profit Margin	-0.1	0.1	3.5%	6.5%	7.8%
T03	Annual Turnover Growth	-0.3	0.3	5.8%	2.6%	26.3%
T04	Return on Capital Employed	-0.3	0.3	1.7%	3.1%	6.5%
T05	Current Ratio	-4.0	6.5	0.5%	0.3%	6.5%
T06	Quick Ratio	-1.2	1.6	1.0%	1.0%	6.5%
T07	Debtor Days	-32.0	31.0	3.1%	1.3%	9.0%
T08	Stock Days	-40.0	38.0	3.7%	1.3%	13.9%
T09	Creditor Days	-29.0	28.0	7.5%	3.3%	12.5%
T10	Gearing Ratio	-2.8	2.6	3.6%	3.2%	6.5%
T11	Equity Ratio	-0.2	0.2	2.3%	3.7%	6.5%
T12	Interest Coverage Ratio	-5.0	5.5	8.7%	17.3%	7.1%
T13	Total Debt to EBITDA	-1.5	1.5	12.4%	7.8%	6.5%
T14	NOFF to Financing Charges	-1.5	2.3	12.3%	14.0%	21.3%
Volatilities						
V01	Gross Profit Margin	-	3.0	-	0.4%	12.9%
V02	Operating Profit Margin	-	3.0	-	7.0%	7.8%
V03	Annual Turnover Growth	-	4.5	-	12.3%	26.3%
V04	Return on Capital Employed	-	3.0	-	7.1%	6.6%
V05	Current Ratio	-	1.5	-	0.7%	6.5%
V06	Quick Ratio	-	1.5	-	0.7%	6.5%
V07	Debtor Days	-	1.5	-	1.8%	10.3%
V08	Stock Days	-	1.5	-	1.5%	16.9%
V09	Creditor Days	-	1.5	-	1.7%	13.0%
V10	Gearing Ratio	-	3.0	-	4.3%	10.6%
V11	Equity Ratio	-	3.0	-	1.7%	6.5%
V12	Interest Coverage Ratio	-	3.0	-	7.7%	7.1%
V13	Total Debt to EBITDA	-	3.0	-	7.2%	10.6%
V14	NOFF to Financing Charges	-	3.0	-	8.9%	21.3%
Size variables						
S01	log(Turnover)	-	-	-	-	1.1%
S02	log(Total Assets)	-	-	-	-	0.0%
Macro-economic factors						
M01	Industry Score	-	-	-	-	0.0%
M02	Country of Residence Score	-	-	-	-	0.0%
M03	Country of Sales Score	-	-	-	-	0.0%
Subjective questions						
Q01	Competitive Leadership	-	-	-	-	0.0%
Q02	Trading Area	-	-	-	-	0.0%
Q03	Market Conditions	-	-	-	-	0.0%
Q04	Supplier Risk	-	-	-	-	0.0%
Q05	Accounting Risk	-	-	-	-	0.0%
Q06	Customer Concentration	-	-	-	-	0.0%
Q07	Stock Liquidity	-	-	-	-	0.0%
Q08	Access to Capital	-	-	-	-	0.0%

Solutions

The expected missing data has been replaced by median values. This is a simple yet robust method which causes the replaced variable to have a neutral effect on the rating class.

Handling the unexpected missing values proved to be less straightforward. We were able to retrieve the variables that are the basis for the *Gross Profit Margin* and *Annual Turnover Growth* and have thus recalculated their values. Unfortunately it was impossible to exactly recalculate the values of *Operating Profit Margin*, since one of the source variables was missing as well. Instead values has been derived based on all but this variable. Empirical research indicates that roughly 75% of the *Operating Profit Margin* values will be correct, whereas the impact on the other 25% will usually be lower but possibly higher than the real value.

4.3.3 Transformations

Section 4.1 indicated that transformations are required for proper data analysis, at least for the financial ratios. In this section we will examine which transformations are required for the variables given in the previous section.

We will utilise three types of transformation:

1. Standardisation to Gaussian variables
2. Percentile calculations
3. Transformations to scores based on expert opinions

The first type of standardisation can be used for all types of variable. The percentile calculations apply to the raw ratio values only. We will take industry effects on these raw ratio values into account as well. The latter transformation is similar to the non-parametric univariate default estimate from section 4.1, and will be calculated for all variables.

Standardisation

We have chosen to transform all variables into standardised Gaussian variables. This assumes that the underlying population of variables is normally distributed. Although this is rarely the case, it is a common transformation in credit rating literature, with acceptable results (Falkenstein 2002).

After the outlier replacement process from section 4.3.1, we can safely perform the standard Gaussian normalisation:

$$z = \frac{x - \mu}{\sigma}$$

Percentiles

The quality of the value of a ratio largely depends on the industry. For example, even the top players in the supermarket business will have very low profit margins when compared to other industries. We will eliminate the ratio's dependence on industry by comparing the ratio to its industry peers.

A natural choice for this comparison is the use of percentiles. A percentile is the rank of a value in an ordered group of values, i.e., the percentage of group members that perform worse than the member in question. The percentile values might be either ascending or descending, depending on whether a high value is good or bad.

For each of the 68 different industries, the 75^{th}, 50^{th} and 25^{th} percentile values are determined. These three percentile values are called quartiles, and will be referred to as Q_1, Q_2, and Q_3 respectively. The peer group assessment calculations will be based on these three pre-determined values and not on a real-time comparison with their peers. If a counterparty is active in multiple industries, a weighted average is taken over each of the quartiles.

The formula for calculating the percentile will have to be an S-shaped function that maps the variable x onto a percentile in the range $[0, 1]$. ABN AMRO has chosen the following formulae, where the percentile values are assumed to be in ascending order ($Q_1 \geq Q_2 \geq Q_3$).

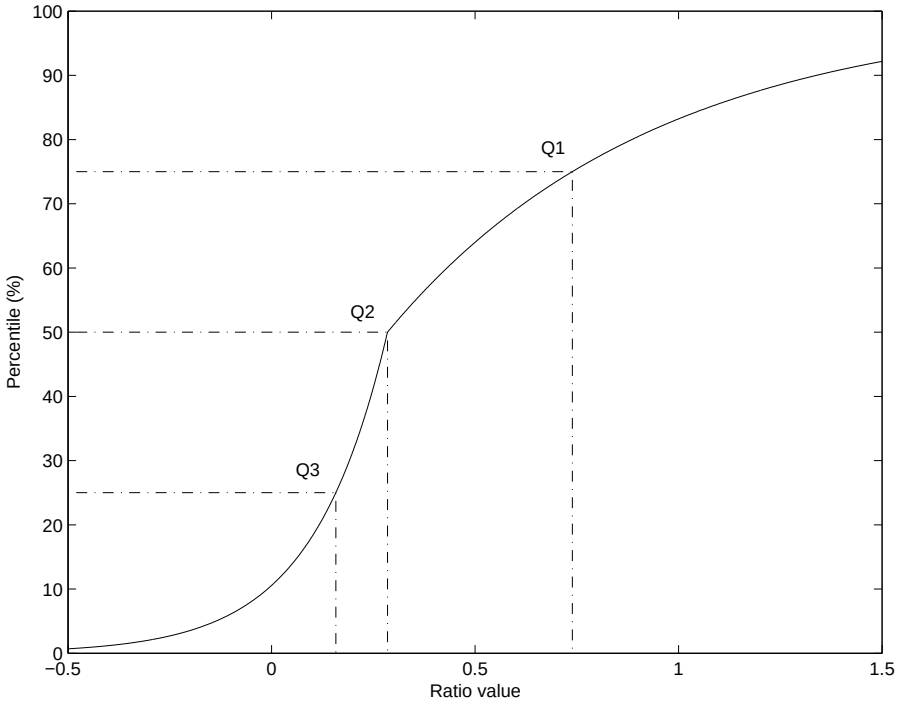


Figure 4.1 — Percentile example for *Operating Profit Margin* in industry peer group Agriculture Production, with $Q_1 = 0.739$, $Q_2 = 0.285$, and $Q_3 = 0.158$

$$percentile(x) = \begin{cases} 1 - \dfrac{2^{(Q_2 - x)/(Q_1 - Q_2)}}{2} & x \geq Q_2 \\ \dfrac{2^{(Q_2 - x)/(Q_3 - Q_2)}}{2} & x < Q_2 \end{cases}$$

The formulae for percentile values in descending order are similar and can be found in Dikkers (2005b). A graphical representation of these formulae is given in figure 4.1.

Because of the S-shape of the function, this type of transformation is robust to outliers. The original variables without outlier removal have therefore been used as inputs.

For companies in certain areas of industry, particular ratios might not be relevant. A service industry will not have any *Cost of Goods Sold* and thus not have any *Creditor Days*. The percentile calculations for these values will have to be omitted, for there is no peer group information available.

Scores

The third way to transform the raw values is a translation into scores. From a research point of view, this transformation is similar to Falkenstein's univariate default estimate per input as described in section 4.1. Within ABN AMRO insufficient default data was available to use default estimates in this mapping function. Instead, the experts estimated scores on a scale $[-50, 50]$.

Setting scores for the answers to subjective questions is a straightforward task: a score that represents the quality is assigned to each possible answer. The macro-economic scores are already scores but on a scale $[0, 100]$. Subtracting 50 leads to the score on the required scale.

Finally the continuous inputs need to be transformed. For each of these inputs, the experts have chosen five to seven benchmark values that they corresponded to scores. In the case of the raw ratio value of the *Current Ratio*, five values in the range of 0.7 to 3 have been associated with scores:

Benchmark value (V)	0.7	0.8	1	2	3
Associated score (S)	-45	-20	0	20	45

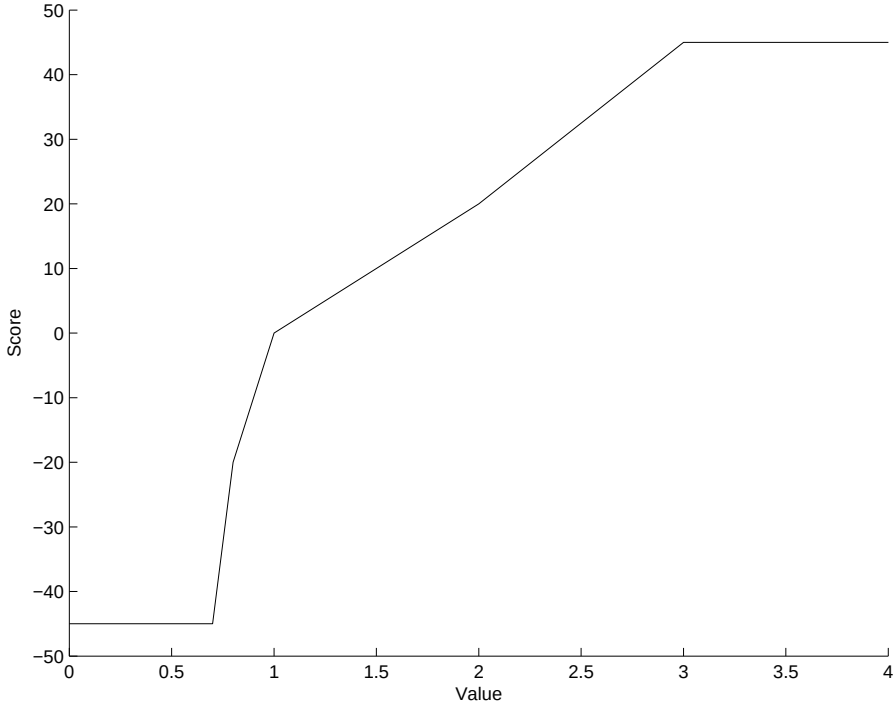


Figure 4.2 — Score example for the raw ratio value of *Current Ratio*

Based on these benchmark values and scores we can define a piecewise-linear non-parametric scoring function. The score corresponding to a particular variable is calculated by means of linear interpolation between the two closest benchmark values, cf. figure 4.2 for the *Current Ratio*. The score will be cut off at the minimum and maximum, i.e., no extrapolation takes place. It will now indicate the quality of an input on the scale $[-50, 50]$. In formula:

$$
\begin{aligned}
score(x) = \quad & \begin{cases} s_1, & x < v_1, \\ s_i + (x - v_i)\dfrac{s_{i+1} - s_i}{v_{i+1} - v_i}, & v_i \leq x < v_{i+1}, \\ s_n, & x \geq v_n, \end{cases} \\
\text{subject to} \quad & v_1, \ldots, v_n \in V, \\
& s_1, \ldots, s_n \in S.
\end{aligned}
$$

4.4 Exploratory data analysis

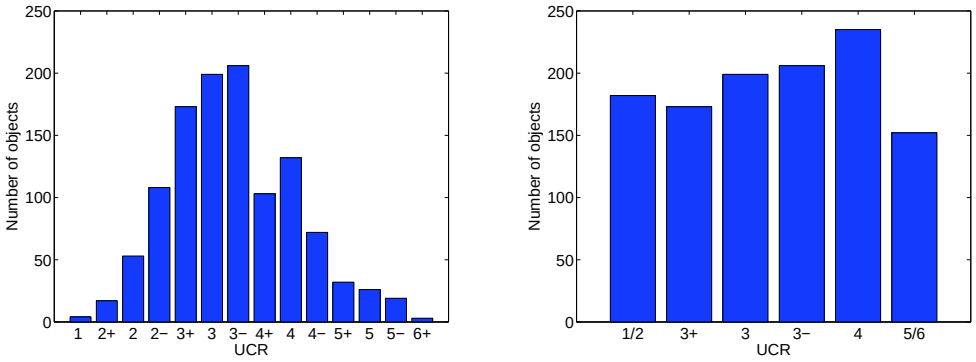
The experts have indicated many variables that could be used in further research. Before we can develop a model, however, we will perform thorough research on the data. The distribution of UCRs is given in the first subsection. Second, each variable should have predictive power, i.e., there should be a direct or indirect relation between a variable and the UCR class. The second subsection gives the results of several ways to assess the direct relation. Next we will discuss collinearity among the variables, and we will wrap up with the conclusions and the presentation of the final variable sets.

In this section we have focused on the raw values with outliers removed, and on the percentile

transformation of the financial ratios. When examining the results of the scores transformation, we found that for nearly all of the variables the predictive value and collinearity among other variables was similar to that of the raw value.

4.4.1 Distribution

We would expect that most of the counterparties are of average risk and thus have received an average UCR, i.e., around 3. Figure 4.3a shows us that this assumption approximately holds.



(a) *Original* (b) *After grouping* UCRs.

Figure 4.3 — Sample distribution

Many pattern recognition techniques require a reasonable amount of data in each class, which is clearly not the case for the outer classes. The ordinal nature of the rating problem allows us to group two or more adjacent UCR classes. The development of the models in the next chapters has therefore been based on data divided over only six classes. We have grouped UCR classes 1, 2+, 2, and 2− to 1/2; 4+ and 4 to 4; and 4−, 5+, 5, 5−, and 6+ to 5/6. The resulting distribution of counterparties is nearly uniform, as can be seen in figure 4.3b. The PDs associated with the new classes are set equal to the average PD of the classes they are formed by, weighted by their population sizes.

Another interesting aspect is to examine whether the data is indeed ordinally ranked, as we have stated in section 1.4. *Fisher discriminant analysis* is a linear discriminant analysis technique that, like all other LDA variants, uses new features that are linear combinations of the independent variables. These features are constructed in such a way that the means of the different classes are maximally separated. Since our data is ordinal, we expect that it is easier to separate classes that lie further apart than classes that are close to each other. More information on Fisher analysis can be found in Duda et al. (2001).

Figure 4.4 shows the empirical distribution among the classes for the first (and most predictive) Fisher feature. The x axis gives the value of this feature, and the y axis shows the number of occurrences. We can easily see that our ordinality assumption holds. Another observation is the large overlap between the different classes. This indicates it will be difficult to linearly separate the data.

4.4.2 Predictive power

This subsection describes the predictive power of variables with respect to the UCR. The UCRs can be interpreted in two ways: as metric values, and as ordinal values. For the metric interpretation, the UCR will be replaced by the historical probability of default or its logit. The ordinal interpretation only takes the rank of the UCR into account. After careful consideration, we have

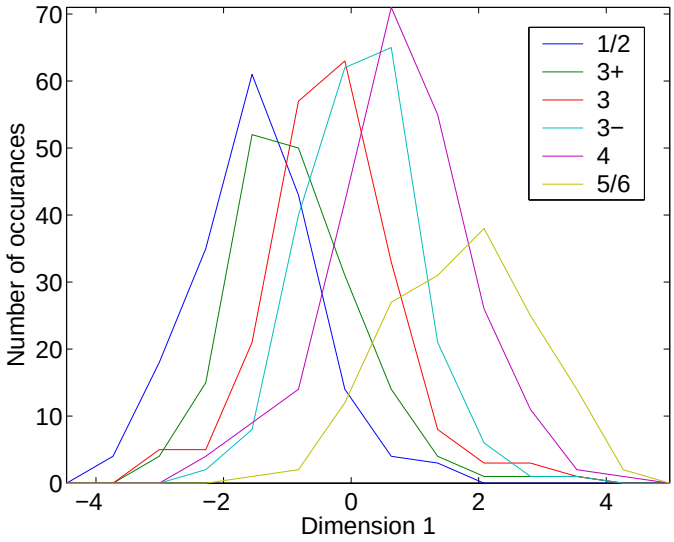


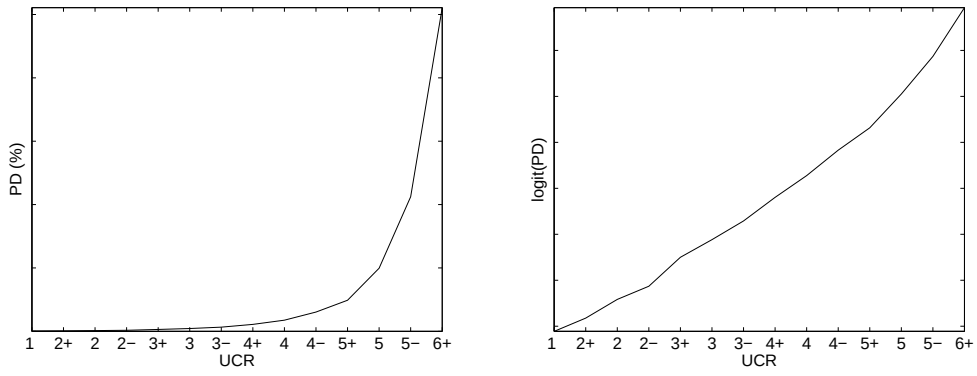
Figure 4.4 — Fisher analysis

chosen to research three measures of univariate predictive power:

- Pearson's correlation coefficient
- Spearman's rank order correlation coefficient
- Analysis of variance

The first technique regards the dependent variable as continuous, the second as ordinal, and the third technique nominal. This way we avoid the introduction of a bias towards continuous, ordinal, or nominal techniques.

The determination of the predictive power of the independent variables in this section is based on the fourteen-class data. In this way we try to include as much information as possible. A reduction to six classes yields higher observed values for the predictive power of the variables, but hardly affects the mutual order of importance of the independent variables.



(a) PD (b) *logit*(PD)

Figure 4.5 — PD versus the logit(PD) per UCR class. The y axes have intentionally been left blank for confidentiality reasons.

Pearson's correlation coefficient

The most popular method to calculate the correlation between two metric variables $\mathbf{x}$ and $\mathbf{y}$ is *Pearson's product-moment correlation coefficient*:

$$r = \frac{\text{Cov}(\mathbf{x}, \mathbf{y})}{\sigma_{\mathbf{x}} \sigma_{\mathbf{y}}}$$

where σ is the sample standard deviation. Pearson's correlation coefficient has a range $-1 \leq r \leq 1$, where $r \to \pm 1$ indicates a strong correlation and $r \to 0$ implies no correlation. Its significance can be determined using the t statistic with $n - 2$ degrees of freedom. Pearson correlation is greatly influenced by outliers, unequal variances, non-normality, and non-linearity.

We associate an estimate of the probability of default with each UCR class. These estimates are the actual default percentages over the year 2004, and are shown in figure 4.5a. Both the dependent and the independent variables are now metric, except for the answers to the subjective questions, which are still ordinal. A logit transformation of the probability of default might provide us with better results. This is motivated by the near-linear relationship between the rating class rank and the logit, as can be seen in figure 4.5b. Another conclusion that can be drawn from this observation is that the logit of the PD is merely a linear transformation of the rank of the UCR class. There is therefore no need to include correlations with the UCR rank as well. We will only calculate the correlation coefficient to both the original value and its logit.

The dependent variables are evidently not normally distributed, since there are only fourteen possible values for y_i. The outliers have been replaced with values set by experts, as described in section 4.3.1. Setting all these outliers to the same values however introduces a new distortion in the distribution. The Jarque-Bera test for normality indicates that only six out of 69 independent variables are normally distributed on a 0.1% significance level. Despite all of these disadvantages and violations, Pearson's correlation coefficient does give a good indication of the univariate predictive power of the variables.

Spearman's rank order correlation coefficient

Treating the dependent variables as continuous violates many assumptions. On the other hand, UCRs can be seen as ordinal data. *Spearman's rank order correlation coefficient* is widely used for ordinal data. It does not assume a linear relationship or any specific distribution, and is robust with respect to outliers. Therefore the original data set can be used without outlier replacements.

The two variables that we are researching, $\mathbf{x}$ and $\mathbf{y}$, are ranked, where each x_i is replaced by its rank $R(x_i)$ and y_i by $R(y_i)$. Ties are treated by averaging the tied ranks. Spearman's ρ is now calculated exactly as Pearson's correlation coefficient, and will thus inherit its range $-1 \leq \rho \leq 1$. The t test for significance can be used as well.

Analysis of variance

One-way analysis of variance (ANOVA) tests whether the means of the groups formed by values of the dependent variable are different enough not to have occurred by chance. The standard ANOVA assumes that the independent variable is normally distributed in each category of the dependent variable. Second, the independent variable should have the same variance in each category of the dependent variable. We have seen that this first assumption is heavily violated. A more appropriate ANOVA variant for this type of data is the non-parametric *Kruskal-Wallis H-test*.

First, the elements of variable $\mathbf{x}$ are ranked. For each of the $i = 1, \ldots, c$ classes, the sum of the ranks $\mathcal{R}_i$ is calculated. Kruskal-Wallis' H is now given by:

$$H = \frac{12}{n(n+1)} \sum_{i=1}^{c} \frac{\mathcal{R}_i^2}{n_i} - 3(n+1),$$

where n_i is the number of samples in class i. This statistic approximates a $\mathcal{X}^2$ distribution with $c - 1$ degrees of freedom if the null hypothesis of equal populations is true.

Table 4.6 — Univariate predictive power of the variables

Variable	Sign	Pearson PD	Pearson logit(PD)	Spearman rank	ANOVA $\mathcal{X}^2$	Reason for exclusion
R01	–	-9.9%**	-18.7%**	-21.0%**	56**	
R02	–	-23.5%**	-26.1%**	-30.5%**	122**	
R03	0	+1.1%	+0.5%	+0.5%	8	ANOVA value
R04	–	-26.5%**	-32.7%**	-32.8%**	147**	NE with R12
R05	–	-14.6%**	-20.1%**	-19.7%**	69**	NE with P05
R06	–	-14.2%**	-21.3%**	-23.0%**	72**	NE with P05
R07	+	+3.5%	+4.1%	+3.3%	8	ANOVA value
R08	+	+2.0%	+3.7%	+1.0%	12	ANOVA value
R09	–	+4.7%	+3.7%	+4.4%	8	unexp. sign
R10	+	+0.6%	+11.1%**	+18.7%**	63**	NE with P10
R11	–	-26.8%**	-31.9%**	-27.9%**	108**	NE with P11
R12	–	-30.3%**	-44.9%**	-44.0%**	252**	
R13	+	+9.0%**	+24.2%**	+26.8%**	103**	NE with P13
R14	–	-14.5%**	-21.2%**	-24.2%**	80**	NE with P14
P01	–	-7.7%*	-15.5%**	-16.6%**	36**	NE with R01
P02	–	-20.8%**	-27.9%**	-27.1%**	99**	NE with R02
P03	–	+0.2%	-0.5%	-0.3%	10	ANOVA value
P04	–	-22.4%**	-30.4%**	-29.5%**	122**	NE with R12
P05	–	-16.8%**	-22.4%**	-21.1%**	68**	
P06	–	-14.9%**	-18.8%**	-17.6%**	49**	NE with P05
P07	–	-4.7%	-7.0%*	-7.3%*	15	ANOVA value
P08	–	-5.8%	-1.5%	+1.1%	17	ANOVA value
P09	–	+3.8%	+8.2%*	+9.0%**	16	unexp. sign
P10	–	-5.1%	-19.0%**	-20.8%**	81**	
P11	–	-26.5%**	-33.3%**	-30.3%**	130**	
P12	–	-28.9%**	-42.6%**	-40.7%**	208**	NE with R12
P13	–	-14.9%**	-28.5%**	-26.4%**	97**	
P14	–	-15.4%**	-23.3%**	-24.5%**	80**	
T01	–	-9.3%**	-9.0%**	-9.5%**	25*	
T02	–	-7.0%*	-10.7%**	-11.0%**	20	ANOVA value
T03	0	-0.0%	+0.9%	+0.0%	4	ANOVA value
T04	–	-7.4%*	-9.0%**	-14.0%**	29**	
T05	–	-4.0%	-7.5%*	-10.7%**	35**	NE with T06
T06	–	-5.5%	-10.1%**	-11.1%**	30**	
T07	+	-8.2%**	-2.6%	+2.6%	14	unexp. sign
T08	+	-2.4%	-0.2%	-0.2%	11	ANOVA value
T09	–	-2.9%	+0.5%	+1.9%	22*	unexp. sign
T10	+	-1.6%	-0.2%	+7.7%*	26*	unexp. sign
T11	–	-14.9%**	-13.7%**	-10.5%**	36**	
T12	–	-6.5%*	-19.9%**	-22.1%**	71**	
T13	+	+8.7%**	+9.5%**	+10.9%**	24*	
T14	–	-4.3%	-5.5%	-6.2%	17	ANOVA value
V01	+	+10.9%**	+15.3%**	+17.2%**	41**	
V02	+	+13.0%**	+23.4%**	+28.2%**	95**	NE with V04
V03	+	+3.9%	+8.0%*	+9.5%**	22	ANOVA value
V04	+	+14.1%**	+23.6%**	+27.8%**	92**	
V05	+	+6.0%	+4.2%	+1.4%	18	ANOVA value
V06	+	+6.3%*	+5.2%	+3.8%	13	ANOVA value
V07	+	+10.2%**	+13.0%**	+18.0%**	49**	NE with V08
V08	+	+12.1%**	+15.1%**	+15.4%**	38**	
V09	+	+10.7%**	+11.4%**	+10.6%**	34**	NE with V08
V10	+	+1.9%	+1.5%	+1.7%	10	ANOVA value
V11	+	+13.1%**	+17.9%**	+21.6%**	60**	
V12	+	+10.3%**	+18.8%**	+18.6%**	52**	NE with V04
V13	+	+9.5%**	+11.9%**	+12.5%**	31**	NE with V04
V14	+	+11.9%**	+17.0%**	+17.5%**	44**	
S01	–	-11.1%**	-5.6%	-5.0%	20	ANOVA value
S02	–	-9.6%**	-5.2%	-4.2%	18	ANOVA value
M01	–	-9.2%**	-14.3%**	-13.5%**	46**	
M02	–	-8.3%**	-17.7%**	-18.3%**	49**	
M03	–	-7.8%**	-15.1%**	-16.3%**	38**	
Q01	–	-	-	-29.5%**	117**	
Q02	–	-	-	-6.2%*	17	ANOVA value
Q03	–	-	-	-4.0%	12	ANOVA value
Q04	–	-	-	-15.3%**	35**	
Q05	–	-	-	-17.7%**	59**	
Q06	–	-	-	-18.4%**	56**	
Q07	–	-	-	-9.2%**	16	ANOVA value
Q08	–	-	-	-17.6%**	47**	

*: 5% significance, **: 1% significance, grey variable : selected for reduced data set, NE: nearly equivalent

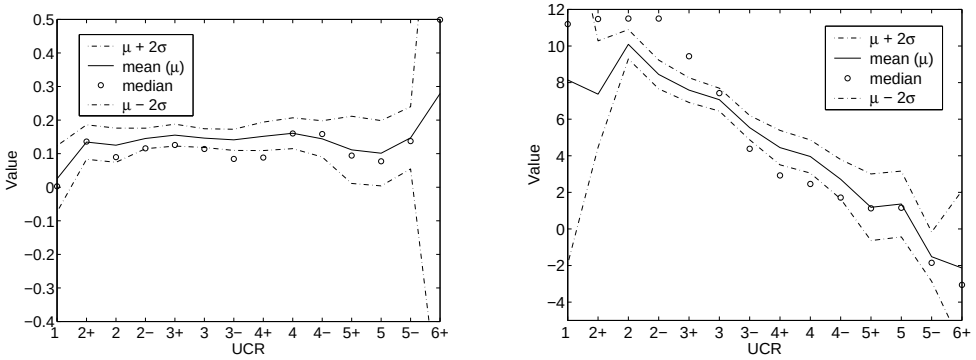
Discussion

The results of both the correlation metrics and the ANOVA analysis are given in table 4.6. The second entry in the table is the expected sign of the correlation, based on section 4.1. Volatilities should be positively correlated to the probability of default, whereas macro-economic scores, size variables, and answers to subjective questions should always have a negative coefficient. Raw values and trends should always share the same sign, but the sign itself depends on the ratio. Note that due to its U-shape we have an expectation of zero for the trend and raw value of *Annual Turnover Growth*.

The third, fourth and fifth column show Pearson's and Spearman's correlation coefficients. The asterisk characters indicate whether the derived coefficients are significant, by testing the null hypothesis that there is no correlation among the variables. Spearman's coefficients are systematically higher than the Pearson's. This is presumably caused by non-linear relationships and by the replacement of outliers with maximum and minimum values before calculating Pearson's correlation coefficient. The sixth column gives the $\mathcal{X}^2$ value of Kruskal-Wallis ANOVA.

When the correlation coefficients are approximately zero, but the ANOVA value indicates that the group means are significantly different, the variable might be interesting after all. The *Annual Turnover Growth* is expected to meet these assumptions due to its hypothesised U-shape. A graphical representation of the univariate relation might provide more insight. The mean, median, and standard deviation are calculated for each of the fourteen UCR classes. The results for *Annual Turnover Growth* are shown in figure 4.6. For comparison *Interest Coverage Ratio* has been included as well. The solid line connects the mean values per UCR class; the dots represent the median values; and the dashed lines indicate the 95% confidence intervals. We can conclude from figure 4.6a that there is no direct relation between *Annual Turnover Growth* and a counterparty's UCR.

One observation from these and other graphs is the large 95% confidence intervals around the extreme UCR values. This is caused by the relatively small number of observations for these UCR classes. The estimated error around the mean decreases when the number of observations increases.



(a) *Annual Turnover Growth* (b) *Interest Coverage Ratio*

Figure 4.6 — Univariate relationship: mean, median, and confidence intervals

There are a number of other remarkable observations. Unexpected signs in both correlation coefficients exist for the raw and percentile values of *Creditor Days* and for several trends, implying that our hypothesis regarding these variables does not match our observations. The predictive value of variables with a 'wrong' sign is questionable.

4.4.3 Collinearity

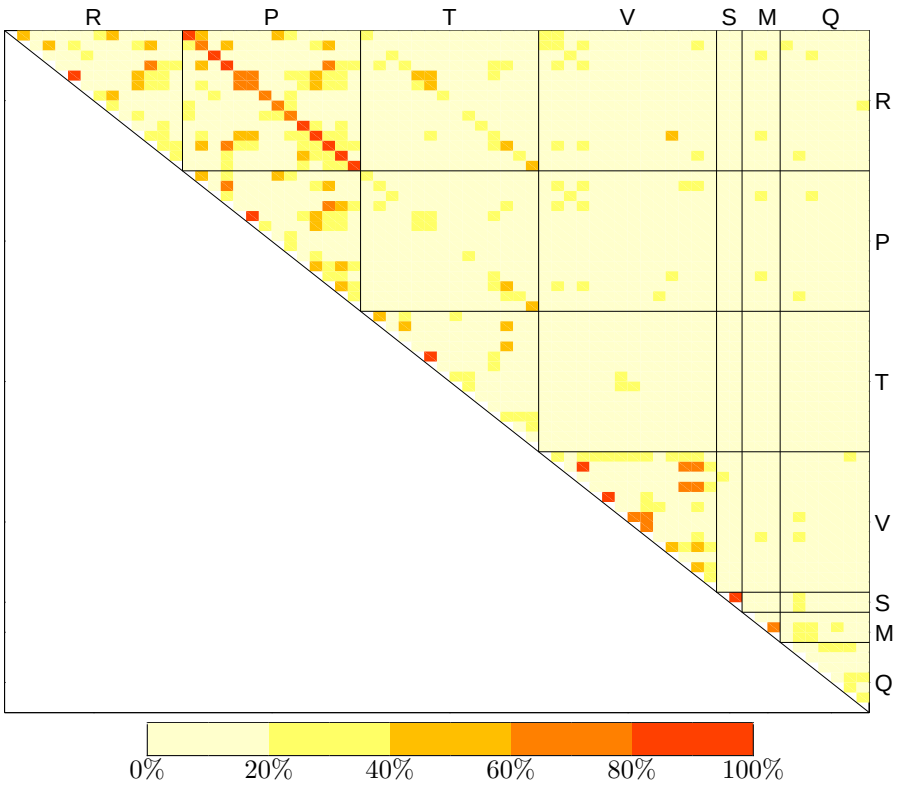
For many statistical models, variables should have as little correlation as possible. A table containing the correlations among all variables was constructed, where Pearson's r was used for correlations among metric variables, and Spearman's ρ for correlations between ordinal/ordinal and ordinal/metric combinations.[2]

Figure 4.7 gives a graphical representation of this table. Both the x and y axis represent the independent variables in the same order as table 4.5: starting with raw values R01 to R14, then percentile values P01 to P14, trends T01 to T14, volatilities V01 to V14, size variables S01 and S02, macro-economic scores M01, M02, and M03, and finally the answers to subjective questions Q01 to Q08. The (absolute) correlation coefficient between two variables is given by the colour of its corresponding cell.

First we will discuss the main aspects of this table. One of the first eye-catchers is the 'red' diagonal in the R,P square. This indicates that R1 is strongly correlated with P01; R02 with P02; all the way up to R14 with P14. These correlations were expected: percentile values are transformations of the original ratio values. The orange two-by-two square in the P,R block and the red spots in the R,R and P,P blocks have a common cause: R05 (*Current Ratio*) and R06 (*Quick Ratio*) are, not surprisingly, highly correlated because of their similar underlying formulae. We can see that their derivatives coincide as well.

More surprising is the substantial positive correlation between any ratio and its trend: higher ratio values coincide with steeper trend lines and vice versa. The cause is the relatively small number of ratios that is used to calculate the trend (three to four). A high or low one-off ratio value will have an equivalent effect on the trend. Percentiles versus trends evidently yields similar results. Finally, many of the volatilities are correlated with one another.

Table 4.7 — Collinearity among independent variables



[2]Note that the value of Spearman's and Pearson's coefficients do not have the same meaning.

Table 4.8 zooms in on one of the most interesting parts of the collinearity table: the ratio values. We have used the percentile values for easier comparison: if there would be any correlation between two variables, their correlation coefficient should be positive.

In section 4.2.1 we divided the ratios into four groups: operations, liquidity, capital structure, and cash flow and debt service. The groups are divided by the dotted lines. We would expect that the correlations among variables within a group are high. This holds for most of the variables, with a few exceptions. These exceptions coincide with our findings in the previous subsection, and concern *Annual Turnover Growth* (P01), *Debtor Days* (P07), *Stock Days* (P08), and *Creditor Days* (P09). Because we are working with percentile values, we would expect all positive correlation coefficients or perhaps slightly negative ones when they are uncorrelated. The table shows large negative values for the correlation of P01 with P08, P06 with P07, and nearly all percentiles with P09. The cause of the large negative correlation coefficients lies in the fact that they are all based on *Cost of Goods Sold* or *Turnover* (cf. appendix A.2). This variable becomes zero in case of service industries, and apparently has a large influence in the correlations.

On overall we can conclude that the collinearity among the variables is substantial. We assume that the removal of several variables from the data set will have a positive influence on the performance of statistical and artificial intelligence techniques. The variable selection/removal procedure is described in the next section.

Table 4.8 — Collinearity among the percentiles

	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	P13	P14
P01		44%	-5%	21%	9%	10%	-12%	-40%	30%	2%	10%	13%	10%	10%
P02			12%	62%	19%	17%	1%	-15%	3%	7%	24%	51%	18%	11%
P03				19%	-4%	-7%	15%	16%	-15%	-2%	2%	13%	5%	6%
P04					14%	14%	9%	12%	-7%	14%	8%	67%	42%	24%
P05						85%	-7%	-11%	-15%	25%	51%	31%	22%	4%
P06							-20%	12%	-8%	16%	40%	27%	21%	3%
P07								15%	-32%	12%	7%	10%	8%	7%
P08									-29%	4%	-6%	11%	11%	8%
P09										-16%	-21%	-14%	-11%	-2%
P10											40%	24%	43%	21%
P11												31%	31%	11%
P12													42%	31%
P13														23%
P14														

4.5 Conclusion and variable selection

We have seen that many variables indeed have predictive value to determine the UCR of a counterparty; we have reached correlations up to 45%. In the next chapters, we will compare several techniques to the existing MRA system. This system uses the complete data set, where all variables have been converted into scores. This data set will therefore be used for comparison. Many statistical and artificial intelligence techniques however are negatively influenced by collinearity and variables that have no predictive power. A reduced data set might improve the results of these techniques.

Another question is whether the information incorporated in the scores is actually useful. All research peers have achieved reasonable to good results based on (standardised) raw values, thus without expert knowledge. Two extra data sets are therefore added to our research: the complete raw data set, and a reduced one.

The procedure to reduce both the raw values data set and the scores data set is as follows. A first set of variables is created based on the following criteria:

- the variable is significantly[3] correlated according to either Pearson or Spearman
- the sign of the correlation coefficient matches the expected sign
- the variable has significantly[3] different group means according to the ANOVA value

When only the last criterion applies, the variable is plotted against the UCR to determine a possible non-monotone relationship (cf. figure 4.6).

A final requirement is (relative) independence with respect to the other variables. When two variables are highly correlated, the one that has the least predictive power according to the Kruskal-Wallis ANOVA value will be deleted. We have experimented with different settings for 'highly correlated': the results were compared to the optimised input sets of several statistical techniques (cf. section 8.2). We have concluded that 50% leads to data sets that are comparable to many optimised sets. Highly correlated macro-economic scores and answers to subjective questions will not be removed from the data set, as experiments showed a negative impact on the performance.

Table 4.9 — Final data sets

Description	Size	Variables
Full data set	69	All inputs
Full scores set	62	All inputs
Reduced data set	27	R01, R02, R12, P05, P10, P11, P13, P14, T01, T04, T06, T11, T12, T13, V01, V04, V08, V11, V14, M01, M02, M03, S01, S04, S05, S06, S08
Reduced scores set	28	R05, R10, R12, R13, P01, P11, P14, T01, T04, T06, T10, T11, T12, T13, V01, V02, V03, V07, V11, V14, M01, M02, M03, S01, S04, S05, S06, S08

In the end there are four different data sets on which the tests will be performed: two complete sets and two reduced sets. Table 4.6 in the correlations subsection has highlighted the selected variables in grey, and included the reason for removal from the data set. Table 4.9 gives a summary and enables comparison with the reduced scores data set.

We can see that the reduced data set and the reduced scores set largely consist of the same variables and are almost equal in size. This might indicate that the impact of adding expert knowledge is rather small. We will come back to this issue in chapter 8, when we discuss the results of applying different techniques on the two data sets.

[3] Significance level of 5%

Chapter 5

New rating framework

The system that is currently in place at ABN AMRO is an expert system called Moody's Risk Advisor (MRA). The first section discusses this system at a high level. We will see that the implementation is lacking in many respects. Two conclusions can be drawn from this section. There is a demand for a new rating system based on the current model. Many of the drawbacks that are listed can hence be solved. More importantly, we should examine how the current model performs in comparison to proven mathematical techniques. Both suggestions will be addressed in the next chapters.

The second section of this chapter describes our new design of a generic rating framework for corporate counterparties. This framework will serve as a starting point for the development of multiple kinds of rating models. The third section will give a flavour of the implementation details. The interested reader should refer to Dikkers (2005b) for further details. In the evaluation section we will see that our newly developed framework solves many of the disadvantages that were present in MRA.

As aforementioned, the presented framework is merely the starting point for the implementation of different rating models. An improved version of MRA that is implemented in this framework is presented in chapter 6. Possible extensions with statistical models are addressed in chapter 7 and 8.

5.1 Current system analysis

With the combination of MFA and MRA, Moody's KMV has provided a robust system for creditworthiness assessment. MRA runs centrally on a Citrix server. The input data combined with a derived UCR is stored on this server for future analysis. We do not intend to describe the MRA system itself. The interested reader can consult documentation of Moody's KMV (2002, 2004) or (Kumra et al. 2000). A description of the model will be given in chapter 6, where we present our own implementation of the MRA model. Our analysis aims at finding the drawbacks of the MRA system, which are related to the following aspects:

- Transparency
- Data storage
- Response speed
- Flexibility
- Prediction accuracy

5.1.1 Transparency

At the start of our project, there was no detailed documentation concerning the characteristics of MRA. The lack of documentation resulted in the fact that not only users regarded MRA as a grey

box, but ABN AMRO's model owners as well. The latter party is responsible for the tuning of the model parameters and the calibration of the model, but was unable to perform this task properly without the full knowledge of the model. A first attempt had been made within ABN AMRO to replicate the MRA system in MS Excel in 2002, but this system did not predict the same UCRs as the original system. The transparency problem thus remains relevant.

5.1.2 Data storage

When the rating of a counterparty is saved, all data that is used to derive this UCR needs to be stored; both for internal and Basel-II purposes. The MRA database, however, proved to be incomplete. For instance several financial ratios were not saved at all, and the financial statements lacked unique identifiers. We have performed an extensive research to ensure the storage of all values. The resulting recommendations can be found in our internal report (Dikkers and Quere 2004). These recommendations have been implemented in July and August 2004. This research has resulted in the bank being Basel-II compliant with respect to the input data for the wholesale portfolio.

We have been able to recalculate most of the missing data, but as we have seen in section 4.3.2 some inputs remain missing. Several hundreds of observations before August 2004 have therefore become unusable for further research, but data quality is guaranteed as from August 2004.

5.1.3 Response speed

It takes a few seconds to open and rate a counterparty in MRA. This is an acceptable amount of time for account managers, but is problematic when it comes to rating multiple counterparties. Research on the impact of model changes requires a batch run on a complete portfolio of typically 10,000 counterparties. Not only ABN AMRO's model owners require this feature. Many departments are interested in *stress testing*, i.e., running the MRA system for different future scenarios. Applications can be found in the industrial sector or for country research, or in stressing the robustness of the calculated economic capital.

The main cause of the slow processing is evident: MRA was originally designed as a production system on a per-counterparty basis (Duda et al. 1987). The functionality to save data for one counterparty into a database was later added by Moody's KMV. Batch processing is performed by automatically opening, saving, and closing each counterparty record in MRA. A test run on approximately 10,000 counterparties took over twelve hours.

5.1.4 Flexibility

MRA is proprietary software of Moody's KMV. This gives several advantages, such as maintenance and support, but has some drawbacks as well. The software is very expensive to start with. The main problem however lies in the inability of ABN AMRO to experiment with model changes, such as adding or removing a ratio. Model changes have to be tested by Moody's KMV at high costs and with long response times, where the impact of the proposed changes is still questionable.

5.1.5 Prediction accuracy

The prediction accuracy of MRA on the cleaned data from January 2004 until December 2004 is 43%. Without benchmarks we cannot conclude whether this achievement is good or bad. For this means we will implement statistical and artificial intelligence techniques. The results of this comparison can be found in chapter 8.

5.1.6 Conclusion

The disadvantages of the current implementation are substantial. We have solved the data storage problems, but the other aspects cannot be solved without taking matters into our own hands.

Moreover, due to these problems, a demand for a new rating system has emerged within ABN AMRO in the past few years.

For this means we will develop a new generic rating framework in chapter 5. Within this framework, we will be able to incorporate both a new version of MRA[1] and several statistical techniques. The former addition is presented in chapter 6, whereas the statistical techniques are addressed in chapter 7 and 8.

5.2 New framework design

5.2.1 Audience

We have identified three different types of applications for the new credit rating program:

- Parameter tuning, performed by the Credit Ratings and Portfolio Management department
- Stress testing, performed by the Industrial Sector Research and Country Risk Management departments
- Counterparty rating for all account managers, i.e., replacement of MFA/MRA as production system

The first and second application pose similar requirements to the system. The rating system needs to determine the UCRs of all counterparties in a portfolio based on new parameters, new macro-economic scores, or forecasts of financial values. The portfolio used for parameter tuning typically contains about 10,000 counterparties, implying the need for a very efficient program. Both the input data, parameter data, and macro-economic data is provided through databases. The output will be a database as well, containing the derived UCR per counterparty. Analysts can assess the impact of the aforementioned renewed values, for instance by creating migration matrices that compare the counterparties' present UCRs with the UCR in the new situation, or by computing statistics like concordance measures with shadow ratings (cf. section 8.1).

The third application is merely a possible future application of the system. The replacement of MFA/MRA has not been taken into account in the design phase; the focus has been on the requirements posed by the first two applications. We will, however, discuss some implications of the production system as well.

5.2.2 Requirements

The system should be able to accept data in two ways. The first way is to feed the data of many counterparties, which can most easily be done by using databases. The second way is to enter financial statements, answer subjective questions, and provide other relevant data directly. The users are content with the user interface of the present spreading tool MFA. There is therefore no need to implement a new user interface for the spreading part at this point in time. Moreover, when MFA/MRA is replaced, the new system will be integrated into the *Generic Rating* ABN AMRO *Counterparty Engine*, or GRACE. This engine has standard user interfaces for data entry.

The requirements therefore focus on the characteristics of the core engine:

- All input values are stored
- All intermediate output values are stored
- The response time when running the program does not exceed two minutes on a standard ABN AMRO workstation for a typical portfolio of 10,000 counterparties
- The system is properly documented
- The model owner is able to change model parameters very easily
- The model owner is able to change the model structure reasonably easily
- The model owner is able to add other than existing rating techniques reasonably easily

[1] With permission of Moody's KMV under certain conditions that have been posed to ABN AMRO.

5.2.3 Model

There are two different sources of information. First, the data can originate from the spreading tool MFA. A user can enter the financial statements of one or more counterparties in MFA. All fields are stored in the underlying MFA database. Financial ratios have not yet been derived. We will utilise the user interface of MFA, and only implement a procedure to retrieve the data from the underlying database and calculate the financial ratios. The other source is a historical database named *Internal Ratings Database* (IRD) that contains counterparties that have been rated in the past. This database contains all model inputs, and we therefore only have to implement a smart copy and paste procedure to retrieve the correct data.

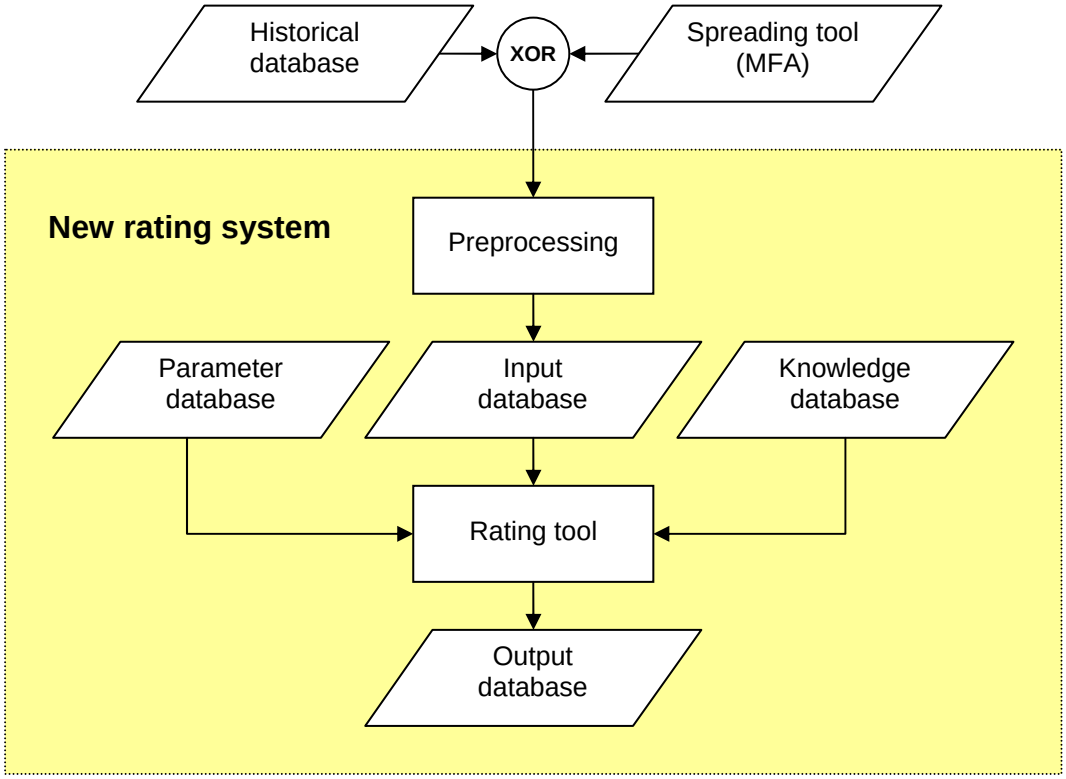


Figure 5.1 — Model

We want to be able to read from any of the two sources. Since the structures of both source databases change regularly, we will first convert the values of these databases into a common format in the preprocessing step. For this means we have developed a database that contains *all and only* the relevant inputs: the *input database*. Data quality in the input database will be guaranteed by this preprocessing step.

Most rating techniques have associated model parameters that might change over time. Since the parameters are completely independent from the data, we have implemented a different database called the *parameter database*. The structure of this database depends on the rating technique that is used. MRA, for instance, uses hundreds of parameters that are complexly divided over several tables, whereas linear regression only requires the beta coefficients to be known.

The last type of input is the macro-economic scores. Strictly spoken these are neither counterparty-dependent data nor model parameters. A third database is therefore used to store the country and industry scores: the *knowledge database*. The quartile values of the different ratios for the different industries are stored in this database as well.

The three databases all serve as inputs for the centre of our rating system: the *rating tool*.

The rating tool implements one or more rating models. In the next chapter, we will present our improved implementation of the MRA model. Statistical techniques can be added as well, as can be read in chapter 7. The rating tool can thus be seen as the core and most important aspect of our rating system.

When the calculations of the rating tool have been performed, the UCR will be stored into the *output database*. This database is used to store all intermediate values that are considered useful as well. This concludes data flow through the rating system.

5.3 Implementation

5.3.1 Choice of language

We have chosen to separate the preprocessing step from the core rating tool for several reasons. The preprocessing step can appear in many different shapes. We will implement procedures for both the MFA and the IRD database, but others might be included as well. A stress testing application might for instance use the core of the rating tool, but needs to populate the input database directly instead of through the MFA or IRD database. Secondly, the structure of both the MFA and the IRD database changes regularly, and so does the procedure to populate the input database.

The rating tool itself, however, rarely changes. Only when the underlying model is changed, this core needs updating. It does, however, need to be very fast. For tuning purposes for instance, the model owner changes a few parameters, runs the rating tool on a large portfolio of counterparties, and reviews the results. This implies the need for a fast and robust programming language.

It should be possible to have multiple rating methods that are based on the same data in our rating tool. For this means it is common practice to separate the data from the functions. This separation of data and rating methodology is best implemented using the object-oriented programming paradigm. Object-oriented programming has several advantages. First, the code is easier to maintain due to abstraction. Only the essential features of data are accessible from other classes, without the inclusion of background details. If the underlying data representation changes, we do not need to alter functions that are based on this data. Another advantage is code reusability. We can easily add additional features to an existing class without modifying it.

We have chosen to use C++ for the rating tool itself. C++ is a fast, robust, and well developed programming language that utilises the object-oriented paradigm. Moreover, C++ was a language that was already used by some of ABN AMRO's model owners. The use of C++ would simplify the transfer of knowledge, creating a more sustainable situation after completion of the project. The rating tool has been developed using MS Visual Studio C++ 6.0.

As mentioned above, the preprocessing part needs regular updating. On the other hand, the speed is not of the same pivotal importance as for the core rating program. We have therefore chosen to use Visual Basic for Applications (VBA) to implement the preprocessing procedures. Nearly all of ABN AMRO's model owners are familiar with this language. A larger flexibility is thus ensured. All databases have been implemented in MS Access, and the preprocessing program has been built in VBA within these databases.

5.3.2 Generic rating tool

In this section we will give a taste of the implementation of the new rating system. The designs of the different databases, the VBA procedures to populate these databases, and the user interface have been left out of the report. These details can be found in Dikkers (2005b).

It should be said that the user interface that has been presented is meant to be a starting point for stress testing and tuning applications. The way to stress and tune has been left out of scope of this project. At the time of writing, both a tuning application and a prototype stress testing application have been developed by ABN AMRO colleagues based on our rating tool.

We will now move to the implementation details. We do not attempt to give an exhaustive description of the model, but merely point out the most interesting parts of our generic rating tool.

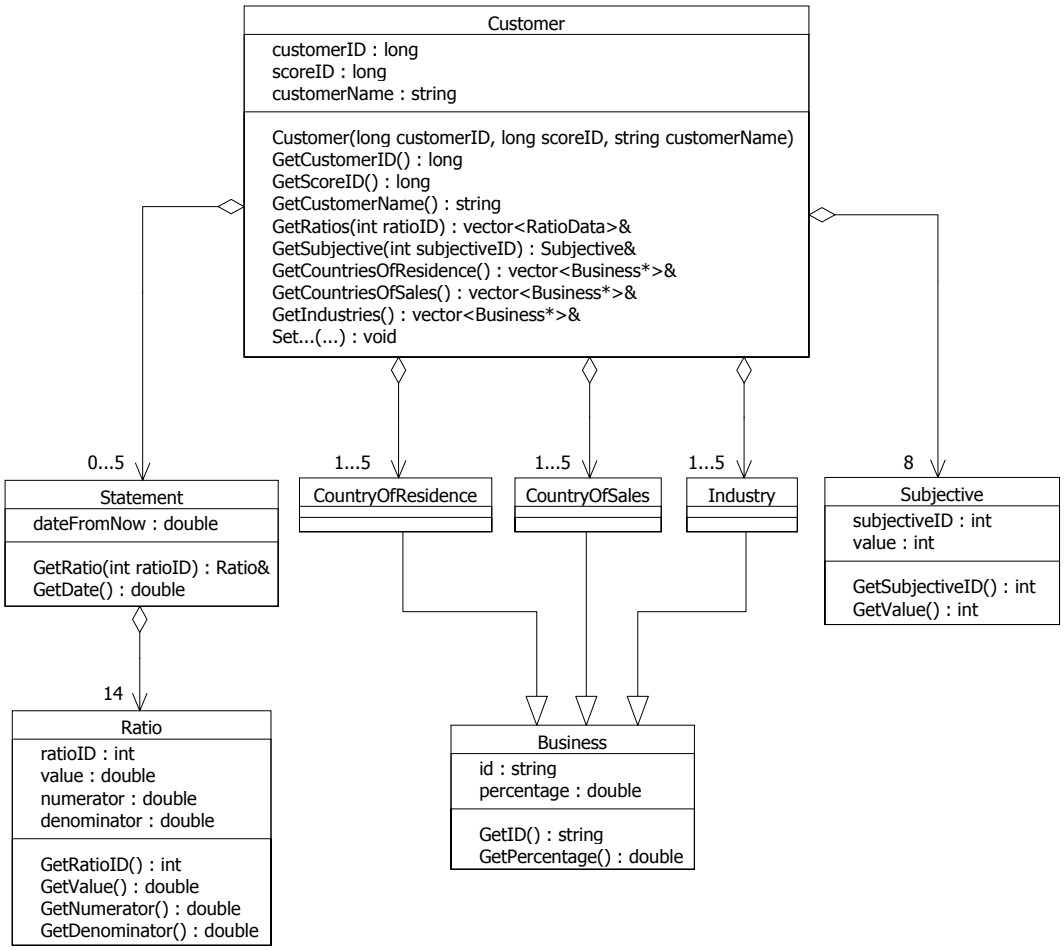


Figure 5.2 — UML diagram of class Customer

The class Customer[2] characterises a counterparty at a specific point in time. Figure 5.2 shows the corresponding UML diagram. Each Customer is described by up to five objects of the class Statement, where each Statement consists of fourteen Ratios.[3] Another important aspect of a Statement is its date, which is used in trend and volatility calculations.

Besides the financial statements, eight subjective questions (Subjectives) are answered. Finally, a counterparty is located in up to five CountryOfResidence, CountryOfSales, and Industry objects, which are all derived from the class Business. The percentages for each type of Business will sum up to 1, e.g., a counterparty is located in the Netherlands for 60% and in Belgium for 40%.

The characteristics of a Customer can be accessed by means of *get* functions. GetSubjective takes the id of the subjective question as argument and returns the answer that an account manager has given to this question. GetCountriesOfResidence returns an unmodifiable vector with up to five pointers to the CountryOfResidence the counterparty is located in; GetCountriesSales and

[2]The class names are given in the sans serif font.

[3]All numbers that are given in this section indicate the current configuration. Due to the object-oriented nature changing these numbers only requires one or two small amendments in the source code.

GetCountriesIndustries are similar.

More complicated is the GetRatios function. This function provides the caller with all the information necessary to calculate the trend or volatility over the past few years. GetRatios takes the id of the financial ratio as an argument, and returns an unmodifiable vector of RatioData elements. The structure RatioData consists of a Ratio and its date. This provides us with all the necessary data to perform trend or volatility calculations.

A helicopter overview of the complete rating tool is depicted in figure 5.3. The separation between counterparty data, model parameters, and macro-economic scores, which are provided by three different databases, is present in our class diagram as well. The model parameters are presented in an object of class Parameters, and the country and industry scores in an instance of KnowledgeBase.

Now we would like to determine the UCR of a counterparty. For each Customer, a CustomerScore object is created. The UCR of the corresponding counterparty is derived upon creation of the CustomerScore object. These CustomerScores are collectively held in the wrapper class Scores.

5.4 Evaluation

We have proposed and implemented a framework that meets all the requirements that we have posed in section 5.2.2. The input and output databases store all values of interest. The procedure to read from a database can be easily adjusted by any person with VBA knowledge.

The rating models themselves will be implemented in the C++ core of the rating system: the rating tool. Amendments of these models or adding new models requires C++ knowledge and is less straightforward. The object-oriented nature of the program, however, does enable rapid prototyping of new rating models.

We have presented the complete credit rating framework in this chapter, including a generic rating tool. The design and implementation of rating techniques in the generic rating tool will be addressed in the next two chapters. We will postpone the assessment of the response speed requirement to section 6.3, since we assume that the calculations required for the model will be the bottleneck.

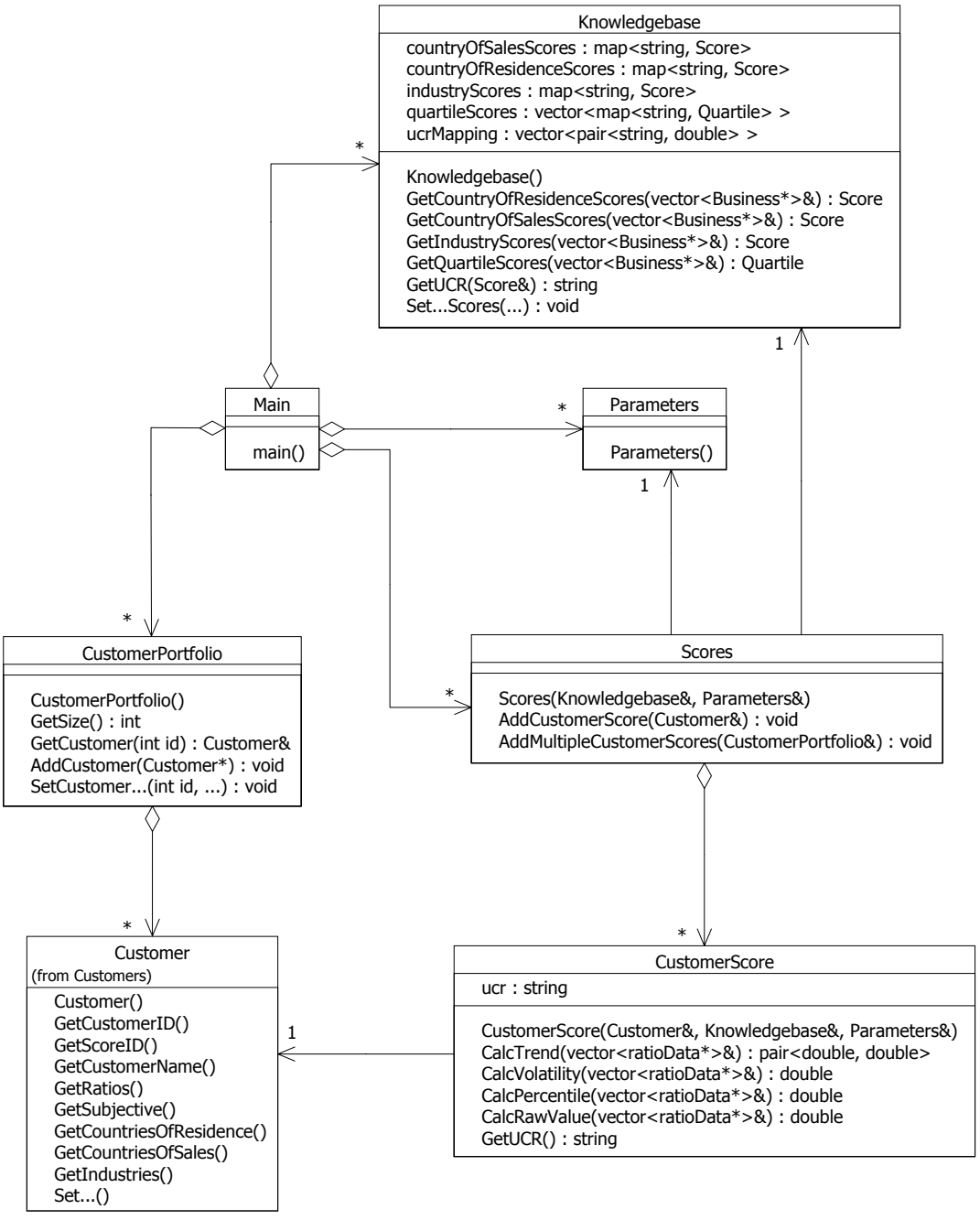


Figure 5.3 — UML diagram of generic rating tool structure

Chapter 6

Incorporating the MRA model

The mathematical foundations of the MRA system were not known up to this date. This implies that there is only little knowledge on its underlying assumptions and the violations of these assumptions. The first section discloses the exact mathematics behind the model and the assumptions it is based on.

For a complete understanding, we have redesigned the MRA model and incorporated it into the rating framework from the previous chapter. As has been described in section 5.1, transparency is not the only aspect of interest. The described model solves the other disadvantages of MRA as well, which are response speed and flexibility. We have named this specific implementation of the rating framework 'Quantitative Consultancy's MRA', or QC-MRA.

The evaluation of the model and its implementation are given in the third section. Several recommendations are given to improve the transparency of the model and better meet the assumptions from the first section.

6.1 Theory

The exact characteristics of MRA have so far been unknown to ABN AMRO model owners. The major problem lies in undocumented deviations from the standard model, both in the MRA implementation and in ABN AMRO tailored version. We initiated extensive research to resolve this problem. We have interviewed MRA experts at Moody's KMV headquarters in London (Stark 2004) and have had e-mail conversations with Syntel authors Risch and Duda (Duda 2004). Our main source, however, is our own implementation of MRA, which is described in the next section. We have gradually come to a very close replica of MRA with respect to the outcome by developing a model, comparing the results, changing the model, comparing again, and so on. Many of the presented results were later incorporated in a first and second version of the MRA documentation that is provided by Moody's KMV (2004).

6.1.1 Tree structure

MRA is based on the Syntel language, which is a language for designing decision networks based on user input. This language was developed in the 1980s by Duda et al. (Duda et al. 1987; Risch et al. 1988). We will only focus on the language aspects that have been used in the generic MRA version and leave the remainder out of the scope of this report.

Any system built in Syntel can be viewed as a directed acyclic graph, or tree, with nodes as atomic elements. The leaves of the tree are formed by the independent variables, i.e., the model inputs. The model inputs can be either numerical, such as raw ratio values, or categorical, such as answers to subjective questions.

The dependent variables (non-leaf nodes) are called *assessments*, and indicate the quality of the node. The parent node of a raw ratio value for instance indicates the quality of this raw

ratio, whereas higher up the tree we can find assessments of business concepts like *Operations* or *Liquidity*. The root node of the tree is the final output, and indicates the quality of the counterparty. In case of the ABN AMRO system, this root assessment will be translated into a UCR.

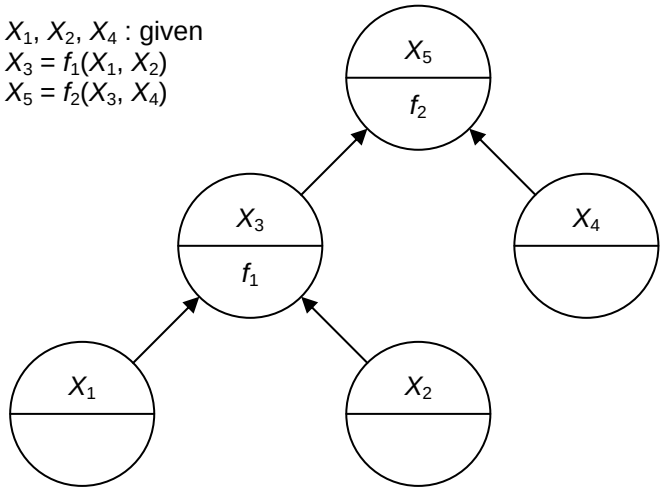


Figure 6.1 — Basic structure of a Syntel system

Any dependent node Y is a function of its underlying variables: $Y = f(X_1, X_2, \ldots, X_n)$. A graphical representation is given in figure 6.1. Most expert systems use *if-then* production rules to infer the value of a dependent node from its children. The function $f(\cdot)$ is then composed of possibly many of these production rules. These rules are however poorly suited to express relations between continuous independents and a continuous parent variable. A more natural way to express the relation between the children and the parent is to *add votes* (Duda 2004). In the MRA case, the parent has a default value $v_0 = 50$, and each independent child can positively or negatively change this value based on its own value. This is the *weighted voting function*. The weighted voting function $f(\cdot)$ first maps the children into a common continuum called vote space and then sums them:

$$Y = f(X_1, X_2, \ldots, X_n) = v_0 + v_1(X_1) + v_2(X_2) + \ldots + v_n(X_n) \tag{6.1}$$

For each dependent node, the voting functions $v_i(\cdot)$ are specified by the knowledge engineer. As we will see in the next subsection, the summation assumes independence among the individual X_i. In the special case that all voting functions $v_i(\cdot)$ are linear, $f(\cdot)$ behaves like a linear regression function. This is, however, not required.

6.1.2 Inexact reasoning

In MRA each node has an associated probability distribution. The goal of these probability distributions is two-fold: to handle missing data, and to incorporate imperfect knowledge. The introduction of probability theory considerably complicates the evaluation of function $f(\cdot)$. The value of each dependent variable now is a probability distribution that depends on the distributions of its arguments X_i. Because we have assumed that the X_i are statistically independent, the joint probability distribution is merely the product of the distributions of the arguments:

$$P(y) = \iint \cdots \int v_0(P_0(x_0))v_1(P_1(x_1)) \cdots v_n(P_n(x_n))\delta((x_0 + x_1 + \ldots + x_n) - y)dx_0dx_1 \cdots dx_n, \tag{6.2}$$

where $P_i(x_i)$ and $P(y)$ are the probability density functions of variates X_i and Y respectively, and $\delta(\cdot)$ is a delta function. Note that compared to equation 6.1, the default value v_0 has become

a variate instead of a constant. MRA assumes that the default value for each node is normally distributed with mean 50 and standard deviation 3: $v_0(P_0(x_0)) \sim \mathcal{N}(50, 3^2)$.

At the time this model was developed, it was too costly to perform this type of calculations throughout the tree. Therefore two additional assumptions were made:

- each variate X_i, defined by $P_i(x_i)$, is normally distributed;
- the variate mapped onto vote space $v_i(X_i)$ is normally distributed as well.

This latter assumption only holds if the first assumption is met and $v_i(\cdot)$ is a linear function. If $v_i(\cdot)$ is non-linear, e.g., logarithmic, $v_i(X_i)$ is no longer normally distributed. When the normality assumptions are met, each mapped variate $v_i(X_i)$ can be represented by $v_i(X_i) \sim \mathcal{N}(\mu_i, \sigma_i^2)$, where

$$\mu_i = \int v_i(P_i(x_i))dx_i$$
$$\sigma_i^2 = \int P_i(x_i)(v_i(x_i) - \mu_i)^2 dx_i$$

Now we can utilise the fact that the sum of n normally distributed variates $v_1(X_1) + v_2(X_2) + \ldots + v_n(X_n)$ with means and variances $(\mu_1, \sigma_1^2), (\mu_2, \sigma_2^2), \ldots, (\mu_n, \sigma_n^2)$ respectively is another normal distribution. Equation 6.2 then simplifies to $Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$, where

$$\mu_Y = \mu_1 + \mu_2 + \ldots + \mu_n \tag{6.3}$$
$$\sigma_Y^2 = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_n^2 \tag{6.4}$$

6.1.3 Soft saturation

The newly calculated variate Y is on a scale $\langle -\infty, \infty \rangle$. MRA then reduces the range of Y to $\langle 0, 100 \rangle$. Users that do not have knowledge of probability theory can now just focus on the mean value of the assessment, which always lies between 0 and 100. A second advantage is that none of the assessments can become an outlier. The following S-shaped transformation function is applied to variate Y:

$$S(x) = 100 \cdot \frac{1}{1 + \exp(\frac{-(x-50)}{25})} \tag{6.5}$$

This function is referred to as the *soft saturation function*. It takes the same form as the logistic distribution function, which implies that the derived variate $Y' = S(Y)$ is now assumed to come from the logistic distribution. When Y' is used as input for a higher level assessment, however, it is treated as any other input and is thus assumed to be normally distributed again.

6.1.4 Discrete character

Either the complete probability distribution or the distribution parameters $\mu_{Y'}$ and $\sigma_{Y'}^2$ can be used as inputs higher up the tree. MRA authors have chosen for the former, but have discretised the distribution as follows.

Recall that each node is described by a variate Y' that has a probability distribution on the scale $[0, 100]$. Instead of saving the complete probability distribution, the range is divided into seven equally sized intervals of which the area is stored. We can calculate the area a under the probability function of variate Y' for each bucket j with interval $[b_{j-1}, b_j\rangle$:

$$a_j = \int_{b_{j-1}}^{b_j} S(P(y))dy, \qquad j = 1, \ldots, 7, \quad b_i = \frac{100i}{7},$$

where $S(\cdot)$ is the soft saturation function and $P(y)$ is the probability distribution function of variate Y. Each node can now be seen as a probability distribution histogram. Evidently, all the a_j sum up to one.

One advantage of this approach is that the mappings onto vote space $v_i(\cdot)$ can be amended accordingly. Instead of defining a smooth continuous function, we can define a step function that takes one value on the complete interval $[b_{j-1}, b_j\rangle$. We thus only have to assign a function value for each of the seven buckets.

An example illustrates this point, where we have omitted variable index i to improve readability. Suppose we want to find the discretised version of the simple linear function $v(X) = 0.5X - 25$. In the discrete form, this function will become a vector multiplication

$$v'(\mathbf{a}) = \mathbf{w}^T\mathbf{a},$$

where both $\mathbf{w}$ and $\mathbf{a}$ are 7 by 1 vectors. Each of the seven weights[1] w_j can be derived by taking the average value of the original function $v(X)$ on the interval $[b_{j-1}, b_j\rangle$. In case of a linear function this simplifies to the value of the midpoint of this interval: $w_j = v(\frac{1}{2}(b_{j-1} + b_j))$. In our example, $w_1 \simeq v(7.14) = -21.43$, $w_2 \simeq v(21.43) = -14.28$, $w_3 \simeq v(35.71) = -7.14$, etc. Note that the distance between the w_j are all equal, which holds for all linear functions. The reverse holds as well: if the weights are ordinal and equally distanced, the function $v'(\cdot)$ can be regarded as a discrete variant of a linear function.

Knowledge engineers will thus have to provide seven values on each level of the tree. There are however no restrictions on how to choose these values. This implies there are seven degrees of freedom for each voting function, i.e., for each node of the tree.

6.1.5 Missing data

With respect to missing data, the Syntel language deviates from the procedure as has been described in subsection 4.3.2. There are two possible ways to handle undefined or missing X_i. The first way is to ignore the undefined node. Y is now formed by the distributions of remaining X_i. When all inputs of the parent node (besides the default value that is always available) are missing, the parent node will be undefined as well.

The use of prior probabilities is another way to handle missing values. A weighted voting function $v_i(\cdot)$ typically has a minimum and a maximum value it can take, i.e., its range is $[\min_{v_i(\cdot)}, \max_{v_i(\cdot)}]$. When the distribution of X_i is unknown, its value in vote space can take any value in this range. The distribution of $v_i(X_i)$ is therefore defined as uniformly distributed: $v_i(X_i) \sim \mathcal{U}(\min_{v_i(\cdot)}, \max_{v_i(\cdot)})$. To include this distribution in its parent assessment, we will have to assume normality again. The mean and variance of the uniform distribution are to be the parameters of the normal distribution:

$$\mu_i = \frac{1}{2}(\min_{v_i(\cdot)} + \max_{v_i(\cdot)}) \tag{6.6}$$

$$\sigma_i^2 = \frac{1}{12}(\max_{v_i(\cdot)} - \min_{v_i(\cdot)})^2 \tag{6.7}$$

Suppose for instance that the value *Competitive Leadership* has not been provided to the system. *Competitive Leadership* has four associated answers: weak, moderate, strong, and dominant. A typical voting function for this subjective question is:

$$v(x) = \begin{cases} -15 & \text{if } x = \text{weak} \\ -2 & \text{if } x = \text{moderate} \\ 6 & \text{if } x = \text{strong} \\ 20 & \text{if } x = \text{dominant} \end{cases}$$

The variate X mapped onto vote space is now uniformly distributed over the range that $v(X)$ can take: $v(X) \sim \mathcal{U}(-15, 20)$. The assessment for *Competitive Leadership* can now loosely be indicated with $Y \sim \mathcal{N}(50, 3^2) + \mathcal{U}(-15, 20)$. The uniform distribution is strangely assumed to be

[1] Often the weights are referred to as votes. To avoid confusion we use the term weights.

normally distributed, and hence equations 6.3 and 6.4 can be applied again in combination with equations 6.6 and 6.7:

$$\mu_Y = 50 + \frac{1}{2}(-15 + 20)$$
$$\sigma_Y^2 = 3^2 + \frac{1}{12}(20 - -15)^2$$

Evidently the soft saturation function has to be applied to complete the *Competitive Leadership* assessment.

6.1.6 Evaluation

Until now, ABN AMRO model owners have seen MRA as a 'grey' box. Even Moody's KMV could not give a full explanation on the working of MRA or its assumptions. In this section, we have revealed the assumptions on which the MRA system relies. In short, these are:

- Experts can correctly formulate the relation between each parent node and its children
- The children of each parent node are statistically independent
- All variables, both inputs and intermediates, are normally distributed
- The result of any transformation on a normally distributed variable is normally distributed

The latter assumption is by definition violated due to the soft saturation function. Moreover, one idea behind the weighted voting functions is to enable experts to express non-linear relations. Recall that we are working with ratios that might have infinitely large values, where linear functions will lead to infinitely large (or small) votes. Non-linear mapping functions can reduce the range to a restricted interval. The result of a non-linear function applied to a normally distributed variate, however, no longer follows the normal distribution, and thus violates the latter assumption.

We can conclude that the model involves many shifts between probability distributions that have no theoretical foundation. One of the authors of Syntel, Duda, admits in an e-mail conversation with us that Syntel (and thus MRA) borrows concepts from probabilistic reasoning, but that it is not derived from a consistent, rigorous mathematical model (Duda 2004). It further relies heavily on normality assumptions. The last aspect worth mentioning is the lack of function restriction in the voting functions, introducing seven degrees of freedom in each node. This implies the need for huge amounts of data if we would like to automatically learn the weights from the data.

6.2 Model

In this section we will discuss the implementation of the MRA techniques in our new rating tool. We will refer to this application as QC-MRA: Quantitative Consultancy's variant of MRA. The focus will be on the default behaviour of the system. There are many exceptions to this procedure; these are listed in (Dikkers 2005a). The object-oriented nature of our generic rating tool from chapter 5 has been used to implement the described model. Several classes have been reimplemented to serve the needs of the MRA model. The source code documentation (Dikkers 2005b) contains a more detailed description of all classes and functions. This documentation is composed of easily browseable web pages that have been created using the Doxygen toolkit[2]. A class diagram of the main classes of QC-MRA can be found in figure C.1 in the appendix.

[2] http://www.doxygen.org

Inputs

We will follow a bottom-up approach to explain the model. Each lowest-level node, or leaf, is represented by a raw model input. Remember that all variables in the Syntel tree are probability distributions. Therefore, the raw input is represented by a normally distributed variate with the raw value as its mean, and a standard deviation of zero.

Each raw model input has an associated assessment, which evidently lies one level higher up the tree. Equations 6.3 and 6.4 from section 6.1 are applied: let $Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$, and simplify to

$$\begin{aligned} \mu_Y &= 50 + v(X) \\ \sigma_Y^2 &= 3^2 \end{aligned}$$

Note that both X and $v(X)$ have become scalars, because the variance of X equals zero. The value $v(X)$ has already been introduced in section 4.3.3, where we have named it a *score*. To complete the calculation of the assessment, we apply the soft saturation function.

Overall ratio assessment

The assessment of a ratio depends on four aspects: the raw ratio value, percentile value, trend, and volatility. The subtree of a overall ratio assessment is shown in figure 6.2. All input values, which are represented by parallelograms, are first translated into their corresponding assessments. Next, the trend and the volatility assessments are combined using the weighted voting function. This intermediate assessment is then aggregated with the percentile assessment. Finally, the raw value assessment is included to lead to the overall ratio assessment.

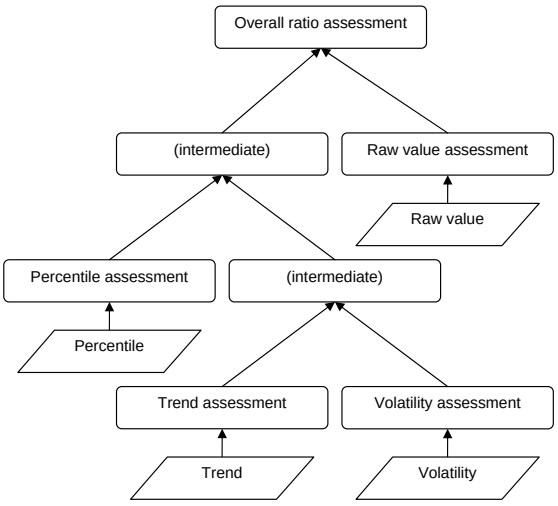


Figure 6.2 — Overall ratio assessment subtree

Missing values in this part of the tree are ignored. If for instance no trend and no volatility values are available, their assessments will be undefined. This implies that the parent node of trend and volatility is undefined as well. The parent node of the percentile assessment and this node will thus be based on only the percentile assessment.

Higher levels

If we look at the complete tree in figure 6.3, we can find the financial ratios as the items without a surrounding box on the left-hand side. Several ratio assessments are combined into an intermediate node. Each of these ratio assessments represents a subtree as in figure 6.2.

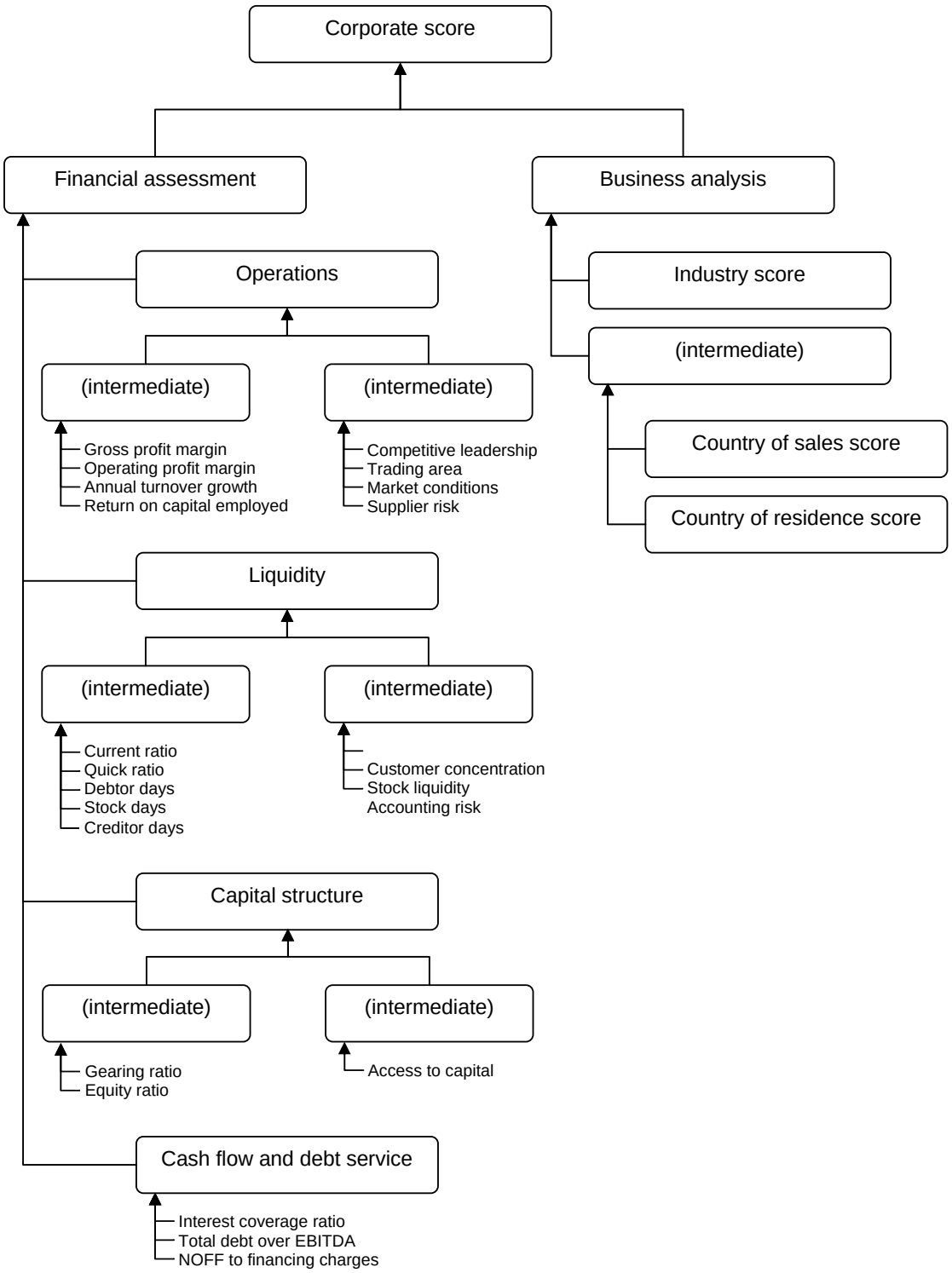


Figure 6.3 — The complete tree

The variables without the surrounding box on the right-hand side represent the answers to subjective questions. Similar to the other model inputs, the answers have been translated into assessments. These assessments are combined into intermediate assessments as well. Two intermediate assessments are aggregated to form the *Operations*, *Liquidity*, *Capital Structure*, or *Cash Flow and Debt Service* assessment. These four assessments are called the *financial pillars*.

Missing values in this and the remaining parts of the tree will not be set to undefined, but have associated prior probabilities as described in the previous section.

The four financial pillars are combined to form the *Financial Assessment* on the left-hand side of the tree. On the right, the less complicated *Business Analysis* subtree is formed by the three macro-economic model inputs: *Industry Score*, *Country of Residence Score*, and *Country of Sales Score*. The latter two are first combined into an intermediate assessment. The root of the tree is named the *Corporate Score*. A lookup table exists to translate this assessment into a UCR.

6.3 Evaluation

QC-MRA evidently inherits all properties of the rating framework it is built in. It therefore meets all requirements regarding data storage and flexibility (cf. section 5.4). The remaining two requirements are response time and system documentation. A final requirement is that we expect our system to produce the same results as the original MRA application.

For the comparison of MRA and the approximation of MRA in our own model, we have created a portfolio of 15,000 counterparties. The original MRA system ran for over twelve hours to complete the task. QC-MRA was tested on the same portfolio, and finished the job in thirteen minutes, which is a response time reduction of over 98%.

With a small amendment we can even meet our two-minute requirement from section 5.2.2. A quick analysis learnt that storing data was the bottleneck. *All* intermediate assessments of the tree are saved, which are 86 fields per counterparty. If we drop the requirement to store all intermediate values and focus on the main few, QC-MRA can process 15,000 counterparties in less than two minutes, which does meet our speed requirement. This implies a response time reduction of over 99.7% compared to MRA.

We have included user-friendly and browseable source code documentation to our rating tool (Dikkers 2005b). Model owners that are new to the system can hence easily get acquainted with our generic rating tool and the MRA implementation specifically.

The final question is whether QC-MRA derives the same results as MRA. When comparing the output UCRs of both programs, we found identical UCRs for 96% of the counterparties, which is a very good result. The 4% deviations were caused by undocumented unpredictable calculations in the original MRA program. One often seen deviation occurs in *Interest Coverage Ratio* calculations: if the denominator of this ratio equals zero and the numerator is larger than zero, we would expect the *Interest Coverage Ratio* to be $+\infty$. In rougly one out of three times, however, its value will strangely be $-\infty$. The impact of changing a value from $+\infty$ to $-\infty$ is substantial, even in the MRA tree.

6.4 Recommendations

Several of the legacy implementation choices of Syntel and MRA in particular do not seem to add value to the system. One of the main reasons that MRA is regarded as a grey box is its use of the soft saturation function. We have seen that the idea of the the soft saturation is to restrict the output to the scale $[0, 100]$. This goal can, however, also be reached by restricting the minimum and maximum values of the independent voting functions in such a way that their addition will never exceed 100 or come below 0. In the current MRA settings, this is already the case for 95% of the nodes. When the remaining 5% are amended accordingly, we can safely remove the soft saturation transformation from the model. More information can be found in our confidential report (Dikkers 2004).

The main reason for the discrete character of the model can be found in the lack of processing speed and storage capacity in the late 1980s. Nowadays these aspects no longer pose a problem. It might be worthwhile to develop a continuous variant of the MRA system. The discrete voting functions can now be replaced by continuous functions. By restricting the function class of the functions that might be used, we can reduce the degrees of freedom at each node. The impact of this model change will be substantial; at each node both the function and its parameters need to be estimated. This procedure is similar to the initial model development process in 1998 (Zondag 1998a, 1998b).

Other recommendations regard the model inputs. Our main concern is the frequent violation of the independence requirement of the inputs. In section 4.4.3 we have seen that many variables that serve as model inputs are highly correlated. The theoretical fundamentals are based on statistical independence between the variables, and we should therefore reduce collinearity as much as possible. A second concern is the limited (or absent) predictive value in many of the model inputs. These variables will only add noise to the outcome of the model. We suggest a radical reduction of the number of highly correlated inputs and inputs with poor predictive power. Two possible sets of variables have been presented in section 4.5 at the introduction of reduced sets.

Our final recommendation concerns the tree structure. In the current set-up, the tree is built up from many different intermediate nodes. At each node, extra uncertainty is added, soft saturation is applied, and values are rounded off. Moreover, at each node, the parameters for the weighted voting function have to be estimated by experts. The current tree in its discrete nature therefore requires model experts to estimate 437 parameters in total. The added value of having so many intermediate nodes is questionable. In the *Capital Structure* pillar, for instance, only one subjective question serves as input for the right-hand side of this pillar. The left part is given by only two ratios. To combine these three inputs, in total 32[3] parameters have to be estimated by experts. Experts will never be able to express their knowledge in such a detailed way. We therefore suggest to reduce the model size and remove several intermediate nodes. One option would be to let the *Financial Assessment* contain the four financial pillars and one extra pillar *Subjective questions*. This latter pillar is then formed by all the subjective questions. All the boxes that have *intermediate* as their title in figure 6.3 can thus be removed, reducing the degrees of freedom by 43.

Similarly, on the lowest level of the financial ratios (cf. figure 6.2), we could simplify the way the four inputs are combined. One option is to allow the *Overall Ratio* assessment be directly composed of the four financial ratio inputs. Two almost meaningless intermediate nodes would thus be removed.

6.5 Conclusion

For the first time, the characteristics and assumptions of the MRA model are revealed. MRA is based upon loose statistical and mathematical grounds, but currently proves to be a reasonable model for ABN AMRO's credit rating needs. We can conclude that our own implementation of MRA, QC-MRA, produces UCRs similar to MRA, and can therefore replace MRA in tuning and stress testing applications. Moreover, QC-MRA can safely be integrated into ABN AMRO's rating framework GRACE, without losing any of the current functionality. By this means we will solve most of the drawbacks highlighted in the previous chapter.

If our recommendations in subsection 6.4 are followed as well, the model will be more transparent and easier to maintain. This will provide a good intermediate solution for ABN AMRO. ABN AMRO has decided to follow these recommendations and has started a project to implement QC-MRA into GRACE.

Another advantage is the fact that ABN AMRO now has the possibility to gradually move to more statistically underpinned models. These models can be easily added to our system. As credit

[3] We need to estimate 7 values for both ratio assessments, 4 for this particular subjective question, and 7 for each intermediate node. This makes a total of 32 parameters to estimate.

rating data from January 2004 onwards has now become available, the road to proven mathematical techniques has been paved. The next chapter discusses several models based on statistics, both standard regression techniques and state-of-the-art support vector machines approaches.

Chapter 7

SVMs in ordinal classification

In our problem definition we stated our desire to compare the results of MRA to that of proven and state-of-the-art techniques. Chapter 3 described the most commonly used techniques for the credit rating problem. In earlier times, statistical credit rating models were based on linear regression and linear discriminant analysis. Later, logistic regression was introduced and is still widely used throughout banks. Support vector machines are a state-of-the-art technology that have only been applied to the credit rating problem for the past two or three years.

Statistics provides useful advantages compared to expert systems like MRA. The quality of measurement can be determined, usually by means of the variance. We have seen that MRA gives a 'variance' as well, but this value has no theoretical foundation and only gives a rough indication concerning the variability around the mean. Probably the most interesting aspect of statistical techniques is that the model parameters can be estimated from the data. The model owners are no longer dependent on the subjective opinion of experts. Overall, statistics imply easier maintenance of the model.

We will benchmark MRA against three techniques:

- Support vector machines
- Ordinary least squares regression
- Ordinal logistic regression

This chapter will outline our design choices regarding support vector machines. Both regression techniques are more straightforward and require less explanation to the reader. We will only discuss these techniques briefly in section 8.2. We have left source code and model details out of the scope of this document, and have focused on the implementation choices on a higher level.

There are two important aspects in designing the support vector machine classifier: how to utilise the ordinal nature, and how to tune the hyperparameters. These two aspects will be discussed in the following two sections.

7.1 Support vector machines in ordinal classification

Support vector machines were originally designed for the binary case. The ordinal nature of the credit rating problem, however, provides us with extra information that should be incorporated in the classifiers. Ordinality borrows concepts from both classification and regression. Like in classification, the output is a finite set, and like regression, there exists an ordering among the elements in the set.

Support vector machines in combination with the (ordinal) credit rating problem has been researched in three other papers (cf. chapter 3). Friedman (2002) does not provide us with implementation details, but as he merely describes the SVM methodology as a classification technique, we can safely assume that ordinality has not been applied in his CreditModel program. Van Gestel et al. (2003) apply the all-pairs scheme to the credit rating problem, and determine the final

class using the majority voting algorithm. Huang et al. use Crammer and Singer's formulation for multi-class SVM classification, which is a single machine approach that does not take the ordinal character into account. The extra information that is provided by the ordinal nature of the problem is utilised in neither of the three papers.

Incorporation of ordinality might prove advantageous to the credit rating problem. Ordinal problems, however, have only scarcely been researched. We will discuss two ordinal approaches that have recently been proposed. Further, we propose a new approach that not only handles ordinality, but is more robust for overlapping data compared to competing techniques as well.

7.1.1 A general multi-class approach

In section 2.5, we discussed several ways to extend a binary classifier to a multi-class classifier, among which the one-against-all and the all-pairs techniques. Allwein et al. (2000) propose a generalised framework to handle multi-class problems. Their idea is to associate each of the c classes with a row of a *coding matrix*

$$\mathbf{M} \in \{-1, 0, +1\}^{c \times g}$$

for some g. In total $j = 1, \ldots, g$ binary classifiers f_j are trained with labelled data of the form $(\mathbf{x}_i, M(y_i, j))$ for all objects i in the training set, but omitting the objects for which $M(y, j) = 0$. Each of the binary classifiers f_j tries to minimise the costs $l(\cdot)$ on the induced binary problem, usually the misclassification error.

In the one-against-all approach, for example, $\mathbf{M}$ is a $c \times c$ matrix in which all diagonal elements are –1 and all other elements +1. The all-pairs approach involves a $c \times \binom{c}{2}$ matrix in which each column corresponds to a distinct pair (ω_1, ω_2). In this column, $\mathbf{M}$ is –1 in row ω_1, +1 in row ω_2, and 0 in the other rows. Both matrices have been given for the four-class case in table 7.1.

Table 7.1 — Coding schemes for the four-class problem. Rows represent the classes and columns the binary classifiers.

(a) *One-against-all*

$$\mathbf{M} = \begin{array}{c} \begin{matrix} f_1 & f_2 & f_3 & f_4 \end{matrix} \\ \begin{pmatrix} -1 & +1 & +1 & +1 \\ +1 & -1 & +1 & +1 \\ +1 & +1 & -1 & +1 \\ +1 & +1 & +1 & -1 \end{pmatrix} \end{array} \begin{matrix} \\ \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{matrix}$$

(b) *All-pairs*

$$\mathbf{M} = \begin{array}{c} \begin{matrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \end{matrix} \\ \begin{pmatrix} -1 & -1 & -1 & 0 & 0 & 0 \\ +1 & 0 & 0 & -1 & -1 & 0 \\ 0 & +1 & 0 & +1 & 0 & -1 \\ 0 & 0 & +1 & 0 & +1 & +1 \end{pmatrix} \end{array} \begin{matrix} \\ \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{matrix}$$

Suppose we want to classify a new object $\mathbf{x}$. Let $\mathbf{f}(\mathbf{x})$ be the vector that represents the outcome of the binary classifiers:

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_g(\mathbf{x}))$$

We will now need to *decode* the predictions of the f_j. Let $\mathbf{M}(r)$ denote row r of $\mathbf{M}$, which is thus associated with class ω_r. The object will be assigned to the class for which the row $\mathbf{M}(r)$ is 'closest' to $\mathbf{f}(\mathbf{x})$ with respect to some distance measure d.

One distance measure is given by the number of positions where the sign of prediction $f_j(\mathbf{x})$ differs from the corresponding matrix entry $M(r, j)$:

$$d_H(\mathbf{M}(r), \mathbf{f}(\mathbf{x})) = \sum_{j=1}^{g} \left(\frac{1 - \text{sign}[M(r, j) f_j(\mathbf{x})]}{2} \right)$$

In essence this is similar to calculating the Hamming distance between row $\mathbf{M}(r)$ and the signs of the $f_j(\mathbf{x})$. If either of these values is zero, however, this component adds 1/2 to the sum. The pattern is thus classified to class ω_i where

$$i = \arg\min_r d_H(\mathbf{M}(r), \mathbf{f}(\mathbf{x}))$$

This method of combining the f_j is called *Hamming decoding*. Note that majority voting is a special case of Hamming decoding.

7.1.2 Single machine approach

Herbrich et al. (2000) are the first to explicitly take ordinality into account in support vector machines. In their article they present a distribution independent approach that maps objects to scalar *utility values*. Their main idea is to find a classifier $\mathbf{w}$ such that

$$\mathbf{w}^T\mathbf{x}^{(1)} > \mathbf{w}^T\mathbf{x}^{(2)} > \mathbf{w}^T\mathbf{x}^{(3)} > \dots,$$

where $\mathbf{x}^{(i)}$ denotes any $\mathbf{x}$ in class ω_i. In binary comparisons,

$$\begin{aligned}\mathbf{w}^T\mathbf{x}^{(1)} &\geq \mathbf{w}^T\mathbf{x}^{(2)} + 1\\ \mathbf{w}^T(\mathbf{x}^{(1)} - \mathbf{x}^{(2)}) &\geq 1\end{aligned}$$

The classification is thus based on the difference between two vectors. For this means a new training set S' is created, that consists of ℓ^2 objects $\mathbf{x}'_i$ that represent the difference between two original objects. The corresponding class y'_i is given by the sign of the rank difference:

$$S' = \{(\mathbf{x}_i - \mathbf{x}_j, \operatorname{sign}[y_i - y_j])\}_{i,j=1}^{\ell} = \{(\mathbf{x}'_i, y'_i)\}_{i=1}^{\ell^2},$$

where the cases $y_i = y_j$ are omitted from S'.

The problem of ordinal regression is thus reduced to a classification problem on pairs of objects, where the number of constraints compared to the standard SVM problem has grown to ℓ^2:

$$\begin{aligned}\min_{\mathbf{w},\boldsymbol{\xi}} \quad & \mathcal{J}(\mathbf{w},\boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{\ell^2}\xi_i, \qquad C \in \mathbb{R}^+\\ \text{subject to} \quad & \begin{cases} y'_i\mathbf{w}^T\mathbf{x}'_i \geq 1 - \xi_i, & i = 1,\dots,\ell^2\\ \xi_i \geq 0, & i = 1,\dots,\ell^2\end{cases}\end{aligned}$$

Section 2.5 gives the solution to this optimisation problem. In this same section we have defined latent variable z as the signed distance to the separating hyperplane. The aforementioned utility value is the same as our latent variable:

$$z = U(\mathbf{x}) = \mathbf{w}^T\mathbf{x} = \sum_{i=1}^{\ell^2}\alpha_i y'_i K(\mathbf{x}'_i, \mathbf{x})$$

Finally, the decision boundaries are estimated.

This technique, however, involves ℓ^2 constraints in the optimisation problem and the creation an $\ell^2 \times \ell^2$ Gram matrix. The credit rating data set consists of approximately 1200 counterparties, implying over one million constraints and a matrix of more than one trillion entries. This is currently numerically infeasible to work with.

7.1.3 Logistic regression approach

Chang and Lin (2004) describe a hybrid multi-class classifier that combines ordinal logistic regression with support vector machines. The *one-against-preceding* coding scheme is used to produce $c-1$ binary classifiers in the c-class problem. For each classifier $1 \leq j < c$, the objects $\mathbf{x}_i$ are classified as:

$$\mathbf{x}_i \in \begin{cases} -1 & \text{if } y_i \leq j\\ +1 & \text{if } y_i = j+1\end{cases}, \qquad i = 1,\dots,\ell$$

The latent variable z_{ij} is calculated for all training samples $i = 1,\dots,\ell$ for each classifier $j = 1,\dots,c-1$. The vector $\mathbf{x}_i$ of each object can be represented as the $(c-1)$-dimensional vector $\mathbf{z}_i$.

Table 7.2 — Coding schemes for the logistic regression approach in the four-class problem

(a) *One-against-preceding*

$$\mathbf{M} = \begin{array}{c} \begin{matrix} f_1 & f_2 & f_3 \end{matrix} \\ \begin{pmatrix} -1 & -1 & -1 \\ +1 & -1 & -1 \\ 0 & +1 & -1 \\ 0 & 0 & +1 \end{pmatrix} \end{array} \begin{matrix} \\ \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{matrix}$$

(b) *Succeeding-against-preceding*

$$\mathbf{M} = \begin{array}{c} \begin{matrix} f_1 & f_2 & f_3 \end{matrix} \\ \begin{pmatrix} -1 & -1 & -1 \\ +1 & -1 & -1 \\ +1 & +1 & -1 \\ +1 & +1 & +1 \end{pmatrix} \end{array} \begin{matrix} \\ \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{matrix}$$

This vector $\mathbf{z}_i$ is thus associated with class y_i. We can now fit an ordinal logistic regression model on the data set $S = \{(\mathbf{z}_i, y_i)\}_{i=1}^{\ell}$ using the theory from section 2.2.

Chang and Lin's hybrid LOGREG-SVM approach is based on the latent variables, and can thus be solved as $c-1$ binary SVM problems with ℓ constraints and $\ell \times \ell$ Gram matrices. Their coding scheme, however, deviates from what we would expect from LOGREG theory, where $c-1$ classifiers compare the first j classes with the remaining $c-j$ classes:

$$\mathbf{x}_i \in \begin{cases} -1 & \text{if } y_i \leq j \\ +1 & \text{if } y_i > j \end{cases}, \qquad i = 1, \ldots, \ell$$

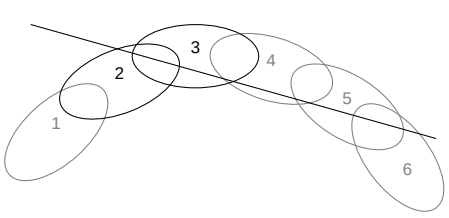
We will refer to this scheme as the *succeeding-against-preceding* scheme. This standard technique is symmetric, whereas reversing the order in one-against-preceding scheme might influence its outcome. Chang and Lin do not discuss why they deviate from the standard approach.

We do not agree with the one-against-preceding coding scheme. Inspired by Chang and Lin, we propose a modified version of their hybrid LOGREG-SVM approach: fitting an ordinal logistic regression model on the *succeeding-against-preceding* scheme.

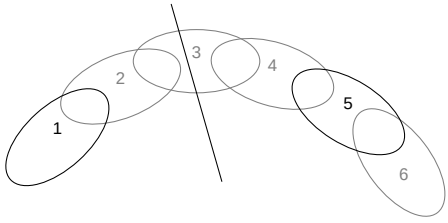
7.1.4 New approach: robust tree decoding

We will now describe a decoding scheme that not only utilises the ordinal character of the credit rating problem, but is robust for overlapping data as well. Recall from section 4.4.1 that our specific problem suffers heavily from overlapping classes. Classifiers that separate between two successive classes might therefore not always find an appropriate decision boundary due to distortion by the overlapping data points. This is illustrated in figure 7.1a: we would expect a nearly vertical decision boundary, but a suboptimal and thus less reliable boundary might be found.

On the other hand, we can safely assume that separation between classes that lie further apart (and are thus less overlapping) will be easier and hence lead to more reliable decision boundaries. This is illustrated by the Fisher analysis depicted as in figure 4.4, where we can see that the lowest class can easily be separated from the highest class using Fisher's first feature only. Typical decision boundaries of successive classes and classes that lie further apart are depicted in figure 7.1.



(a) *A typical decision boundary between two successive overlapping classes 2 and 3*



(b) *A typical decision boundary between two classes 1 and 5 that lie further apart*

Figure 7.1 — Two typical decision boundaries

This inspired us to construct a new decoding technique that utilises the ordinal ranking and the decreasing reliability of classifiers as classes lie closer to each other. We propose a *robust tree decoding* scheme. This scheme is based on the results of the ordinary all-pairs coding scheme. First, the classifier that is assumed to be most reliable is consulted: the (1 vs. c) classifier. If the classifier decides in favour of class 1, the (1 vs. $c-1$) is consulted next, and the (2 vs. c) otherwise. Basically a path through a classifier tree is followed. Figure 7.2 shows the tree for the four-class ordinal problem. The decoding algorithm is given in figure 7.4.

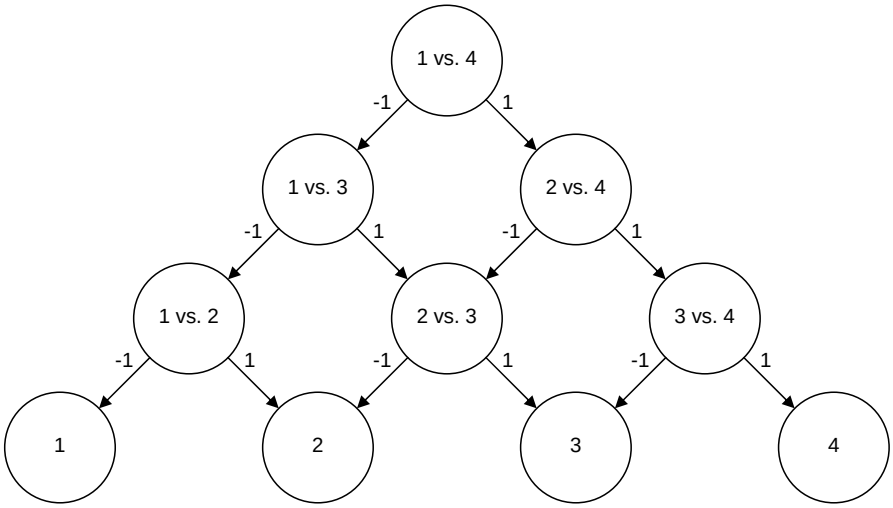


Figure 7.2 — Robust decoding tree that takes the ordinal character into account

This tree has several advantages over other decoding techniques:

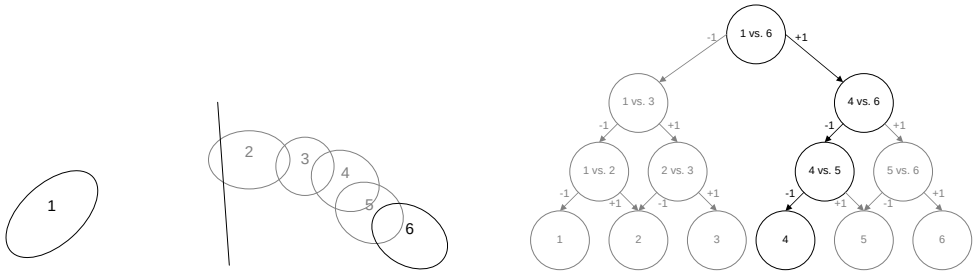
- Fewer evaluations
- Robustness
- Utilises the ordinal character

Since the robust tree decoding technique uses the all-pairs coding scheme, its training time depends on the training of $\binom{c}{2}$ classifiers. The reduction in evaluation time, however, is substantial compared to ordinary Hamming or loss decoding techniques. Instead of evaluating all $\binom{c}{2}$ classifiers, only $c-1$ classifiers have to be evaluated.

We could have followed the binary search paradigm, and cut the search space in half each time. The evaluation time would then depend on only $\lceil {}^2\log(n) \rceil$ classifiers. This approach has one large disadvantage. Since we are working with ordinal classes, we cannot assume that the distance between each of the classes is equal. The (1 vs. c) classifier, for instance, might perfectly divide between these two classes, but cannot be expected to separate between other classes as well. An example is depicted in figure 7.3a. The first binary classifier assigns a new pattern that is actually of class '2' to the right part of the tree, i.e., to classes 4–6. Even if all classifiers (including the first one) correctly classify this new pattern, it will never be classified as '2'. It will be assigned to class '4' at best.

This is where the robustness comes in. By proceeding with just one step at a time, we avoid large deviations in terms of distance between classes. When i classifiers along the path of the tree return the wrong class, the distance between the actual class and the predicted class will be i. At the lowest level of the tree, where succeeding classes are compared, only one of the $c-1$ classifiers is consulted. All other (presumably less reliable) classifiers are left out of the decision process.

One disadvantage of robust tree decoding is its bias towards the middle classes. When all the classifiers produce a random output, the distribution among the classes will be binomially distributed. We might want to include a small bias towards the outer classes in each node of the



(a) *The decision boundary between class 1 and 6 might not be between 3 and 4*

(b) *A correct decision of all classifiers leads to a wrong class*

Figure 7.3 — The binary tree approach might not suffice

```
function d = decode_ordinaltree(f,M)
% Compute the distance between 'f' and the rows of 'M'
%
% >>  distance = decode_ordinaltree(encoded_data, codingscheme);

[g,c] = size(M);

low = 1;
high = c;

while low ~= high
    % find the relevant classifier
    i = find(M(low,:) + M(:,high) == 0);
    if f(i) > 0
        high = high - 1;
    else
        low = low + 1;
    end
end
d = abs([1:c] - low);
```

Figure 7.4 — Robust tree decoding algorithm

tree. We should examine the latent outcome z of the classifiers instead of only the sign. Where we would normally decide on class –1 if $z < 0$ and class +1 otherwise, we can now classify the patterns to class –1 if for instance $z < 0.05$. The methodology used to derive this threshold is interesting matter for future work, and can be related to estimation of prior probabilities in the work of Hastie et al. (1998).

7.1.5 Conclusion

We have discussed three multi-class SVM approaches that take the ordinal character of the credit rating problem into account: Herbrich et al.'s single machine method, a modified version of Chang and Lin's logistic regression approach, and our own robust tree decoding. We have chosen to implement the latter two. We have included two decoding approaches that are based on the same encoding schemes and standard Hamming decoding schemes. In total we have come to four coding/decoding scheme combinations that have been listed in table 7.3.

For a fair comparison, we will slightly change to the Hamming decoding scheme. If the max-

Table 7.3 — Coding/decoding schemes used in our research

#	Coding scheme	Size	Decoding scheme
1	All-pairs	$\binom{c}{2}$	Hamming (similar to majority voting)
2	All-pairs	$\binom{c}{2}$	Robust tree
3	Succeeding-against-preceding	$c-1$	Hamming
4	Succeeding-against-preceding	$c-1$	Logistic regression

imum value is reached by more than one class, the average of their indices is taken. This value is rounded to the nearest integer to form the final class. When the average is exactly in between two classes, it is randomly rounded off upwards or downwards.

7.2 Hyperparameter estimation

Support vector machines have two types of hyperparameters to estimate: the regularisation parameter C and zero or more kernel parameters. Recall from section 2.5 that the regularisation parameter $C > 0$ determines the trade-off between regularisation and empirical risk minimisation. A small C allows for many classification errors, whereas the large C can loosely be said to lead to a more complex classifier.

The set of kernel parameters to estimate depends on the kernel function. Recall from section 2.5 that the most commonly used kernel functions are:

- Linear $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)$
- Polynomial $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + c)^d, \qquad c \in \mathbb{R}, d \in \mathbb{N}, \gamma \in \mathbb{R}^+$
- Gaussian RBF $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right), \qquad \sigma \in \mathbb{R}^+$

In our research we will mainly focus on the radial base function (RBF) for several reasons. The RBF kernel non-linearly maps samples into a higher dimensional space. The RBF kernel can handle the case when the relation between the dependent and independent variables is non-linear, in contrast to the linear kernel. Furthermore, it is shown that the linear kernel is a special case of the RBF kernel (Keerthi and Lin 2003). The number of hyperparameters influences the complexity of the model selection. The polynomial kernel function has more hyperparameters (three) than the RBF kernel function (one). The RBF function has less numerical difficulties as well, since $0 \leq K(\mathbf{x}_i, \mathbf{x}_j) \leq 1$, whereas for the polynomial kernel the value might go to infinity when d is large.

Loosely we can say that the RBF function gives significant response only in a 'neighbourhood' of its centre, i.e., if the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is small. The size of the neighbourhood is given by the parameter σ^2, and is often referred to as the *bandwidth* of the kernel. In regular support vector machines, the size of the bandwidth controls the complexity of a classifier. A larger bandwidth results in less support vectors that are required to describe the classifier, i.e., a more sparse classifier and thus a smoother function. If the bandwidth is too small, the system will overfit on the data. On the other hand, if the bandwidth is too large, the system will be unable to properly separate the data.

In the least squares variant of SVMs, however, we cannot easily state that a larger bandwidth leads to a less complex classifier. LS-SVMs do not provide a sparse solution, so the number of support vectors does not decrease when the bandwidth increases. Recall from section 2.5 that the classification function is given by

$$y = \text{sign}[\sum_{i=1}^{\ell} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b]$$

Since $\boldsymbol{\alpha}$ practically never equals zero in the LS-SVM approach, each $K(\cdot,\cdot)$ influences the final decision. When the bandwidth is large, many different $K(\cdot,\cdot) > 0$ contribute to the solution. A large bandwidth thus might lead to a more complex classifier, in contrast with regular SVMs. Without proper empirical or theoretical research, it is impossible to say in what way the bandwidth influences the complexity of the classifier in a specific problem.

Our goal is to find a proper combination (C, σ^2) such that the classifier can accurately predict out-of-sample data. We have seen before that this goal might not be reached by simply minimising the training error. To avoid the overfitting problem, we will use *cross-validation*. v-fold cross-validation splits the training set into v subsets of equal size. Sequentially, one subset is tested using the classifier that is trained on the remaining $v-1$ subsets. Each object of the complete training set is thus predicted once, so the cross-validation accuracy is the percentage of data that is correctly classified.

We have adopted a grid search mechanism to tune the hyperparameters. Basically, pairs of (C, σ^2) are tried and the one with the best cross-validation accuracy is chosen. Both Hsu et al. (2003) and Van Gestel et al. (2004) indicated that trying exponentially growing sequences of C and σ^2 is a practical method to identify good parameters. The grid search is performed with ten samples per parameter uniformly distributed in log space. The parameter ranges were $\ln(C) \in [-3, 12]$ and $\ln(\sigma^2) \in [-3 + \ln n, 9 + \ln n]$, where n is the number of features.[1] The prediction accuracy is determined using 5-fold cross validation. The misclassification error rate function serves as an evaluation function.

Remember that our SVM classifier is built up from multiple binary classifiers. The grid search will thus be applied to all binary classifiers, and different classifiers might have different parameters.

7.3 Implementation

Instead of ordinary support vector machines, we have chosen to implement least squares support vector machines for several reasons. LS-SVMs solve a linear system of equations, and hence are faster than regular SVM implementations. Faster algorithms imply more time to experiment with different kernels, parameter settings, and parameter optimisation methods. Secondly, Van Gestel et al. were the first to report extremely good results in their paper (2003), where LS-SVMs outperformed neural nets by a factor two and logistic regression by a factor 1.5 in classification accuracy. It would be interesting to see whether we can achieve similar results.

We implemented our system in MathWorks MATLAB [2]. We can easily convert MATLAB functions into a C++ function library using the MATLAB compiler. The compiler produces both a header file and a library file. By including the former and linking the latter we can use the full functionality of the MATLAB functions from within our rating tool.

The least squares support vector machines were implemented by the LS-SVM toolbox[3]. The Pattern Recognition Tools (PRTools) toolbox[4] provided the data structure and several functions for data exploration purposes.

[1] n is taken into account since $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ of the RBF kernel function is proportional to n (Van Gestel et al. 2004).
[2] http://www.mathworks.com
[3] http://www.esat.kuleuven.ac.be/sista/lssvmlab
[4] http://www.prtools.org

Chapter 8

Experimental results and analysis

This chapter presents the results of our experiments. The performance measures are given first. The experimental set-up of the two regression techniques is presented next. Section 8.3 discusses the results of all four techniques: linear regression, logistic regression, support vector machines, and MRA. This chapter is concluded with the evaluation of the different results.

8.1 Performance measures

Percentage correctly classified

The *percentage correctly classified* (PCC) measures the proportion of correctly classified objects on a sample of data. For the ordinal problem this might not always be the most appropriate performance criterion. The PCC tacitly assumes equal misclassification costs for all incorrectly classified objects. The ordinal nature of the problem however implies that a one-notch deviation is less problematic than a five-notch difference. We will therefore include the *percentage correctly classified within one notch* (PCC-1). Another implicit assumption is a relatively balanced class distribution of the examples. The transformation from fourteen to six classes helps us meet this requirement.

Kendall's τ

The Basel-II committee suggests the use of *Kendall's* τ to measure the degree of concordance between two ratings in their internal rating validation report (BIS 2005). The to be replicated rating is given by data from a credit rating agency or (in our case) a credit rating committee. Our goal is to achieve a high concordance between the predicted ratings and the to be replicated ratings. In case of high concordance, the predicted ratings will inherit the discriminatory power of the ratings from the agency or credit committee.

Kendall's τ is a measure of correlation between two ordinal variables $\mathbf{x}$ and $\mathbf{y}$ of length n. For all $\binom{n}{2}$ possible comparisons between the pairs (x_i, y_i) and (x_j, y_j) one determines whether the pairs are concordant ($x_i > x_j \wedge y_i > y_j$ or $x_i < x_j \wedge y_i < y_j$), discordant ($x_i > x_j \wedge y_i < y_j$ or $x_i < x_j \wedge y_i > y_j$), or tied ($x_i = x_j$ or $y_i = y_j$).

Let C be the number of concordant pairs and D the number of discordant pairs. Kendall's τ is now calculated as:

$$\tau = \frac{C - D}{\binom{n}{2}}.$$

Kendall's τ is equivalent to Spearman's ρ (cf. subsection 4.4.2) regarding the underlying assumptions, but its interpretation is different. Spearman's ρ is given in terms of proportion of variability accounted for. Kendall's τ can be seen as the difference between the probability that in the observed data the two variables are in the same order versus the probability that the two variables are in a different order.

Goodman-Kruskal Γ

Another measure of the concordance degree is the *Goodman-Kruskal* Γ. This degree is given by the surplus of concordant pairs over discordant pairs, as a percentage of all pairs ignoring ties.

$$\Gamma = \frac{C - D}{C + D},$$

The Goodman-Kruskal Γ is basically equivalent to Kendall's τ, except that the former explicitly takes ties into account.

NcNemar's $\mathcal{X}^2$

We will use the *McNemar's* $\mathcal{X}^2$ *test* to examine whether the predictive performance of one model is significantly better or worse than another. The McNemar test is a non-parametric test of the hypothesis that two related binomial variables have the same means. First, a contingency table is created that gives the number of objects that is correctly classified by both models (a), the number of objects correctly classified by model 1 but incorrectly by model 2 (b), etcetera:

	+	−
+	a	b
−	c	d

The columns give the frequencies of the correctly (+) and incorrectly (–) classified counterparties for model 1, and the rows for model 2. The McNemar statistic is now calculated as:

$$\mathcal{X}^2 = \frac{(b - c)^2}{b + c}$$

This value has one degree of freedom, and a p value can be derived from the $\mathcal{X}^2$ distribution.

8.2 Design of regression techniques

Both the regression techniques require fewer design choices than support vector machines. Regression does not involve hyperparameter tuning, and ordinality is naturally taken into account by logistic regression. One complicating aspect in regression, however, is its sensitivity to (partial) collinearity among variables. Ordinality and collinearity will be discussed in the subsections below. We have used J.P. LeSage's Econometrics toolbox[1] for MATLAB.

8.2.1 Ordinality

Ordinal logistic regression has specifically been designed for ordinal problems and requires no further tuning. Linear regression gives us several possibilities for the output that is to be estimated: the UCR rank, the probability of default or the logit of the PD (cf. section 2.1). We have seen in section 4.4.2 that the UCR rank and the logit of the PD are linearly related, and that researching only the latter will suffice. We will thus research both the PD and the logit(PD) as dependent variables.

8.2.2 Optimised sets

We attempt to determine the best predictive model using regression analysis. This involves removing the variables that do not significantly contribute to the regression equation. A widely

[1] http://www.spatial-econometrics.com

used technique for this means is *backward elimination*. The steps of backward elimination are as follows:

1. The regression equation with all of the independent variables is computed.
2. For each variable, the additional contribution of this variable on top of the contribution of all other variables is calculated using the *partial F test*. We have calculated the t statistic for each variable, which indicates whether a variable significantly differs from zero. In this case the t statistic is the square root of the F statistic.
3. The variable with the t statistic that indicates that it makes the least contribution to the regression equation is eliminated.
4. The regression equation is re-estimated using the remaining predictor variables.
5. If the difference between the results of the original regression model and the reduced model is significant[2] according to *Wald's F test*, the process is stopped. Otherwise the process is repeated as from step 2.

The backward elimination technique will be applied to all four data sets: both the (already) reduced and the full sets of raw data and scores.

8.3 Results

The out-of-sample performance of the implemented techniques is estimated using an out-of-sample test set. Cross-validation would give an even better view on predictive power. The SVM hyperparameter learning, however, can take several days on an average desktop computer, which demands as few runs as possible. We have therefore chosen to randomly select 80% of the data objects for the training set and the remaining 20% for the test set. Since unbalanced data sets might influence the performance of some of our techniques, we have ensured that the number of objects in both the test and the training set remains balanced. For a fair apple-to-apple comparison, we have used this same data set for all of the techniques.

In the following subsections, both regression and support vector machine techniques will be discussed. A mutual comparison is given in the fourth subsection. A qualitative review of the results is given in the evaluation section.

8.3.1 Ordinary least squares regression

We have applied ordinary least squares regression to all available data sets:

- Raw data and scores (cf. chapter 4)
- Full and reduced data set (cf. chapter 4)
- Optimised and not optimised (cf. section 8.2)
- PD and logit(PD) (cf. section 8.2)

The results are presented in table 8.1. The first column lists the models' rank based on the out-of-sample test set performance. The actual PCC on the test set is given in the fourth column with the train set performance between parentheses. The fifth and sixth column list Kendall's τ and Goodman-Kruskal's Γ respectively. Next comes the test set performance on the six rating classes. The final column gives $\bar{R}^2$, which indicates the fraction of variation in the dependent variable Y from its mean that is explained by the regression. This value is adjusted for the degrees of freedom. The three best performing models with respect to the PCC have been coloured grey.

One immediate observation is the underperformance of the models that are based on the PD compared to logit(PD) models. We can see that the PD models have extremely high accuracies in both class 1/2 and class 4 in comparison with the other classes. This can be explained completely in terms of intervals. The cut-off values are determined as the midpoint between the PDs of two

[2] Significance level of 5%.

Table 8.1 — Performance of the linear regression classifiers

Rank	Data set	# vars	PCC test (train)	τ	Γ	Per-class PCC 1/2	3+	3	3−	4	5/6	$\bar{R}^2$
	PD											
11	Full data set, all	69	32.8% (31.9%)	0.34	0.63	41%	14%	10%	21%	83%	13%	0.30
5	Full data set, optimised	10	34.9% (30.7%)	0.35	0.67	60%	11%	10%	10%	89%	16%	0.29
9	Reduced data set, all	27	33.6% (32.2%)	0.36	0.68	57%	6%	5%	21%	85%	13%	0.30
10	Reduced data set, optimised	11	33.2% (30.9%)	0.34	0.67	65%	6%	5%	17%	83%	10%	0.29
15	Full scores set, all	62	31.0% (32.2%)	0.33	0.62	51%	6%	3%	12%	81%	23%	0.33
13	Full scores set, optimised	11	32.3% (30.7%)	0.34	0.64	54%	11%	10%	10%	85%	10%	0.32
7	Reduced scores set, all	28	34.1% (32.3%)	0.36	0.68	54%	14%	3%	21%	85%	13%	0.32
16	Reduced scores set, optimised	11	30.6% (31.4%)	0.35	0.66	54%	11%	3%	10%	83%	10%	0.32
	logit(PD)											
14	Full data set, all	69	31.9% (37.3%)	0.42	0.62	27%	37%	18%	41%	45%	19%	0.46
3	Full data set, optimised	13	35.3% (36.8%)	0.45	0.68	19%	43%	30%	48%	51%	13%	0.45
4	Reduced data set, all	27	35.3% (35.0%)	0.45	0.67	19%	40%	28%	41%	57%	19%	0.45
12	Reduced data set, optimised	14	32.3% (36.3%)	0.46	0.69	11%	34%	33%	43%	53%	10%	0.45
1	Full scores set, all	62	38.8% (36.5%)	0.44	0.65	24%	49%	25%	50%	55%	23%	0.48
6	Full scores set, optimised	14	34.1% (36.2%)	0.43	0.65	16%	40%	23%	48%	55%	13%	0.47
2	Reduced scores set, all	28	35.8% (35.7%)	0.44	0.66	24%	43%	18%	55%	49%	19%	0.47
8	Reduced scores set, optimised	14	33.6% (36.1%)	0.43	0.65	16%	37%	20%	55%	49%	16%	0.47

classes. The interval in which a counterparty is assigned to class 4 is about 25 times larger than the interval of class 3+. Secondly, if the output of the regression model is negative, the counterparty will be classified as class 1/2. We will leave the regression models based on the PD out of the scope in the remainder of our report, and shift our focus to the logit(PD) models.

The McNemar test shows that there is no significant difference in predictive performance between the top six logit(PD) models. The input space of the unoptimised full data set is apparently too rich, whereas in the optimised reduced data set too many inputs have been eliminated. The models that are based on scores seem to perform better than those on raw data. Surprisingly, the two models that take many inputs perform best, where we would expect that data and scores sets that are optimised for linear regression achieve better results.

Table 8.2 — Linear regression beta coefficients of full data set, optimised

	Variable	Beta	*t*-stat
-	Intercept	-5.66	-189.8**
R12	Interest Coverage Ratio (raw value)	-0.35	-10.1**
P11	Equity Ratio (percentile)	-0.24	-7.3**
P13	Total Debt to EBITDA (percentile)	-0.22	-6.4**
P14	NOFF to Financing Charges (percentile)	-0.12	-3.7**
V04	Return on Capital Employed (volatility)	0.12	3.6**
M01	Industry Score	-0.15	-5.0**
M02	Country of Residence Score	-0.21	-5.1**
M03	Country of Sales Score	-0.15	-4.1**
Q01	Competitive Leadership	-0.23	-7.3**
Q03	Market Conditions	-0.11	-3.4**
Q05	Accounting Risk	-0.10	-3.1**
Q06	Customer Concentration	-0.10	-3.2**
Q08	Access to Capital	-0.14	-4.3**

*: 5% significance, **: 1% significance

The third best performing model is based on the full raw data set that has been optimised using backward elimination. We will use this model for further research, for it contains only thirteen variables and still has a good performance. Table 8.2 lists the inputs that have been used. Because we have normalised our inputs, the *beta coefficients* (in our notation w_i in the formula $\mathbf{w}^T\mathbf{x}$) indicate the univariate strength of each variable. When collinearity is largely removed from the model, we expect the signs of the beta coefficients to follow our hypotheses in table 4.5. The

volatility has a positive sign, whereas all other variables are negative. This observation meets our assumptions. Second, the Student's t statistic is given. A large absolute value is associated with a low probability that the beta coefficient equals zero but has another value by chance. All coefficients are found relevant at a 1% significance level.

The results of ordinary least squares regression are theoretically only valid if all assumptions from section 2.1 are met. The assumptions of no simultaneity, no collinearity, and expected value 0 for the residuals are met. We could not reject Spearman's null-hypothesis of homoscedasticity for any of the variables, which implies there is no significant heteroscedasticity. The stronger assumption of normally distributed residuals, however, is not met. The p-value for the Jarque-Bera test (numerically) equals zero, but the non-normality is shown even better in figure 8.1. The straight red dash-dotted line represents the normal distribution fitted to the residuals, and the residuals themselves are given by the blue markers.

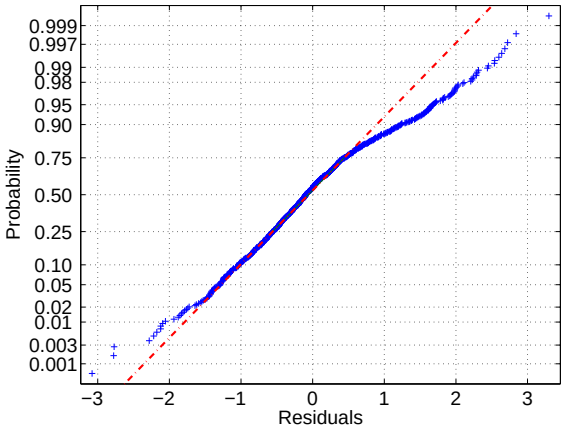


Figure 8.1 — Normal probability plot of the residuals

8.3.2 Ordinal logistic regression

The classification results of ordinal logistic regression are shown in table 8.3. The structure of the table is similar to the linear regression performance table, except for the last column. The LL stands for log-likelihood. The log-likelihood of the model is the value that is maximised by the process that computes the maximum likelihood value for the beta coefficients.

The McNemar test does not indicate significant differences in predictive performance between any of the logistic regression models. Like linear regression, the models that are based on scores seem to perform better than those based on raw data. Another interesting observation is that the (optimised) reduced scores models and reduced raw data models outperform their optimised versions when taking all inputs. Apparently the data reduction method from section 4.4.2 leads to a better reduced set than backward elimination on all variables.

Table 8.3 — Performance of the logistic regression classifiers

Rank	Data set	# vars	PCC test (train)	τ	Γ	Per-class PCC 1/2	3+	3	3−	4	5/6	LL 10^3
8	Full data set, all	69	35.8% (39.2%)	0.42	0.61	43%	29%	33%	26%	49%	32%	-1.29
6	Full data set, optimised	15	37.9% (39.2%)	0.45	0.65	54%	29%	35%	26%	47%	36%	-1.32
4	Reduced data set, all	27	38.4% (39.6%)	0.47	0.67	60%	17%	30%	29%	51%	42%	-1.32
3	Reduced data set, optimised	14	38.8% (37.8%)	0.48	0.69	62%	11%	30%	29%	53%	45%	-1.33
5	Full scores set, all	62	38.4% (41.6%)	0.45	0.64	51%	34%	33%	26%	45%	42%	-1.27
7	Full scores set, optimised	17	37.5% (39.5%)	0.45	0.64	60%	34%	23%	29%	47%	32%	-1.30
2	Reduced scores set, all	28	39.7% (40.5%)	0.45	0.65	54%	34%	35%	26%	53%	32%	-1.30
1	Reduced scores set, optimised	16	40.9% (40.3%)	0.46	0.66	60%	34%	28%	31%	55%	36%	-1.31

Table 8.4 — Logistic regression beta coefficients of reduced scores set, optimised

	Variable	Beta	*t*-stat
-	Intercept 1	-2.53	-21.6**
-	Intercept 2	-1.26	-13.8**
-	Intercept 3	-0.10	-1.2
-	Intercept 4	1.09	12.0**
-	Intercept 5	2.95	22.0**
R12	Interest Coverage Ratio (raw value)	-0.59	-7.6**
R13	Total Debt to EBITDA (raw value)	-0.74	-9.9**
P11	Equity Ratio (percentile)	-0.38	-5.0**
P14	NOFF to Financing Charges (percentile)	-0.35	-5.2**
T12	Interest Coverage Ratio (trend)	-0.20	-2.9**
V02	Operating Profit Margin (volatility)	-0.22	-3.1**
V03	Annual Turnover Growth (volatility)	-0.18	-2.7**
V11	Equity Ratio (volatility)	-0.22	-2.8**
M01	Industry Score	-0.38	-6.0**
M02	Country of Residence Score	-0.39	-4.8**
M03	Country of Sales Score	-0.32	-4.3**
Q01	Competitive Leadership	-0.56	-7.8**
Q04	Supplier Risk	-0.13	-2.1*
Q05	Accounting Risk	-0.24	-3.5**
Q06	Customer Concentration	-0.21	-3.2**
Q08	Access to Capital	-0.35	-5.2**

*: 5% significance, **: 1% significance

8.3.3 Support vector machines

As has been described in chapter 7, we have applied four approaches:

- All-pairs with Hamming (majority voting) decoding
- All-pairs with robust tree decoding
- Succeeding-against-preceding with Hamming decoding
- Succeeding-against-preceding with logistic regression decoding

Note that the first and third approach are merely the non-ordinal variants of the second and fourth approach respectively. In this subsection we will first focus on the linear kernel, and proceed with the Gaussian RBF kernel later.

Linear kernel

The linear kernel involves only one hyperparameter: the regularisation constant C. We would expect that C converges to a minimum in our line search. This, however, does not occur. In figure 8.2 C is plotted against the cross-validated misclassification error. We find a nearly flat function with multiple minima instead of the desired convex function. The differences in misclassification error are very small; often less than one percent point. The effectiveness of the optimisation of C therefore becomes doubtful.

The ordinal character of the problem can easily be observed from this figure as well. The (1 vs. 6) classifier, which tries to separate the highest UCR class '1/2' from the lowest UCR class '5/6', has error rates of less than 10%. The (1 vs. 2) classifier, however, performs at around 40%. In the previous chapter we posed the hypothesis that classes that lie further apart can be separated more easily. Our results strongly support this hypothesis.

The classification results of the linear kernel are listed in table 8.5. The classification accuracy of the linear kernel implementations range from 31.9% to 38.4%. The scores sets, both reduced and full, usually give better results than the raw data set. We can conclude that our LS-SVM classifier with logistic regression decoding performs best, for it outperforms the other techniques on all types of data. Both all-pairs schemes come second. Our robust tree algorithm performs similar to the majority voting decoding scheme, both in classification accuracy and in the concordance measures.

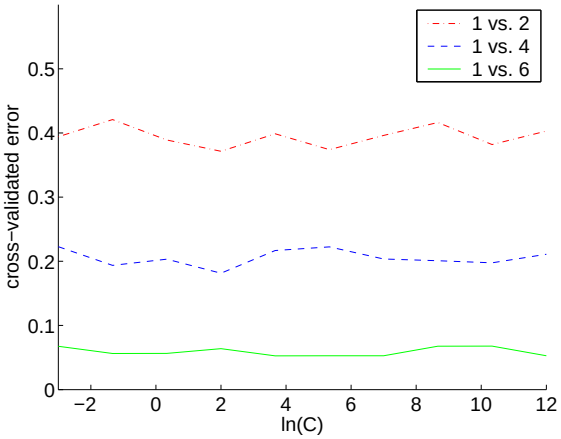


Figure 8.2 — Cross-validated misclassification error for different C of three typical binary classifiers (all-pairs classifier for the reduced scores set)

The differences between neither of the models are, however, statistically significant according to McNemar.

Another observation is a large difference between the training accuracy and the out-of-sample test accuracy. We can conclude that the LS-SVM classifiers are slightly overfit on the train data. A solution to this problem would be to decrease the value of C in the different binary classifiers, such that we allow for more misclassification errors and presumably improve the out-of-sample accuracy.

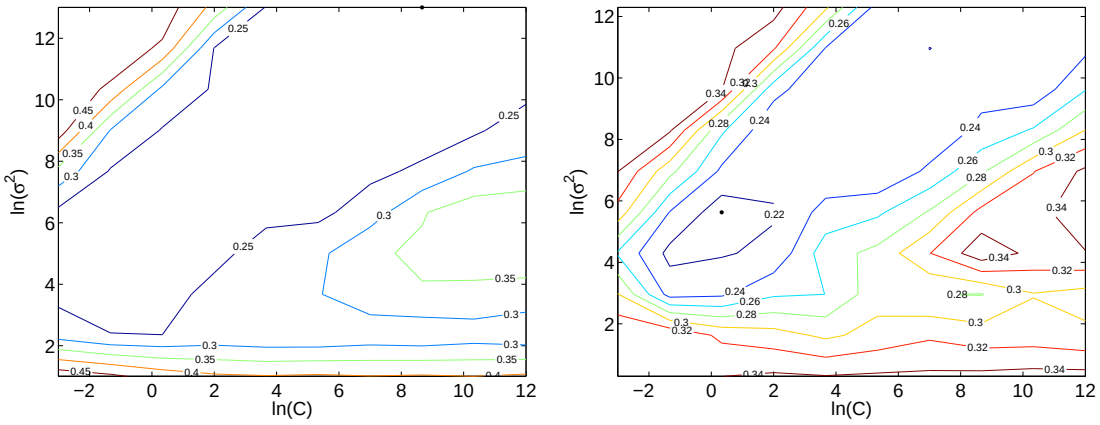
Table 8.5 — Performance of the LS-SVM classifiers with linear kernel

Rank	Data set	# vars	PCC test (train)	τ	Γ	Per-class PCC 1/2	3+	3	3−	4	5/6
	All-pairs scheme with Hamming (majority voting) decoding										
12	Full data set	69	33.2% (49.4%)	0.40	0.57	49%	20%	30%	29%	34%	39%
7	Reduced data set	27	34.9% (42.4%)	0.46	0.65	65%	26%	25%	19%	38%	39%
13	Full scores set	62	32.8% (49.6%)	0.33	0.48	38%	34%	25%	29%	36%	36%
5	Reduced scores set	28	37.5% (47.1%)	0.42	0.60	54%	31%	23%	38%	36%	45%
	All-pairs scheme with robust tree decoding										
11	Full data set	69	34.1% (49.2%)	0.41	0.59	46%	20%	35%	29%	36%	39%
8	Reduced data set	27	34.5% (42.6%)	0.46	0.65	65%	23%	25%	19%	38%	39%
16	Full scores set	62	31.9% (49.4%)	0.33	0.47	38%	37%	23%	26%	34%	36%
3	Reduced scores set	28	37.9% (47.2%)	0.42	0.60	51%	34%	25%	38%	38%	42%
	Succeeding-against-preceding scheme with Hamming decoding										
15	Full data set	69	31.9% (40.9%)	0.41	0.63	14%	43%	30%	45%	34%	23%
14	Reduced data set	27	32.3% (36.6%)	0.46	0.69	8%	54%	33%	38%	43%	13%
9	Full scores set	62	34.5% (42.4%)	0.41	0.61	11%	51%	30%	45%	40%	26%
10	Reduced scores set	28	34.1% (39.3%)	0.43	0.65	8%	51%	28%	55%	40%	16%
	Succeeding-against-preceding scheme with logistic regression decoding										
6	Full data set, all	69	35.3% (38.5%)	0.42	0.62	43%	31%	35%	19%	51%	29%
1	Reduced data set, all	27	38.4% (38.0%)	0.47	0.69	62%	14%	33%	26%	53%	39%
4	Full scores set, all	62	37.5% (41.6%)	0.45	0.64	51%	29%	38%	21%	45%	42%
2	Reduced scores set, all	28	38.4% (40.5%)	0.44	0.64	54%	29%	33%	29%	51%	32%

Radial basis function kernel

The hyperparameter search for the Gaussian RBF kernel introduced a new challenge. A near-equal cross-validation error was found in a complete valley around the line $\ln(C) = \ln(\sigma^2)$, as can be seen in figure 8.3a. The optimal parameters in this case are indicated by the black dot at $(C = e^9, \sigma^2 = e^{13})$. Since this optimum lies on the boundary of our search area, we have

experimented with search areas with boundaries up to $\ln(C) = 20$ and $\ln(\sigma^2) = 20$, but we neither achieved better results nor found an end of this 'valley'.



(a) *No regularisation* (b) *Regularisation*

Figure 8.3 — Cross-validated misclassification error for different $\ln(C)$ and $\ln(\sigma^2)$ of typical binary classifiers (figures do not originate from the same data source)

If the cross-validation error is (nearly) equal, we prefer a less complex classifier. Our data set has overlapping classes, so a smoother function that allows for relatively many errors will presumably perform best. Section 7.2 discussed the impact of both the parameters. The regularisation constant C penalises for errors. The larger the penalty, the more the classifier needs to fit outlier data, the more complex the classifier becomes. Therefore we prefer a small C. The kernel parameter σ^2 is the bandwidth. Due to our LS-SVM choice, we cannot state from theory whether a small or large σ^2 leads to a more complex classifier. Our own experiments have showed us that binary classifiers with a large σ^2 give very bad results: all unseen objects are always assigned to the same class. We therefore prefer to have smaller σ^2.

Regularisation can provide a solution. We have chosen to slightly penalise complex models using the following formula:

$$l'(f(\mathbf{x}), y) = \exp(\ln(C) + \ln(\sigma^2) - \lambda_1)/\lambda_2) l(f(\mathbf{x}), y)$$

The parameters in this function have been chosen in such a way that the penalty for the most possible complex classifier is 5%: $\lambda_1 = -3$ and $\lambda_2 = 550$. Figure 8.3b shows the result of this regularisation on a typical grid search. We can see that with regularisation, a combination with a small C and σ^2 is preferred. The obtained parameters are in this particular case approximately $(C = e^0, \sigma^2 = e^6)$.

Table 8.6 shows the results of the different LS-SVM classifiers based on the RBF kernel function. The results are similar to those of the linear kernel; all observations for the linear kernel hold for the RBF kernel as well. The most prominent observation again regards overfitting on the train data. The succeeding-against-preceding classifiers achieve training accuracies of 40–55%, which is slightly more than in the linear kernel variant. Both all-pairs schemes, however, have training accuracies of 50–80%. Despite our regularisation, the LS-SVM classifiers with RBF kernel are highly overfit on the train data.

We had expected that the RBF kernel function, which allows for separation by non-linear separating hyperplanes, would outperform models based on the linear kernel. There are several possible explanations why this does not occur. First, the best decision boundaries might be (nearly) linear. The similar performance of the RBF kernel is then explained by the fact that the linear kernel function is merely a special case of the RBF function. Aside from similar performance between the RBF and the linear kernel, we do not have any evidence that supports this theory.

Table 8.6 — Performance of the LS-SVM classifiers with RBF kernel

Rank	Data set	# vars	PCC test (train)	τ	Γ	Per-class PCC 1/2	3+	3	3−	4	5/6
	All-pairs scheme with Hamming (majority voting) decoding										
9	Full data set	69	34.5% (63.9%)	0.40	0.60	3%	54%	33%	29%	47%	42%
12	Reduced data set	27	33.2% (52.0%)	0.42	0.67	3%	51%	10%	67%	30%	39%
4	Full scores set	62	37.1% (80.1%)	0.42	0.62	41%	14%	33%	38%	51%	42%
14	Reduced scores set	28	31.9% (66.9%)	0.38	0.55	49%	29%	8%	33%	38%	36%
	All-pairs scheme with robust tree decoding										
8	Full data set	69	34.5% (63.3%)	0.41	0.61	3%	54%	25%	36%	47%	42%
11	Reduced data set	27	33.6% (52.5%)	0.42	0.67	0%	54%	18%	67%	26%	39%
3	Full scores set	62	38.4% (80.2%)	0.43	0.62	38%	17%	38%	41%	51%	42%
13	Reduced scores set	28	32.8% (67.9%)	0.40	0.57	49%	29%	8%	33%	43%	36%
	Succeeding-against-preceding scheme with Hamming decoding										
16	Full data set	69	18.1% (18.0%)	0.04	0.64	0%	0%	3%	98%	0%	0%
15	Reduced data set	27	28.0% (35.1%)	0.43	0.71	19%	83%	20%	29%	19%	0%
10	Full scores set	62	34.1% (46.6%)	0.44	0.68	30%	37%	38%	55%	15%	32%
7	Reduced scores set	28	34.9% (48.5%)	0.42	0.67	38%	37%	15%	71%	32%	10%
	Succeeding-against-preceding scheme with logistic regression decoding										
6	Full data set	69	35.3% (55.7%)	0.42	0.65	41%	51%	15%	14%	62%	26%
5	Reduced data set	27	36.2% (44.6%)	0.43	0.63	43%	37%	33%	21%	45%	39%
1	Full scores set	62	38.8% (42.2%)	0.43	0.62	49%	40%	33%	38%	40%	32%
2	Reduced scores set	28	38.4% (47.7%)	0.42	0.62	41%	40%	33%	21%	62%	29%

Moreover, linear support vector machines are closely related to multiple discriminant analysis; an underperforming technique according to our research peers (cf. chapter 3).

A more plausible cause can be found in the complexity of the data set. Recall that an SVM classifier is built up from binary classifiers. With our data set, this means that, on average, the classifiers are trained on 300 samples.[3] Usually this number will suffice. The data might, however, be too complex to determine proper decision boundaries on this amount of data. In this case we are unable to choose appropriate hyperparameters, resulting in underperforming multi-class SVM classifiers. Moreover, even if we would find appropriate hyperparameters, it might be impossible to properly fit a model.

We suspect that the cause for this complexity lies in data pollution. In appendix B we have explained the data retrieval and cleaning process. Despite all attempts to fully clean the data set, many erroneous objects may still be present. The slack variables allow for some errors, but the support vector machine technique remains vulnerable to outlier data.

8.3.4 Comparison

We have picked the best performing models from each of the categories for mutual comparison and a comparison with MRA. The performance results are given in table 8.7a, and a graphical representation of the PCC and PCC-1 are shown in figure 8.4. In terms of PCC and concordance measures, MRA outperforms the other techniques. If we focus on the classification accuracy that allows for a 1-notch difference, however, linear regression performs best. Again, it is clear that the LS-SVM implementation are overfit on the train data.

The McNemar values corresponding to the different pairs of models are listed in table 8.7b. Note that high p values indicate a high similarity. We can conclude that MRA does not significantly perform better than statistical techniques. Moreover, all researched techniques lead to similar results.

A final interesting aspect is whether one of the techniques consequently underclassifies or overclassifies counterparties. This is depicted in the five charts of figure 8.5. Each bar is 100% of the counterparties in a certain rating class. The blue portion is the percentage of overclassified counterparties, green indicates correct classification, and the red part of the bar gives the percentage of counterparties that has been assigned to a lower rating than their actual rating. The charts of

[3] 2/6 (two classes) of 80% (train data) of 1147 (number of observations) gives approximately 300 observations.

the regression and support vector machine implementations behave as expected, but we can easily see that MRA consequently overestimates the UCR of counterparties. The designers of the model have probably on purpose initiated the MRA model to overestimate ratings. The account manager is given plenty of opportunities to downgrade a counterparty according to his insights, but he can hardly upgrade an initial UCR from MRA. We can therefore not conclude the overestimation is an error; it is merely a design choice.

Table 8.7 — Comparison of the best performing classifiers

(a) *Performance indicators*

Rank	**Data set**	**# vars**	PCC **test (train)**	PCC-1 **test (train)**	τ	Γ
3	LINREG	62	38.8% (36.5%)	79.7% (83.0%)	0.44	0.65
2	LOGREG	16	40.9% (40.3%)	78.4% (80.3%)	0.46	0.66
5	LS-SVM (1)	62	38.4% (80.2%)	75.0% (85.1%)	0.43	0.62
4	LS-SVM (2)	62	38.8% (42.2%)	75.0% (84.8%)	0.43	0.62
1	MRA	62	43.1% (42.0%)	76.7% (73.2%)	0.48	0.69

(b) *McNemar values (p values between parentheses)*

	LINREG	LOGREG	LS-SVM (1)	LS-SVM (2)	MRA
LINREG		0.4 (53%)	0.0 (91%)	0.0 (100%)	0.9 (33%)
LOGREG			0.4 (51%)	0.4 (55%)	0.3 (57%)
LS-SVM (1)				0.0 (90%)	1.5 (23%)
LS-SVM (2)					1.1 (29%)
MRA					

LINREG: linear regression on logit(PD), full scores set, all
LOGREG: ordinal logistic regression, reduced scores set, optimised
LS-SVM (1): all-pairs LS-SVM (RBF kernel) with robust tree decoding, full scores set
LS-SVM (2): succeeding-against-preceding LS-SVM (RBF kernel) with logistic regression decoding, full scores set
MRA: original MRA, i.e., without the proposed changes from section 6.4

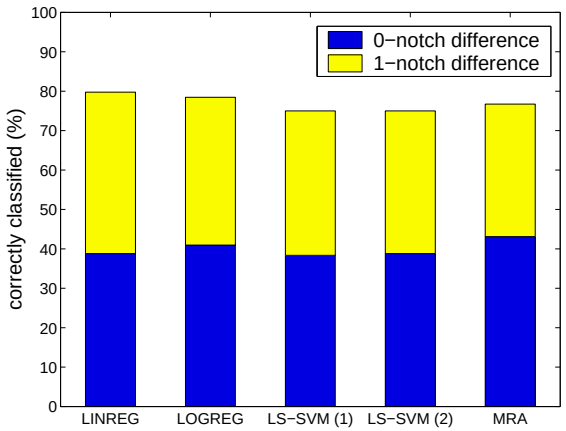
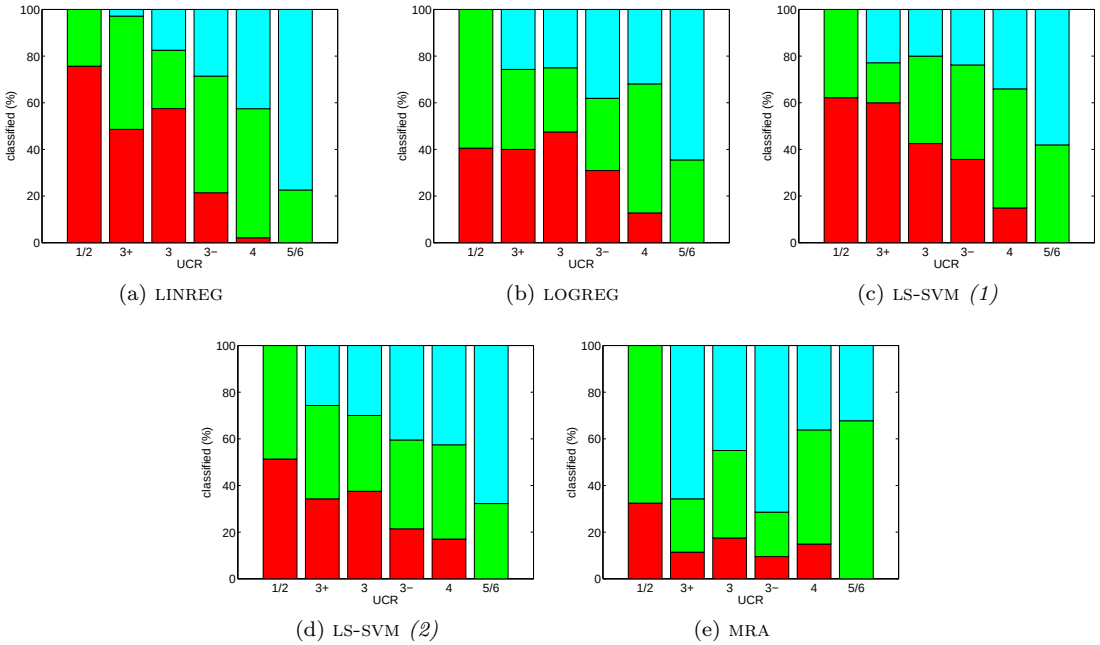


Figure 8.4 — Classification accuracy



(a) LINREG (b) LOGREG (c) LS-SVM *(1)*

(d) LS-SVM *(2)* (e) MRA

Figure 8.5 — Percentage correctly classified per notch (blue: overclassified, green: correctly classified, red: underclassified)

8.4 Evaluation

We have achieved a similar classification accuracy in linear regression, logistic regression, SVMs, and MRA. In chapter 3, however, we have seen that support vector machines can significantly outperform other techniques like linear and logistic regression. Section 3.3 indicated why our research peers achieve such high results with SVMs. The fifteen-class problem of Van Gestel et al. (2003) proved to be an eight-class problem in practice. The most important aspect, however, is the biased 'out-of-sample' test sets that are used by both Van Gestel et al. and Huang et al. (2004). Support vector machines that in practice overfit the data now seem to give good results. The neural networks suffer from the same biased test set and will give unreliably high prediction accuracies as well.

A comparison with research peers does thus not give a correct view of our results. A classifier that randomly assigns classes to objects is expected to give a prediction accuracy of approximately 17%. Compared to this number, the accuracy of both our models and MRA, which is around 40%, is reasonable. If we focus on the percentage correctly classified within 1 notch, however, our models perform around 80%, where a random classifier would only correctly classify 44%[4].

The results of our two SVM techniques that take the ordinal character of the credit rating problem into account, are promising. The logistic regression decoding consequently outperforms Hamming decoding, often even significantly. Our robust tree decoding leads to results that are similar to the majority voting algorithm, where we had expected that the former would outperform the latter. It would be interesting to research whether the robust tree algorithm performs better on an ordinal data set that is easier separable. We still feel that incorporation of the ordinal nature adds value to the classifier, although we cannot prove it with our results.

A remarkable other aspect that we have discussed is the extreme high classification accuracy in the train set compared to the test set. The all-pairs schemes achieve training classification accuracies of 50–80%, indicating that the classifiers are highly overtrained. We have experimented

[4]Due to the uniform distribution of patterns among the classes, the probability is given by the sum of the probability per class, divided by the number of classes: $\frac{1}{6}(\frac{2}{6}+\frac{3}{6}+\frac{3}{6}+\frac{3}{6}+\frac{3}{6}+\frac{2}{6}) = \frac{16}{36} \approx 44\%$

with regularisation to improve the out-of-sample accuracy, with limited success. More research is required to improve this out-of-sample accuracy; we think that a better alignment of the train and test accuracy results in a classifier that can let support vector machines outperform regression techniques. For this means, we could consider a different way to determine the hyperparameters of the model, such as Bayesian methods (Suykens et al. 2002).

A number of other actions can improve our SVM results. We have used the least squares variant of SVMs that does not give a sparse solution. We could either experiment with regular support vector machine approaches, or apply a pruning technique to improve sparsity (Suykens et al. 2002). Other future work lies in the design of the classifiers. Both our robust tree and logistic regression decoding schemes depend on the latent variables of the classifiers. Hastie et al. (1998) discuss a way to calculate posterior probabilities for the classes out of these latent variables. This might prove to be a valuable extension to the proposed decoding techniques.

Chapter 9

Conclusion

In the introduction we stated our project goal: *assess the performance of the present corporate credit rating model*. This report has provided the requested assessment, and tackled it in a three-tier approach:

1. Review of the status quo
2. Design and implementation of a new credit rating system
3. Comparison with mathematical and artificial intelligence techniques

First we will discuss the achievements, and we will conclude our report with suggestions for future work.

9.1 Achievements

Status quo

We have examined both the current system and the current model. The present rating system (MRA) poses several disadvantages. The most important drawbacks regard transparency, data storage, response speed, and flexibility. We have solved the data storage issues, but the remaining aspects demand the development of a new credit rating system. *cf. 5.1*

Aside from the system, the model has been reviewed as well. The rating model that is currently in use at ABN AMRO is regarded as a grey box by users and model owners. We are the first to reveal the assumptions on which the model is based. Hence, we can prove that several of the mathematical fundamentals of the model are invalid. Several recommendations are stated to improve both the theoretical correctness and the transparency of the model. *cf. 6.1* *cf. 6.4*

New rating system

We have designed and implemented a new rating system that overcomes the disadvantages of the current rating system. This system has originally been designed for stress testing and model tuning purposes, but is the starting point for the move to a different model as well. The current model has successfully been redesigned and incorporated into our new system. The new system improves the response time by over 99.7%. Hence, it is suitable for both stress testing and model tuning. Due to its object-oriented nature, the model can easily be extended with more statistically underpinned models. Proper documentation that is easily browseable ensures the transparancy of our system. *cf. 5.2* *cf. 6.2*

We can conclude that the new system can safely replace the existing rating system. Both a stress testing tool and a parameter estimation tool can be built on top of it.[1] Moreover, the system

[1] Both tools have been implemented by ABN AMRO colleagues in the spring and summer of 2005.

can be incorporated in the generic ABN AMRO rating framework GRACE, and replace the current system.[2]

Benchmark with mathematical and AI techniques

Research peers have applied several statistical techniques to the credit rating problem, of which logistic regression and support vector machines are the most promising. We have implemented both these techniques, and have included linear regression as well. In our research we have focused on incorporation of the ordinal character in support vector machines. Two new approaches are introduced: a hybrid support vector machine and logistic regression implementation, and a robust tree decoding technique that takes ordinality into account. Both techniques are compared with the variants that do not utilise the ordinal character, and to all other techniques as well, including the current model.

cf. 3

cf. 7

Both the regression techniques and our proposed support vector machine approaches achieve classification accuracies that are similar to the present credit rating model: around 40%, and around 80% if we allow for a 1-notch error. There is no statistical difference between the different techniques. We can thus conclude that the present rating model performs in line with (proven) statistical techniques. All our models, both the existing and the newly proposed ones, underperform compared to the results of research peers. We have shown, however, that their results are based on highly biased test sets, resulting in overestimated prediction accuracies. A fair comparison of results is therefore impossible.

cf. 8.3

cf. 3.3

We have seen that the statistical models that are based on the *scores*, i.e., variables that have been transformed using expert data, consequently outperform the models with regular inputs. In this way expert knowledge adds value to the different rating systems, both to MRA and to statistical models.

cf. 8.3

Our support vector machine implementations are highly overfit on the train data, despite our attempts to reduce the train set accuracy and improve the out-of-sample accuracy. We think this is caused by the complexity of the data; especially the large overlap between the rating classes. Support vector machines were originally designed for separable problems. Modifications allow for some misclassification errors, but might not be able to cope with the large overlap in our problem. The exact cause of the overtraining, however, merits further research.

The results of our ordinal extensions to the support vector machines are promising. The hybrid support vector machine and logistic regression approach consequently outperforms its non-ordinal variant. Our robust tree decoding algorithm achieves results that are similar to majority voting, which is its non-ordinal variant.

9.2 Recommendations

Our credit rating system can safely be integrated into the existing ABN AMRO credit rating framework. As an intermediate solution, a new version of the existing model can be used, in which our recommendations have been applied. When more data of better quality has become available, ABN AMRO can choose to move to a model that is partly based on statistics and/or artificial intelligence.

There are several possible options to improve our support vector machine approaches in such a way that there is less overfitting and a better out-of-sample performance. We have speculated about the causes of the overfitting, but have not given any proof. First we should try and find whether our speculations hold. This will indicate the most promising directions for improvements.

Aside from the overtraining problem, it would be interesting to include posterior probabilities in both our ordinal support vector machine techniques. But most importantly, our ordinal extensions to support vector machines should be tested on other ordinal problems. Our research only revealed a glimpse of their characteristics.

cf. 8.4

[2] ABN AMRO has decided to implement this recommendation and expects to have our system in place in December 2005.

Bibliography

Allwein, E.L., R.E. Schapire and Y. Singer, "Reducing multiclass to binary: a unifying approach for margin classifiers," *Journal of Machine Learning Research*, no. 1, pp. 113–141, 2000.

Baesens, B., T. van Gestel, M. Stepanova, J.A.K. Suykens and J. Vanthienen, "Benchmarking state of the art classification algorithms for credit scoring," *Journal of the Operational Research Society*, vol. 54, no. 6, pp. 627–635, 2003.

Bank for International Settlements, "Basel II: international convergence of capital measurement and capital standards: a revised framework," Bank for International Settlements, Basel, 2004.

Bank for International Settlements, "Studies on the validation of internal rating systems," Working paper no. 14, Bank for International Settlements, Basel, 2005.

Chang, Y.I. and S.C. Lin, "Synergy of logistic regression and support vector machine in multiple-class classification," *Lecture Notes in Computer Science*, vol. 3177, pp. 132–141, 2004.

Chaveesuk, R., C. Srivaree-Ratana and A.E. Smith, "Alternative neural network approaches to corporate bond rating," *Journal of Engineering Valuation and Cost Analysis*, vol. 2, no. 2, pp. 117–131, 1999.

Chen, K.U. and T.A. Shimerda, "An empirical analysis of useful financial ratios," *Financial Management*, no. 10, pp. 51–60, 1981.

Crammer, K. and Y. Singer, "On the learnability and design of output codes for multiclass problems," in *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, San Francisco (CA), pp. 35–46, 2000.

Caouette, J.B., E.I. Altman and P. Narayanan, *Managing credit risk: the next great financial challenge*, John Wiley and Sons, New York (NY), 1998.

Cristianini, N. and J. Shawe-Taylor, *An introduction to support vector machines*, Cambridge University Press, Cambridge (NY), 2000.

Duda, R.O., P.E. Hart, R. Reboh, J. Reiter and T. Risch, "Syntel: using a functional language for financial risk assessment," *IEEE Expert*, vol. 2, no. 3, pp. 18–31, 1987.

Duda, R.O., P.E. Hart, and D.G. Stork, *Pattern classification*, 2nd edition, John Wiley and Sons, New York (NY), 2001.

Duda, R.O., *Re: question regarding Syntel*, E-mail to H.J. Dikkers, 30 August 2004.

Dutta, S. and S. Shekhar, "Bond rating: a non-conservative application of neural networks," in *Proceedings of the IEEE International Conference on Neural Networks*, San Diego (CA), pp. II443–II450, 1988.

Falkenstein, E., "Credit scoring for corporate debt," in *Credit Ratings: Methodologies, Rationale and Default Risk*, M.K. Ong, Ed., chapter 8, pp. 169–188. Risk Waters Group, London, 2002.

Friedman, C., "CreditModel technical white paper," Standard and Poor's, New York (NY),

2002.

Fung, G. and O.L. Mangasarian, "Proximal support vector machine classifiers," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco (CA), pp. 77-86, 2001.

Gestel, T. van, B. Baesens, J. Garcia and P. van Dijcke, "A support vector machine approach to credit scoring," *Bank en Financiewezen*, vol. 2, pp. 73–82, 2003.

Gestel, T. van, J.A.K. Suykens, B. Baesens, S. Viaene, J. VanThienen, G. Dedene, B. de Moor and J. Vandewalle, "Benchmarking least squares support vector machine classifiers," *Machine Learning*, vol. 54, no. 1 pp. 5–32, 2004.

Hair, J.F., R.E. Anderson and R.L. Tatham, *Multivariate data analysis*, 2nd edition, Macmillan Publishing Company, New York (NY), 1987.

Hastie, T. and R. Tibshirani, "Classification by pairwise coupling," *The Annals of Statistics*, vol. 26, no. 2 pp. 451–471, 1998.

Herbrich, R., T. Graepel and K. Obermayer, "Large margin rank boundaries for ordinal regression," in *Advances in Large Margin Classifiers*, P.J. Barlett, B. Schölkopf, D. Schuurmans and A.J. Smola, Eds., chapter 7, pp. 115–132. MIT Press, Camebridge (MA), 2000.

Horrigan, J.O., "The determination of long term credit standing with financial ratios," *Journal of Accounting Research*, Supplement 1966, pp. 44–62, 1966.

Hsu, C.W. and C.J. Lin, "A comparison of methods for multi-class support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

Hsu, C.W., C.C. Chang and C.J. Lin, "A practical guide to support vector classification," Working Paper, 2003.

Huang, Z., H. Chen, C.J. Hsu, W.H. Chen and S. Wu, "Credit rating analysis with support vector machines and neural networks: a market comparative study," *Decision Support Systems*, vol. 37, pp. 543–558, 2004.

Keerthi, S.S. and C.J. Lin, "Asymptotic behavior of support vector machines with Gaussian kernel," *Neural Computation*, vol. 15, no. 7, pp. 1667–1689, 2003.

Kim, J.W., H.R. Weistroffer and R.T. Redmond, "Expert systems for bond rating: a comparative analysis of statistical, rule-based and neural network systems," *Expert Systems*, vol. 10, no. 3, pp. 167–172, 1993.

Kumra, R., R.M. Stein and I. Assersohn, "Assessing a knowledge-based approach to commercial loan underwriting," Research report no. 2-00-1, Moody's KMV, October 2000.

Kwon, Y.S., I. Han and K.C. Lee, "Ordinal Pairwise Partitioning (OPP) approach to neural networks training in bond rating," *Intelligent Systems in Accounting, Finance and Management*, vol. 6, pp. 23–40, 1997.

Maher, J.J. and T.K. Sen, "Predicting bond ratings using neural networks: a comparison with logistic regression," *Intelligent Systems in Accounting, Finance and Management*, vol. 6, pp. 59–72, 1997.

McCullagh, P., "Regression models for ordinal data," *Journal of the Royal Statistical Society, Series B (Methodology)*, vol. 42, no. 2, pp. 109–142, 1980.

Moody, J. and J. Utans, "Architecture selection strategies for neural networks: application to corporate bond rating prediction," in *Neural works in the capital markets*, A.P. Refenes, Ed., chapter 19, pp. 277–300. John Wiley & Sons, Chichester, 1995.

Moody's KMV, "Moody's Financial Analyst - Guide to business analysis", Moody's KMV, London, 2002.

Müller, K.R., S. Mika, G. Rätsch, K. Tsuda and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2,

pp. 181–202, 2001.

Resti, A., "Replicating agency ratings through multinomial scoring models," in *Credit Ratings: Methodologies, Rationale and Default Risk*, M.K. Ong, Ed., chapter 10, pp. 213–232. Risk Waters Group, London, 2002.

Rifkin, R. and A. Klautau, "In defence of one-versus-all classification," *Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.

Risch, T., R. Reboh, P.E. Hart and R.O. Duda, "A functional approach to integrating database and expert systems," *Communications of the ACM*, vol. 31, no. 12, pp. 1424–1437, 1988.

Shin, K. and I. Han, "Case-based reasoning supported by genetic algorithms for corporate bond rating," *Expert Systems with Applications*, vol. 16, pp. 85–95, 1999.

Shin, K. and I. Han, "A case-based approach using inductive indexing for corporate bond rating," *Decision Support Systems*, vol. 32, pp. 41–52, 2001.

Standard and Poor's, "Corporate ratings criteria", Standard and Poor's, New York (NY), 2004.

Stark, J., Private Communications, 2004.

Surkan, A.J. and J.C. Singleton, "Neural networks for bond rating improved by multiple hidden layers," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, San Diego (CA), pp. II157–II162, 1990.

Suykens, J.A.K., T. van Gestel, J. de Brabanter, B. de Moor and J. Vandewalle, *Least squares support vector machines*, World Scientific Publishing, Singapore, 2002.

Tutz, G. and K. Hechenbichler, "Aggregating classifiers with ordinal response structure," *Journal of Statistical Computation and Simulation*, vol. 75, no. 5, pp. 391–400, 2005.

Utans, J. and J. Moody, "Selecting neural network architectures via the prediction risk: application to corporate bond rating prediction," in *Proceedings of the First International Conference on Artificial Intelligence Application on Wall Street*, New York (NY), pp. 35–41, 1991.

Vapnik, V., *Statistical learning theory*, John Wiley and Sons, New York (NY), 1998.

Webb, A., *Statistical pattern recognition*, John Wiley and Sons, New York (NY), 2002.

Confidential and internal reports

Dikkers, H.J. and R. Quere, "Towards Basel-II compliance in IRD," ABN AMRO confidential report hd-7208.doc, June 2004.

Dikkers, H.J., "MRA model issues – low level," ABN AMRO confidential report hd-7219.doc, August 2004.

Dikkers, H.J., "MRA exceptions," ABN AMRO confidential report hd-xxxx.doc, July 2005.

Dikkers, H.J., "QC-MRA source code documentation," April 2005.

Moody's KMV, "Risk advisor assessment methodology", version 2.0, Moody's KMV, London, 2004.

Zondag, C.P., "LA-Encore, explanatory note for the expert panel," ABN AMRO confidential report, March 1998.

Zondag, C.P., "Comments of the two expert panels," ABN AMRO confidential report, April 1998.

Appendix A

Financial background

A.1 Financial statement

A.1.1 Balance sheet

The balance sheet is the company's basic financial statement. It reflects the value of the company's assets, liabilities, and the equity at a specific point in time (the date of the balance sheet).

Table A.1 — Balance sheet

Assets	**Liabilities**
Net fixed assets	Equity
Financial investments	Reserves
Other non current assets +	Retained earnings +
	Net worth
Fixed assets	Net intangibles -/-
	Tangible net worth
Total inventory	Subordinated debt
Accounts receivable	Minority interest +
Other receivables	
Cash/bank deposits +	**Own and associated means**
Current assets	Long-term provisions
	Long-term debt +
	Long-term liabilities
	Short-term provisions
	Accounts payable
	Short-term bank debt
	Other short-term debt +
	Short-term liabilities
Total assets	**Total liabilities and own means**

A.1.2 Profit and loss account

The profit and loss account is compiled at the end of the fiscal year (or another accounting period) to show gross and net profit or loss.

Table A.2 — Profit and loss account

Turnover/sales
Cost of goods sold -/-
Gross profit/loss
Salaries and wages -/-
Other expenses -/-
EBITDA
Depreciation and amortisation -/-
EBIT
Interest and other financing charges -/-
Net operating profit/loss
Extraordinary gains/losses +/-
Profit/loss before tax and minority interest
Taxation -/-
Minority interest -/-
Net profit/loss

A.1.3 Cash flow statement

Fundamental to the lending decision is an assessment of the borrower's ability to service and repay the debt as scheduled. The cash flow statement shows the borrower's 'free' operational cash flow, which will be available for debt service.

After taking the net profit/loss directly from the profit and loss account, the cash flow statement is sensitised for both cash content (by eliminating non-cash items) and sustainability (by eliminating non-operational items). The financing costs are added back in order to assess the cash available to cover the total financing charges. Finally, the impact of the asset conversion cycle is taken into account: shifts in total inventory, accounts receivable and accounts payable.

Table A.3 — Cash flow statements

Net profit/loss
Depreciation and amortisation +
Interest and other financing charges +
Extraordinary gains/losses -/+
Book gains/losses in asset sales -/+
Minority interest +
Funds from operations
Change in total inventory +/-
Change in accounts receivable +/-
Change in accounts payable +/-
Net operating funds flow (NOFF)

A.2 Financial ratios

Table A.4 shows the ratios that are used throughout this report. The formula inputs can be found in appendix A.1.

Table A.4 — Financial ratios that are chosen as model inputs

Ratio	Formula
Gross profit margin	$\frac{\text{Gross profit/loss}}{\text{Turnover}}$
Operating profit margin	$\frac{\text{Net operating profit/loss}}{\text{Turnover}}$
Annual turnover growth	$\frac{\text{Turnover}_{\text{year } i} - \text{turnover}_{\text{year } i-1}}{\text{Turnover}_{\text{year } i-1}}$
Return on capital employed	$\frac{\text{EBIT}}{\text{Total debt + net worth + minority interest}}$
Current ratio	$\frac{\text{Current assets}}{\text{Current liabilities}}$
Quick ratio	$\frac{\text{Current assets} - \text{total inventory}}{\text{Current liabilities}}$
Debtor days	$\frac{\text{Accounts receivable} \times 365}{\text{Turnover}}$
Stock days	$\frac{\text{Total inventory} \times 365}{\text{Cost of goods sold}}$
Creditor days	$\frac{\text{Accounts payable} \times 365}{\text{Cost of goods sold}}$
Gearing ratio	$\frac{\text{Total debt}}{\text{Tangible net worth}}$
Equity ratio	$\frac{\text{Own and associated means}}{\text{Total assets}}$
Interest coverage ratio	$\frac{\text{EBIT}}{\text{Interest expense}}$
Total debt to EBITDA	$\frac{\text{Total debt}}{\text{EBITDA}}$
NOFF to financing charges	$\frac{\text{NOFF}}{\text{Interest and other financing charges}}$

A.3 Corporate credit ratings equivalents

Both Standard & Poor's Corporation (S&P) and Moody's Investor Service (Moody's) are credit rating agencies that assign ratings to bond issuers and bonds themselves. These ratings are publicly available. ABN AMRO uses its own internal rating system: the Uniform Counterparty Rating (UCR). Table A.5 shows their external rating equivalents.

Table A.5 — Corporate credit ratings equivalents

UCR	S&P	Moody's	Grade
1	AAA	Aaa	Investment grade
	AA+	Aa1	
	AA	Aa2	
	AA−	Aa3	
2+	A+	A1	
2	A	A2	
2−	A−	A3	
3+	BBB+	Baa1	
3	BBB	Baa2	
3−	BBB−	Baa3	
4+	BB+	Ba1	Non-investment grade or high-yield
4	BB	Ba2	
4−	BB−	Ba3	
5+	B+	B1	
5	B	B2	
5−	B−	B3	
6+	CCC+	Caa1	
	CCC	Caa2	
	CCC−	Caa3	
	CC	Ca	
	C	C	

Appendix B

Data collection and cleaning

This appendix describes the data collection and cleaning process. The sections can be best read with section 1.5 concerning the credit rating process in mind.

B.1 Data collection

Recall from section 1.5 that RAPID is the credit proposal system. A credit proposal contains a proposed UCR and a Corporate Rating Sheet PDF document. None of the input variables to derive the initial UCR is provided to RAPID.

To keep track of history, ABN AMRO stores all approved credit proposals into a historical database named IRD. The moment that an approved UCR is provided to RAPID, the credit proposal data will be stored into this database. Unfortunately the inputs to derive the initial and proposed UCR are not fed to the RAPID system and thus not be available to the IRD system. Only the far from complete Corporate Rating Sheet is attached to a credit proposal, saved in the practically unreadable PDF-format. For validation, research and model development purposes we will need to connect the MRA inputs to approved UCRs.

The data collection task was two-fold:

- Implement a procedure to automatically include the storage of model inputs in IRD
- Retrieve data as from January 2004

A logical solution for the first task would be to let all input data automatically flow from MRA into RAPID. This would however be a very expensive approach that required comprehensive RAPID changes. We have chosen an approach that would only affect the data feed to IRD: when a UCR is approved, the corresponding input data is retrieved from the MRA database and stored into IRD together with the RAPID data. We formed a team of both business and IT people to design and implement this procedure, which is referred to as the *Bridge*. This procedure came into production in November 2004.

Our secondary goal was to include historical UCR information as well. Before January 2004, all rating data was only stored on the local machines of account managers. It is therefore infeasible to retrieve this rating data. The present MRA system however stores the data for each counterparty centrally.

B.2 Data cleaning

The historical database eventually contained 3,059 counterparties. This data set is however highly polluted. The step-by-step removal process is given in table B.1. At first we have removed the counterparties that have a non-valid UCR 'X' or are in default (UCR 6, 7, or 8). The latter counterparties are rare.

Table B.1 — Data cleaning

All scores in IRD	3059
Approved UCR is X, 6, 7, or 8	453
	2606
Multiple UCRs for same borrower	597
	2009
Higher UCR due to group support	303
	1706
Various errors	195
	1511
No country or industry score	171
	1340
Unanswered subjective questions	138
	1202
Not same output	55
	1147

Counterparties should be rated at least once a year, but possibly (and preferably) more often. IRD might therefore contain multiple approved UCRs for the same borrower. In theory it would be very useful to have different approved UCRs based on similar but different underlying data. In practice, however, multiple UCRs are often based on the exact same underlying data, for we only have a one-year time window, and only annual financial statements are used. Including multiple UCRs for the same counterparty would thus create a bias in the data set. Aside from the bias, many pattern recognition techniques cannot handle different ys for the same $\mathbf{x}$. We have removed all but the most recent approved UCR of a borrower.

Many counterparties are subsidiaries of a larger group. The parent company of a subsidiary frequently gives a form of guarantee to a bank, in order to negotiate lower interest fees for the subsidiary. The approved UCR of the subsidiary will now be better due to 'group support'. It will no longer reflect the borrower's own strength. UCRs that have been promoted because of group support have been deleted from the data set.

The financial statements are manually entered by account managers and therefore subject to many errors. Some annual financial statements cover one month, where twelve months are expected. Presumably the account manager meant one year instead of one month. Another error is easily made when selecting the 'previous' statement, i.e., the statement that a statement reconciles to. This will usually be the statement of twelve months earlier, but strange other statements have been selected. These have been covered under 'various errors'.

Some variables appear on the Corporate Rating Sheet and are thus unlikely to be omitted. This holds for country scores, industry scores and answers to subjective questions. When one or more of these fields are left empty, the counterparty has been removed from the data set.

In chapter 6.3 we describe a model that should produce similar UCRs as the current MRA version. As we see in this chapter, this new model gives the same UCR in 96% of all cases. The remaining 4% is caused by inpredictable actions of MRA, usually because of the denominator of *Interest Coverage Ratio* being zero.

In the end we have 1,147 customers with relatively clean data. Figure B.1 shows the sample structure by UCR class. The rating mix looks consistent with the European market, where speculative grades are rather uncommon and only a small minority of all rated companies receive top investment grade levels (Resti 2002).

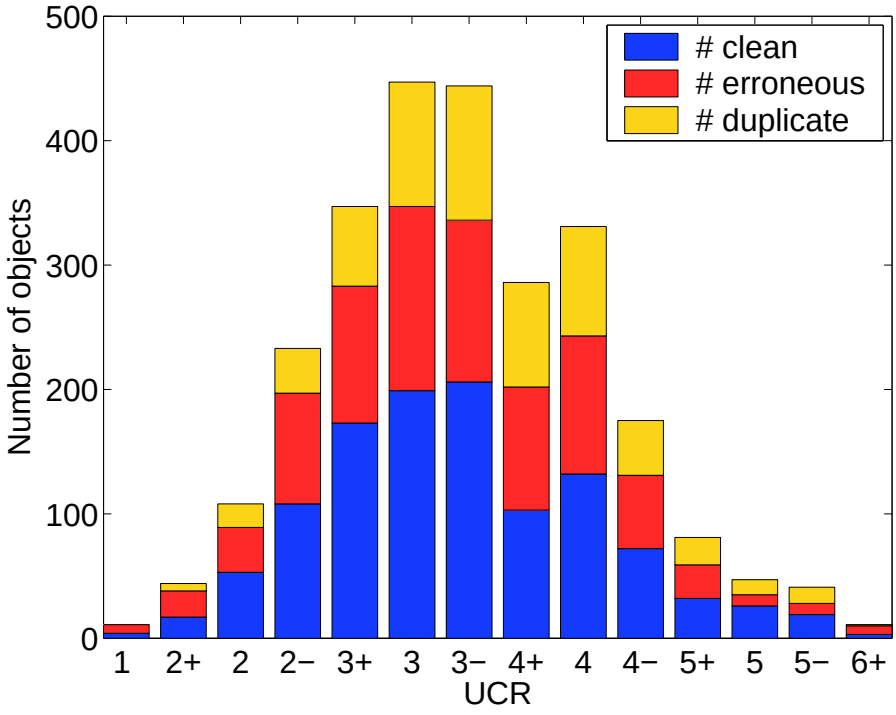


Figure B.1 — UCR distribution: complete versus cleaned

Appendix C

MRA model

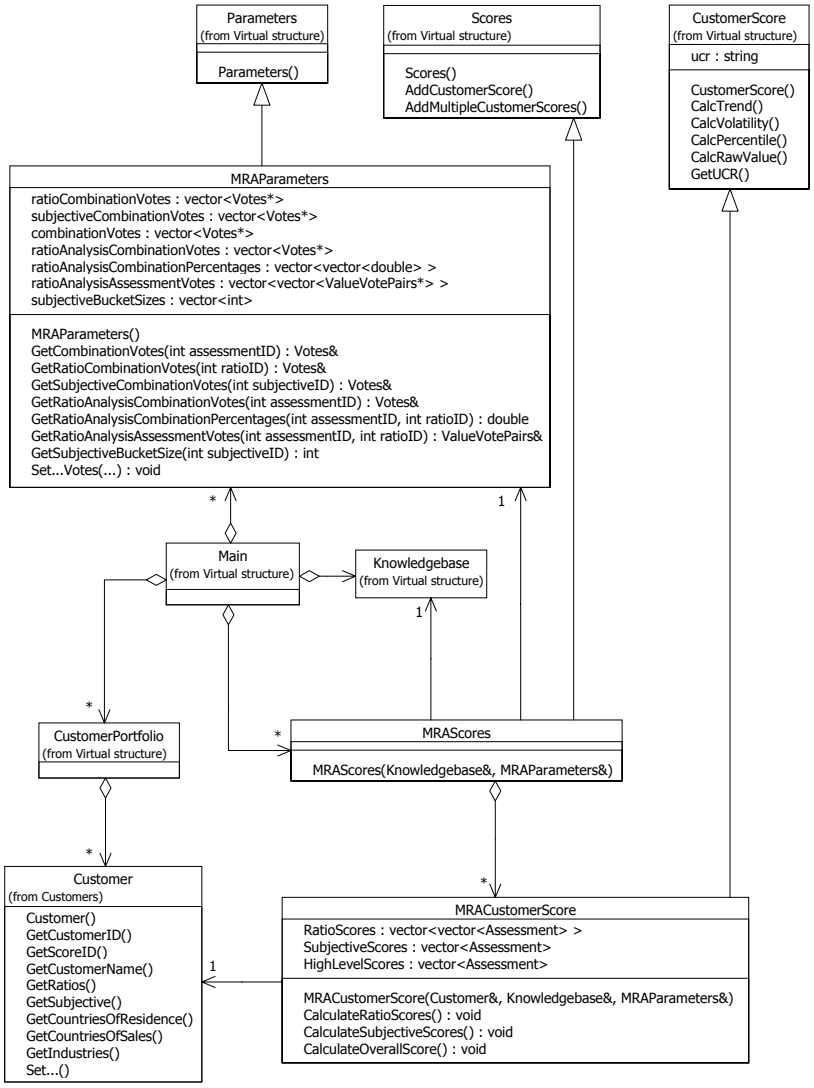


Figure C.1 — UML diagram of the MRA model