CREDIT SCORING METHODS AND ACCURACY RATIO

AYŞEGÜL İŞCANOĞLU

AUGUST 2005

CREDIT SCORING METHODS AND ACCURACY RATIO

A THESIS SUBMITTED TO
THE INSTITUTE OF APPLIED MATHEMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

AYŞEGÜL İŞCANOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF FINANCIAL MATHEMATICS

AUGUST 2005

Approval of the Institute of Applied Mathematics

Prof. Dr. Ersan AKYILDIZ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Hayri KÖREZLİOĞLU
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Kasırga YILDIRAK
Co-Advisor

Prof. Dr. Hayri KÖREZLİOĞLU
Supervisor

Examining Committee Members

Prof. Dr. Hayri KÖREZLİOĞLU _____

Prof. Dr. Gerhard Wilhelm WEBER _____

Assoc. Prof. Dr. Azize HAYFAVİ _____

Assist. Prof. Dr. Hakan ÖKTEM _____

Assist. Prof. Dr. Kasırga YILDIRAK _____

ABSTRACT

CREDIT SCORING METHODS AND ACCURACY RATIO

İşcanoğlu, Ayşegül
M.Sc., Department of Financial Mathematics
Supervisor: Prof. Dr. Hayri KÖREZLİOĞLU
Co-Advisor: Assist. Prof. Dr. Kasırga YILDIRAK

August 2005, 132 pages

The credit scoring with the help of classification techniques provides to take easy and quick decisions in lending. However, no definite consensus has been reached with regard to the best method for credit scoring and in what conditions the methods performs best. Although a huge range of classification techniques has been used in this area, the logistic regression has been seen an important tool and used very widely in studies. This study aims to examine accuracy and bias properties in parameter estimation of the logistic regression by using Monte Carlo simulations in four aspect which are dimension of the sets, length, the included percentage defaults in data and effect of variables on estimation. Moreover, application of some important statistical and non-statistical methods on Turkish credit default data is provided and the method accuracies are compared for Turkish market. Finally, ratings on the results of best method is done by using receiver operating characteristic curve.

Keywords: Credit Scoring, Discriminant Analysis, Regression, Probit Regression, Logistic Regression, Classification and Regression Tree, Semiparametric Regression, Neural Networks, Validation Techniques, Accuracy Ratio.

ÖZ

KREDİ SKORLAMASI VE DOĞRULUK RASYOSU

İşcanoğlu, Ayşegül
Yüksek Lisans, Finansal Matematik Bölümü
Tez Yöneticisi: Prof. Dr. Hayri KÖREZLİOĞLU
Tez Yönetici Yardımcısı: Yard. Doç. Dr. Kasırga YILDIRAK

Ağustos 2005, 132 sayfa

Kredi skorlama, klasifikasyon metodlarının yardımı ile kredi verme işlemlerinde kolay ve çabuk karar verilmesini sağlamaktadır. Fakat en iyi kredi skorlama metodlarının hangi koşullarda iyi performans gösterdikleri ve bunlardan hangisinin en iyi olduğu hakkında kesin bir yargı bulunmamaktadır. Bu alanda bir çok değişik metot kullanılmasına rağmen, lojistik regresyon önemli bir araç olarak görülmekte ve uygulamalarda yoğun bir şekilde kullanılmaktadır. Bu çalışma Monte Karlo simulasyonları kullanılarak lojistik regresyonun parametre tahmininde ki doğruluk ve yanlılık özelliklerini verinin boyut, uzunluk, veri içindeki temerrütteki gözlem oranı ve değişkenlerin tahmin üzerindeki etkileri olmak üzere 4 farklı açıdan incelemeyi amaçlamaktadır. Bu çaışma buna ek olarak Türkiye temerrüt verisi üzerinde, önemli bazı istatistiksel ve istatistiksel olmayan metotlar uygulanmış ile kredi skorlama yapılmakta ve bu veri için metotlar karşılaştırılmaktadır. Son olarak, receiver operating characteristic eğrisi kullanılarak, en iyi method sonuçları kullanılarak derecelendirilmiştir.

Anahtar Kelimeler: Kredi Skorlama, Diskriminant Analiz, Regresyon, Probit Regresyon, Logistik Regresyon, Klasifikasyon and Regresyon Ağaçları, Yapay Sinir Ağları, Geçerlilik Teknikleri , Doğruluk Rasyosu

To my family

ACKNOWLEDGMENTS

It is pleasure to express my appreciation to those who have influenced this study. I am grateful to Prof. Dr. Hayri KÖREZLİOĞLU and Assist. Prof. Dr. Kasırga YILDIRAK for their encouragement, guidance, and always interesting correspondence.

I am indebted to Prof. Dr. Gerhard Wilhelm WEBER for his help, valuable comments and suggestions. I am also very grateful to Assist. Prof.Dr. Hakan ÖKTEM for all his efforts.

My special thanks are to my mother and my dear brother without whose encouragement and supports this thesis would not be possible.

I acknowledge my debt and express my thanks to Oktay Sürücü for his endless support and being with me all the way.

As always it is impossible to mention everybody who had an impact to this work: Yeliz Yolcu Okur, Zehra Ekṣi, İrem Yıldırım, Serkan Zeytun, Tuḡba Yıldız, Zeynep Alparslan, Nejla Erdoḡdu, Derya Altıntan etc.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In a credit granting procedure, a credit company's main aim is to determine whether a credit application should be granted or refused. The credit scoring procedures in fact measure the risk on lending. From the early civilizations this risk has been assessed by the interest rate on it. However, the studies on the financial situations of the governments, companies and individuals has demonstrated that the interest does not diminish the risk. The credit risk should be assessed separately.

The default of a firm is always very costly for both shareholders and credit agencies. Because the credit agencies could lose whatever they give and the shareholders could lose all or nearly all of their value of equity. Here, the problem is to learn default some time before the default in order to be take some precautions. The empirical studies indicated that classification methods gives signals of defaults. However, these methods could act differently according to size, shape, and structure of the data. Therefore, selection of the most suitable method for available data is a much more complex concern.

In literature, as far as we know, the scoring studies shows only accuracy comparisons of two or more models. The close examinations of the methods are gaps in this area. Therefore, this study includes the empirical research on logistic regression. The logistic regression because of its environment is the most

widely used method in studies of credit scoring. However, the conditions in which the logistic regression performs well is not discussed. This study makes a close examination of the parameter estimation of logistic regression in different variable sets and the data sets with different default levels by Monte Carlo simulations.

Moreover, the study presented here also includes the applications of classification methods to the real Turkish credit data. The accuracies of all methods are compared with each other. Furthermore, for Turkish data sets which are very volatile in companies from the different group of companies, the most appropriate model is tried to be selected.

The organization of the thesis is as follows. Chapter 2 gives a brief overview of the development of credit scoring, the related studies and the explanatory variables in the studies. Chapter 3 provides the fundamentals of statistical methods used in credit scoring. In Chapter 4, the important neural network learning algorithms and their derivations are given. Chapter 5 presents the Monte Carlo applications on the parameter estimation and estimation accuracies of logistic regression. Moreover, Chapter 6 provides the accuracy ratio and validations of the methods mentioned in Chapter 3 and Chapter 4. Chapter 7 concludes this thesis.

CHAPTER 2

FOUNDATIONS OF CREDIT SCORING: DEVELOPMENT, RESEARCH AND EXPLANATORY VARIABLES

The research in the area of credit scoring started in the 1930's. After that date, many different works and methods have entered the literature of credit scoring. According to type of methods, we can split the period 1930-2005 into 4 sub-periods. We call the first period as a ***primitive age of credit scoring*** because this includes very basic applications. In this part, research was based only on a ratio analysis. In those years, scientists compared ratios of default and non-default companies and tried to develop an idea of companies' financial performances. As it can be guessed, these type of methods had no predictive power and so they were not very suitable.

The second period of the credit scoring started at 1966 with the application of *discriminant analysis*. By this application, research gained predictive power. However, this method has very strong assumptions on variables and so, the prediction power is not very high. Moreover, this method does not give the idea

of relative performances of the variables. We call this period as ***discriminant age***.

The application of discriminant analysis is a turning point for credit scoring because it opened the door for computer-based methods. After the 1970's, the methods that applied to this area changed rapidly. The main types of methods were the *regression based approaches*. Therefore, we can call this period as ***regression age of credit scoring***. The linear regression was applied firstly, but it did not give good results. Because the credit default probabilities takes only values between 0 and 1, but linear regression can give the results between $-\infty$ and ∞. Then, secondly, *probit regression* came into play. Since it also has strong assumptions of normality, the end of the application of this methods came rapidly. In other words, in the period 1970-1980, the regression type methods were not gone to the fore of discriminant analysis.

In the 1980's, the study of logistic regression increased the interest to the regression since it has no normality assumptions on variables, allows predictions, and interpretation of coefficients, and it gives the output on the interval [0,1]. Although after the 1980's many other statistical methods have been also applied such as *k-nearest neighborhood*, *classification and regression trees*, s*urvival analysis*, etc, this method has kept its importance even nowadays as the most widely-used statistical technique in research.

The year 1990 is an another turning point for credit scoring. In this year, the statistical methods gave their place to the machine learning type methods with the application *neural networks*. Therefore, we name this period as ***machine age***.

2.1 Primitive Age Researchers and Explanatory Variables

The first researchers which we found in the *primitive age* are Ramser and Foster with their 1931 paper [BLSW96]. This was followed by Fitzpatrick in 1932 [BLSW96]. Fitzpatrick investigated nineteen pairs of failed and non-failed companies. He showed a significant difference in the ratios of failed and non-failed companies at least three years prior to failure. After then, Winekor and Smith [BLSW96] in 1935, searched the mean ratios of failed firms ten years prior to failure and detected the breakdown in the mean values when failure was coming [B66].

In 1942, Merwin [BLSW96] studied mean ratios of the failed and non-failed companies in the period 1926-1936 and his result was only differing form Winekor and Smith' work [BLSW96] in the six year before failure [B66].

In these years, the primary ratio was the current ratio. However, some other ratios were also used. For example, Ramser and Foster [BLSW96] studied with *equity / (net sales)*; Fitzpatrick [BLSW96] used *equity / fixed-asset* and *return on stock*; Winekor and Smith [BLSW96] applied its analysis on *(working capital) / (total assets)*; and Mervin [BLSW96] beside *current ratio*, used *total debt/equity*, *(working capital) / (total assets)*.

2.2 Discriminant Age Researchers and Explanatory Variables

The researcher who started the *discriminant age* is Beaver [B66]. In his work, he applied a univariate type discriminant analysis by using: *(cash flow) / (total debt), (current assets) / (current liabilities), (net income) / (total assets), (total debt) / (total assets), (working capital) / (total assets)* [B66]. Then, in 1968, Altman investigated a multivariate discriminant analysis by his famous z-score with 5 variables that are *1. (MV of equity) / (book value of debt), 2. (net sales / total assets), 3. (operating income) / (total assets), 4. (retained earnings) / (total assets), 5. (working capital) / (total assets)* [A68]. Altman obtained 94% and 97% classification accuracy among default and non-default firms, respectively and 95% overall accuracy [HM01].

The discriminant analysis applications were also continued after 1970's. In 1972, Deakin [D72] applied discriminant analysis. In his analysis, he used: *cash / (current liabilities), (cash flow) / (total debt), cash / (net sales), cash / (total assets), current ratio, (current assets) / (net sales), (current assets) / (total assets), (net income) / (total assets), (quick assets) / (current liabilities), (quick assets) / (net sales), (quick assets) / (total assets), (total debt) / (total assets), (working capital) / (net sales) and (working capital) / (total assets)* and obtained 97% overall accuracy [D72] [HM01].

In 1972, Lane and Awh and then, Waters [RG94] in 1974, used the discriminant analysis. Moreover, in 1974, Blum with ratios: *market rate of return, quick ratio, (cash flow) / (total debt), fair market value of net worth, (net quick assets) / (inventory), (book value of net worth) / (total debt), (standard deviation of*

income), (standard deviation of net quick assets) / inventory, slope of income, (slope of net quick assets) / (inventory), trend breaks of income, (trend breaks of net quick assets) / (inventory) applied discriminant analysis [Bl74].

One year later, Sinkey used: *(cash+U.S. treasury security) / (assets), (loans) / (assets), (provision for loan losses) / (operationg expense), (loans) / (capital + reserves), (operating expense) / (operating income), (loan revenue) / (total revenue), (U.S. treasury securities' revenue) / (total revenue), (state and local obligations' revenue) / (total revenue), (interest paid on deposits) / (total revenue), (other expenses) / (total revenue)* [S75]. Then, Altman and Lorris [BLSW96] (1976) acquire 90% classification accuracy with the help of five financial ratios that are

- *(net income) / (total assets),*
- *(total liabilities + subordinate loans) / equity,*
- *(total assets) / (adjusted net capital),*
- *(ending capital-capital additions) / (beginning* capital)
- *scaled age*
- *composite version of the before mentioned ones.*

For closer information, we refer to [AL76] and [HM01].

Another paper with discriminant analysis was published by Altman, Haldeman and Narayan (1977) [BLSW96]. Here, *(retained earnings) / (total assets), (earnings before interest and taxes) / (total interest payments), (operating income) / (total assets), (market value of equity) / (book value of debt), (current*

assets) / (current liabilities) were their ratios. And their results showed a 93 % overall accuracy [HM01].

The investigations which I were able to reach were from the last researchers who made their studies only with discriminant analysis. They are Dambolena and Khory (1980) [DK80]; their ratios: *1. (working capital) / (total assets), 2. (retained earnings) / (total assets), 3. earning before interest and taxes to total assets, 4. (market value of equity) / (book value of debt), 5. sales to total assets,* Altman and Izan (1984) [HM01] and lastly, Pantalone and Platt (1987) [HM01] with 95% accuracy.

2.3 Regression Age Researchers and Explanatory Variables

The first researcher who used regression analysis according to my study was Orgler (1970) [Org70]. In his analysis, Orgler basically used: *current ratio, working capital, cash / (current liabilities), inventory / (current assets), quick ratio, (working capital) / (current assets), (net profit) / sales, (net profit) / (net worth), (net profit) / (total assets), net profit $\leqslant$ 0, net profit, (net worth) / (total liabilities), (net worth) / (fixed assets), (net worth) / (long-term debt), net worth $\leqslant$ 0, sales / (fixed assets), sales / (net worth), sales / (total assets), sales / inventory and sales / receivables.* He, in the hold-out sample, was only able to classify 75 % of the bad loans as bad and 35 % of the good loans as good [Org70].

In 1976, Fitzpatrick applied multivarite regression also. This research was followed by Olhson (1980) [O80]. In his paper, he investigated a logistic regression analysis. He collected data from the period 1970-76. Besides the basic ratios,

that are: *(total liabilities) /(total assets), (working capital) / (total assets), (current liabilities) / (current assets), (total liabilities-total assets) > 0, it takes one, otherwise it takes zero, (net income) / (total assets), (funds provided by operations) / (total liabilities), dummy; One if net income negative for the last two years,* $(NI_t - NI_{t-1})/(|NI_t| + |NI_{t-1}|)$, he also used *size of the company* as an explanatory variable [O80]. He calculated the type one and type two types of errors in different cut points and found for his second model better average error that is 14.4%.

Then, Pantalone and Platt (1987) [HM01] tried logistic regression in their paper. They obtained 98% accuracy in the classification of failed firms and 92% accuracy in that of non-failed firms .

In this time period, the recursive partitioning algorithm was gained to the literature by Altman, Frydman and Kao (1985) [FAK85]. Their explanatory variables were as follows:

- *(net income) / (total assets),*
- *(current assets)/ (current liabilities),*
- *log(total assets)*,
- *(market value of equity)/ (total capitilazation)*,
- *(current assets) / (total assets)*,
- *(cash flow) / (total debt)*,
- *(quick assets) / (current liabilities)*,
- *(earning before interest and taxes) / (total assets)*,

- *log(interest coverage+15)*,
- *cash / (total sales)*,
- *(total debt) / (total assets)*,
- *(quick assets) / (total assets)* .

For a closer information please look at [FAK85].

2.4 Machine Age Researchers and Explanatory Variables

Odom and Sharda in 1990 [OS90] made a comparison between the discriminant analysis and neural networks by using Altman's (1968) explanatory variables [A68]. They collected the data set in the period 1975 to 1982. Their training sample consisted of 74 companies 38 of which were default and hold-out sample was constituted by 55 companies 27 of which default. They concluded that neural networks performed better with respect to both training sample and hold-out sample. Moreover, the study proved that neural networks were more robust than discriminant analysis even in small sample sizes [OS90].

In 1991, Cadden [BO04] and Coats and Fant [BO04] made a comparison of discriminant analysis and neural networks also. After that in the following year, Tam and Kiang [TK92] applied discriminant analysis, logistic regression and neural networks with eighteen explanatory variables. This was followed by another study of Coats and Fant [CF93] in 1993. In the study, Coats and Fant obtained the data from the 1970 to 1989. By using Altman's (1968) variables,

they run discriminant analysis and neural networks and get that the discriminant analysis gave the highest misclassification error that is classifying a default as non-default.

In 1996, Back, Laitinen, Sere and Wesel [BLSW96] worked with a huge set of variables that are: *1. cash /(current liabilities), 2. (cash flow) / (current liabilities), 3. (cash flow) / (total assets), 4. (cash flow) / (total debt), 5. cash / (net sales) 6. cash / (total assets), 7. (current assets) / (current liabilities), 8. (current assets) / (net sales), 9. (current assets) / (total assets), 10. (current liabilities) / equity, 11. equity / (fixed assets), 12. equity / (net sales), 13. inventory / (net sales), 14. (long term debt) / equity, 15. (market value of equity) / (book value of debt), 16. (total debt) / equity, 17. (net income) / (total assets), 18. (net quick assets) / inventory, 19. (net sales) / (total assets), 20. (operating income) / (total assets) 21. (earnings before interest and faxes) /(total interest payments), 22. (quick assets) / (current liabilities), 23. (quick assets) / (net sales), 24. (quick assets) / (total assets), 25. rate of return to common stock, 26. (retained earnings) / (total assets), 27. return on stock, 28. (total debt) / (total assets), 29. (working capital) / (net sales), 30. (working capital) / equity,* and *31. (working capital) / (total assets)* [BLSW96].

In 1998, Kiviluoto [K98] tried the discriminant analysis, neural networks by means of *operating margin, net income before depreciation, extraordinary items, net income before depreciation, extraordinary items of the previous year, equity ratio.* For closer details see [K98].

After that in 1999, Laitinen and Kankaanpaa [LK99] compared discriminant analysis, logistic regression, recursive partitioning, survival analysis and neural networks. The study's ratios were *cash to current liabilities, total debt to total*

assets, operating income to total assets. They examined all the methods from three years prior to failure. Moreover, in the total error, they found neural networks as best one year prior to failure and recursive partitioning as best two and three years prior to failure.

In 1999, Muller and Ronz [MR99] showed a different approach to credit default prediction. They implemented the semi parametric generalized partial linear models to this area. In the paper, the 24 variables were used but not specified .

In 2000, recursive partitioning, discriminant analysis and neural networks were attempted by McKee and Greenstein [MG00]. The ratios were as follows:

- *(net income) / (total assets),*
- *(current assets) / (total assets),*
- *(current assets) / (current liabilities),*
- *cash / (total assets),*
- *(current assets) / sales* and
- *(long-term debt) / (total assets).*

In 2001, Atiya [Ati01] used: the *1. (book value) / (total assets), 2. (cash flow) / (total assets), 3. price / (cash flow ratio), 4. rate of change of stock price, 5. rate of change of cash flow per share, 6. stock price volatility* and he investigated neural networks.

The time table of the studies and methods can be found in Table (2.1). For a close examination please we refer to it, and we abbreviate:

Method **Researchers**	**Year**	Ra.A.	DA	RA	LA	CART	SPR	NN
Ramser, Foster [BLSW96]	1931	X						
Fitzpatrick [BLSW96]	1932	X						
Winakor, Smith [BLSW96]	1935	X						
Merwin [BLSW96]	1942	X						
Beaver [B66]	1966		X					
Mears [M66]	1966	X						
Horrigan [H66]	1966	X						
Neter [N66]	1966	X						
Altman [A68]	1968		X					
Orgler [Org70]	1970			X				
Wilcox [Wil71]	1971	X						
Deakin [D72]	1972		X					
Lane [RG94]	1972		X					
Wilcox [Wil73]	1973	X						
Awh, Waters [RG94]	1974		X					
Blum [Bl74]	1974		X					
Sinkey [S75]	1975		X					
Libby [Lib75]	1975	X						
Fitzpatrick [RG94]	1976			X				
Altman and Lorris [BLSW96]	1976		X					
Altman, Haldeman, Narayan [BLSW96]	1977		X					
Dambolena, Khoury [DK80]	1980		X					
Olhson [O80]	1980				X			
Altman, Izan [BO04]	1984		X					
Altman, Friedman, Kao [FAK85]	1985					X		
Pantalone, Platt [BO04]	1987		X					
Pantalone, Platt [BO04]	1987				X			
Odom, Sharda [BO04]	1990		X					X

Method **Researchers**	**Year**	Ra.A.	DA	RA	LA	CART	SPR	NN
Cadden [BO04]	1991		X					X
Coats, Fant [BO04]	1991		X					X
Tam, Kiang [TK92]	1992		X		X			X
Coats, Fant [CF93]	1993		X					X
Fletcher, Goss [BO04]	1993				X			X
Udo [BO04]	1993				X			X
Chung, Tam [BO04]	1993							X
Altman, Marco, Varetto [BO04]	1994		X					X
Back, Laitinen, Sere, Wesel [BLSW96]	1996		X		X			X
Bardos, Zhu [BO04]	1997		X		X			X
Pompe, Feelders [PF97]	1997		X			X		X
Kivilioto [K98]	1998		X					X
Laitinen, Kankaanpaa [LK99]	1999		X		X	X		X
Muller, Ronz [MR99]	1999				X		X	
Mckee, Greenstein [MG00]	2000		X			X		X
Pompe, Bilderbeek [BO04]	2000		X					X
Yang, Temple [BO04]	2000				X			X
Neophytou, Mar-Molinero [BO04]	2001				X			X
Atiya [Ati01]	2001							X

Table 2.1: The methods and scientists.

Ra.A: Ratio analysis,

DA: Discriminant analysis,

RA: Regression analysis,

LA: Logistic regression,

CART: Classification and regression trees,

SPR: Semiparametric regression,

NN: Neural networks.

CHAPTER 3

STATISTICAL METHODS IN CREDIT SCORING

3.1 Introduction

A credit institute faces with a prejudgement problem of measuring its customer firms' creditworthiness. For this reason, the credit institute primarily collects information about its customer firms' measurable *features*, or; namely, age of the firm, *(current assets) / (current liabilities)* ratio, and so on. Let $X_i \subseteq \Re$ represents the each feature of the customer firm, e.g., X_1 may be the age of the firm, X_2 may be the *current assets/current liabilities* ratio and so on. Then, each customer firm can be described by a tuple of p random variables, namely, by the vector $\mathbf{X} = (X_1, X_2, ..., X_p)$ which indicates a firm's completely all characteristic properties and market and internal performance features. Let the actual values of the variables for a particular customer firm be $\mathbf{x} = (x_1, x_2, ..., x_p) \in X \subseteq \Re^n$. Furthermore, let any different possible value of x_i to variable X_i be called an *attribute* of that feature.

Let us call the space of X as the input space and denote it by Ω since each customer is represented as a point in this space.

According to records of the credit institute, in market, there are two types

of firms: *default firms (D)* and *non-default firms (ND)*. Here, a default firm is a customer firm that did not fulfill its obligation in the past, and a non-default firm is a customer firm that fulfilled its obligation in the past. Moreover, the space of all possible outcomes that has only two elements: D and ND, is called the *output space* Y.

Moreover, let δ_{ND} be the *prior "non-default firms" category probability*, a δ_D stands for *prior class probability of "default firms"*.

According to our concern, objective is to find best scoring procedure $space(X) \longrightarrow space(Y)$ which splits the space Ω into two subspaces: Ω_{ND} and Ω_D, so that classifying new customer firms whose indicator vector belongs to the set Ω_{ND} as "non-default firm" and whose indicator vector belongs to the set Ω_D as "default firm" [HKGB99], [CET02].

In here, a brief review of credit scoring methods which are used in literature most commonly will be given. The above notations are giving to be throughout this chapter.

3.2 Discriminant Analysis

The discriminant analysis is a standard tool for classification. It is based on maximizing the between-group variance relative to the within-group variance.

3.2.1 Decision Theory Approach

According to the discrete or continuous character of the probability distributions, we refer to the discrete or the continuous case in the followings.

Discrete Case

Assume the companies which ask for credit feature vector has a finite number of discrete attributes so that Ω is finite and there is only a finite number of different attributes $\mathbf{x}$.

Suppose $p(\mathbf{x}|ND)$ represent the probability that a non-default firm will has an attribute $\mathbf{x}$. Similarly, $p(\mathbf{x}|D)$ represents the probability that a default firm will has an attribute $\mathbf{x}$. These conditional probabilities can be shown to be

$$p(\mathbf{x}|ND) = \frac{p(\text{firm is a non-default firm and has an indicator vector } \mathbf{x})}{p(\text{firm is a non-default firm})} \quad (3.2.1)$$

and

$$p(\mathbf{x}|D) = \frac{p(\text{firm is a default firm and has an indicator vector } \mathbf{x})}{p(\text{firm is a default firm})}. \quad (3.2.2)$$

Since in a market these conditional probabilities can not be observed directly, they will be obtained by using Bayes rule and directly observable probabilities. To apply Bayes rule, let us define $p(ND|\mathbf{x})$ as the probability that a company an attribute vector $\mathbf{x}$ as a non-default company and let us define $p(D|\mathbf{x})$ as the probability that a company an attribute vector $\mathbf{x}$ as a default company. Then,

$$p(ND|\mathbf{x}) = \frac{p(\text{firm is a non-default firm and has an indicator vector } \mathbf{x})}{p(\text{firm has an indicator vector } \mathbf{x} \,)}, \quad (3.2.3)$$

$$p(D|\mathbf{x}) = \frac{p(\text{company is a default firm and has an indicator vector } \mathbf{x})}{p(\text{firm has an indicator vector } \mathbf{x} \,)}. \quad (3.2.4)$$

Let $\gamma(\mathbf{x})$:= p(firm has an indicator vector $\mathbf{x}$), then (3.2.1), (3.2.3), (3.2.2) and

(3.2.4), respectively, can be put together in the following formulae:

$$p(\text{firm is a non-default firm and has an indicator vector } \mathbf{x})$$
$$= p(ND|\mathbf{x})\gamma(\mathbf{x}) = p(\mathbf{x}|ND)\delta_{ND}$$

and

$$p(\text{firm is a default firm and has an indicator vector } \mathbf{x})$$
$$= p(D|\mathbf{x})\gamma(\mathbf{x}) = p(\mathbf{x}|D)\delta_{D}.$$

Then, by Bayes rule,

$$p(ND|\mathbf{x}) = \frac{p(\mathbf{x}|ND)\delta_{ND}}{\gamma(\mathbf{x})} \tag{3.2.5}$$

and

$$p(D|\mathbf{x}) = \frac{p(\mathbf{x}|D)\delta_{D}}{\gamma(\mathbf{x})}. \tag{3.2.6}$$

Suppose a credit institute loses the amount $c(ND|D)$ of money for each firm if it classifies a default firm as non-default and loses $c(D|ND)$ amount of money for per firm if it classifies a non-default firm as default. These misclassification costs are summarized in Table (3.1).

		Classification Result	
		non-default	default
true value	non-default	0	$c(ND\|D)$
	default	$c(D\|ND)$	0

Table 3.1: Misclassification costs.

Furthermore, let us assume the probability that misclassifying a company as

Ω_{ND} be

$$p(\text{firm is misclassified as non-default}) = p(\mathbf{x}|D)\delta_D, \tag{3.2.7}$$

and the probability that misclassifying a non-default firm as Ω_D be

$$p(\text{firm is misclassified as default}) = p(\mathbf{x}|ND)\delta_{ND}. \tag{3.2.8}$$

Then, the expected cost of misclassifying firms if the firms with attributes belonging to the set Ω_{ND} are accepted and if the firms with attributes belonging to the set Ω_D are refused is

$$\begin{aligned} &c(D|ND)\sum_{\mathbf{x}\in\Omega_D} p(\mathbf{x}|ND)\delta_{ND} + c(ND|D)\sum_{\mathbf{x}\in\Omega_{ND}} p(\mathbf{x}|D)\delta_D = \\ &c(D|ND)\sum_{\mathbf{x}\in\Omega_D} p(ND|\mathbf{x})\gamma(\mathbf{x}) + c(ND|D)\sum_{\mathbf{x}\in\Omega_{ND}} p(D|\mathbf{x})\gamma(\mathbf{x}). \end{aligned} \tag{3.2.9}$$

At this point, the decision rule that minimizes this expected cost is clear. Let us consider cost of classifying a firm with $\mathbf{x} = (x_1, x_2, ..., x_p)$. If a firm puts into Ω_{ND}, then the only cost if it is a default is the expected cost $C(ND|D)p(\mathbf{x}|D)\delta_D$. If a firm puts into Ω_D if it is non-default, then the expected cost is

$$c(D|ND)p(\mathbf{x}|ND)\delta_{ND}. \tag{3.2.10}$$

Therefore, $\mathbf{x}$ can be classified into Ω_{ND} if

$$c(ND|D)p(\mathbf{x}|D)\delta_D \leq c(D|ND)p(\mathbf{x}|ND)\delta_{ND} \tag{3.2.11}$$

is satisfied. For this reason, the decision rule that minimizes the expected costs

is

$$\begin{aligned}\Omega_{ND} &= \{\mathbf{x}|c(ND|D)p(\mathbf{x}|D)\delta_D \leq c(D|ND)p(\mathbf{x}|ND)\delta_{ND}\} \\ &= \left\{\mathbf{x}|\frac{c(ND|D)}{c(D|ND)} \leq \frac{p(\mathbf{x}|ND)\delta_{ND}}{p(\mathbf{x}|D)\delta_D}\right\} \\ &= \left\{\mathbf{x}|\frac{c(ND|D)}{c(D|ND)} \leq \frac{p(ND|\mathbf{x})}{p(D|\mathbf{x})}\right\}. \end{aligned} \tag{3.2.12}$$

That is, we classify a firm as a non-default firm if the above condition is satisfied. Otherwise, classify the firm as a default firm.

Continuous Case

Let us assume that the indicator vector has a finite number of continuous type attributes so that Ω is finite and there is only a finite number of different attributes $\mathbf{x}$. The same procedure from the discrete case is applied to the continuous case. Here, the only difference is that the conditional probability mass functions $p(\mathbf{x}|ND)$ and $p(\mathbf{x}|D)$ are replaced by the *continuous* probability density functions. Then, the expected cost of misclassifying firms if the firms with attributes belonging to set Ω_{ND} are accepted and if the firms with attributes belonging to set Ω_D are refused, will become

$$c(D|ND)\int_{\mathbf{x}\in\Omega_D} f(\mathbf{x}|ND)\delta_{ND}d\mathbf{x} + c(ND|D)\int_{\mathbf{x}\in\Omega_{ND}} f(\mathbf{x}|D)\delta_D d\mathbf{x} \tag{3.2.13}$$

and the *decision rule* that minimizing this expected cost will become

$$\begin{aligned}\Omega_{ND} &= \{\mathbf{x}|c(ND|D)f(\mathbf{x}|D)\delta_D \leq c(D|ND)f(\mathbf{x}|ND)\delta_{ND}\} \\ &= \left\{\mathbf{x}|\frac{c(ND|D)\delta_D}{c(D|ND)\delta_{ND}} \leq \frac{f(\mathbf{x}|ND)}{f(\mathbf{x}|D)}\right\}. \end{aligned} \tag{3.2.14}$$

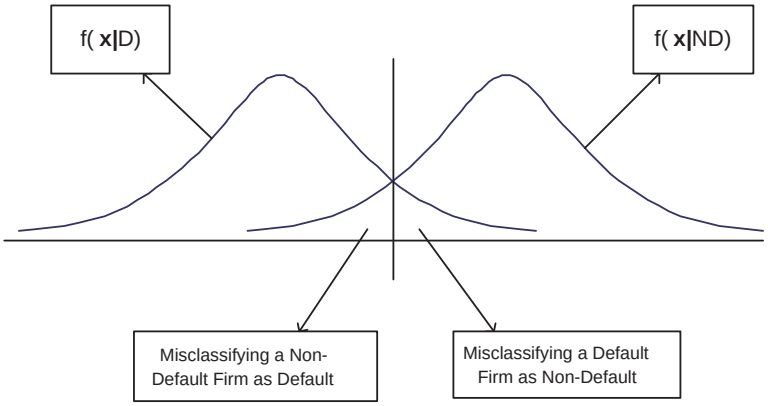

Figure 3.1: Misclassification errors [JW97].

That is, we classify a firm as a non-default firm if the above condition is satisfied. Otherwise, we classify the firm as a default Firm.

In literature, *normal distribution* is the most widely used distribution for discriminant analysis. When using a normal probability density function, three cases can be considered:

A. Univariate Normal Feature Variable x

This is the simplest case of discriminant analysis. In this case, firms are tried to classify with respect to only one feature variable. Let us say X = *(current assets) / (current liabilities)* ratio, and let it follow a normal distribution with mean μ_{ND} and variance σ^2 among the non-default firms, and follow a normal distribution with mean μ_D and variance σ^2 among the default firms. Then, the probability distribution functions can be written as follows:

$$f(x|ND) = (2\pi\sigma^2)^{-\frac{1}{2}} exp(\frac{-(x-\mu_{ND})^2}{2\sigma^2}) \tag{3.2.15}$$

for non-default firms and

$$f(x|D) = (2\pi\sigma^2)^{-\frac{1}{2}} exp(\frac{-(x-\mu_D)^2}{2\sigma^2}) \tag{3.2.16}$$

for default firms.

Then, according to continuous case results, the decision rule for univariate normal case of minimizing this expected cost will become

$$\begin{aligned}
\Omega_{ND} &= \left\{ x | \frac{c(ND|D)\delta_D}{c(D|ND)\delta_{ND}} \leq \frac{f(x|ND)}{f(x|D)} \right\} \\
&= \left\{ x | \frac{c(ND|D)\delta_D}{c(D|ND)\delta_{ND}} \leq \frac{(2\pi\sigma^2)^{-\frac{1}{2}} \exp(\frac{-(x-\mu_{ND})^2}{2\sigma^2})}{(2\pi\sigma^2)^{-\frac{1}{2}} \exp(\frac{-(x-\mu_D)^2}{2\sigma^2})} \right\} \\
&= \left\{ x | \exp(\frac{-(x-\mu_{ND})^2 + -(x-\mu_D)^2}{2\sigma^2}) \geq \frac{c(ND|D)\delta_D}{c(D|ND)\delta_{ND}} \right\} \\
&= \left\{ x | x(\mu_{ND} - \mu_D) \geq \frac{\mu_{ND}^2 + \mu_D^2}{2} + \sigma^2 log\left(\frac{c(ND|D)\delta_D}{c(D|ND)\delta_{ND}} \right) \right\}
\end{aligned} \tag{3.2.17}$$

That means, in our continuous case, we classify a firm as a non-default firm if the above condition is satisfied. Otherwise, we classify a firm as default firm [CET02].

B. Multivariate Normal Feature Vector x with Common Covariance Matrix Σ

In this case, the firms are tried to classify according to p variate feature indices $\mathbf{x} = (x_1, x_2, ..., x_p) \in X \subseteq \Re^n$. It is assumed that $\mathbf{x}$ follows a normal distribution with mean μ_{ND} and variance-covariance matrix Σ among non-default firms and follows a normal distribution with mean μ_D and variance-covariance matrix Σ

among default firms.The probability density function is

$$f(\mathbf{x}|ND) = (2\pi)^{-\frac{p}{2}}(det\Sigma_{ND})^{-\frac{1}{2}}exp(\frac{-(\mathbf{x}-\mu_{ND})^T\Sigma_{ND}^{-1}(\mathbf{x}-\mu_{ND})}{2}). \quad (3.2.18)$$

Similarly, as we learned it in A, our decision rule will become

$$\begin{aligned}
\Omega_{ND} &= \left\{x|\frac{c(ND|D)\delta_D}{c(D|ND)\delta_{ND}} \leq \frac{f(x|ND)}{f(x|D)}\right\} \\
&= \left\{\mathbf{x}|exp\{-\frac{1}{2}((\mathbf{x}-\mu_{ND})\Sigma_N D^{-1}(\mathbf{x}-\mu_{ND})^T - (\mathbf{x}-\mu_D)\Sigma_D^{-1}(\mathbf{x}-\mu_D)^T)\}\right. \\
&\geq \left.\frac{c(ND|D).\delta_D}{c(D|ND)\delta_{ND}}\right\} \\
&= \left\{\mathbf{x}|\mathbf{x}(\Sigma_{ND}^{-1}-\Sigma_D^{-1})\mathbf{x}^T + 2\mathbf{x}(\Sigma_{ND}^{-1}\mu_{ND}^T - \Sigma_D^{-1}\mu_D^T)\right. \\
&\geq \left.(\mu_{ND}\Sigma_{ND}^{-1}\mu_{ND}^T - \mu_D\Sigma_D^{-1}\mu_D^T) + 2log(\frac{c(ND|D)\delta_D}{c(D|ND)\delta_{ND}})\right\}. \quad (3.2.19)
\end{aligned}$$

3.2.2 Functional Form Approach

This approach is also known as *Fisher's discriminant function analysis* after Fisher, 1936 [JW97]. In his work, he tried to fit a linear discriminant function of feature variables that best splits the set into two subsets (see Figure 3.2 for an impression). The Fisher's discriminant function consists of combination of feature variables.

Let $Y = w_1X_1 + w_2X_2 + \ldots + w_pX_p$ be any linear combination of credit performance measure $\mathbf{X} = (X_1, X_2, \ldots, X_p)$. Like *analysis of variance* (ANOVA), the Fisher's discriminant function analysis uses the differences of the mean values of Y in the two subspaces: the space of default firms and space of non-default firms as a splitting criteria. In this analysis, therefore, the weights of X_i's $(i = 1, 2, ..., p)$ are such that they minimize the distance between the sample means of default

and non-default firms over the square root of the common sample variance.

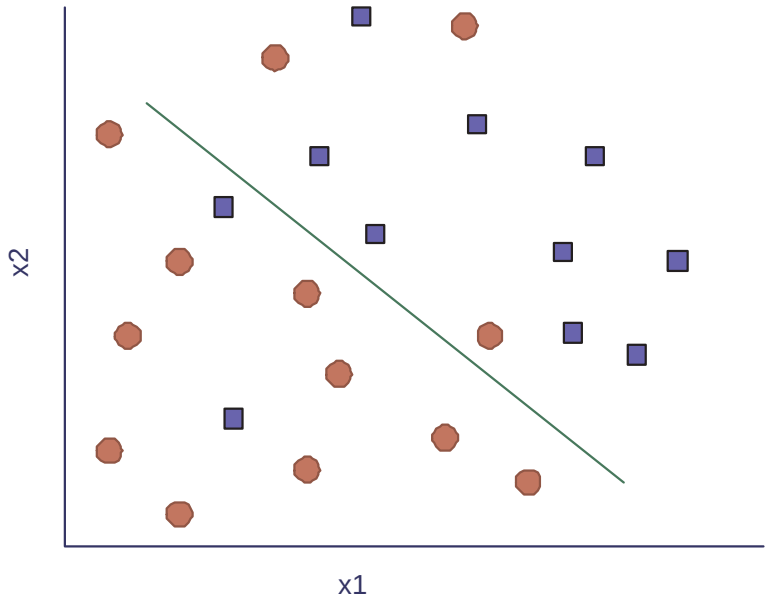

Figure 3.2: Fisher's linear discriminant analysis.

Fisher's Principle is an optimization problem which look as follows:

$$\min \quad J(\mathbf{w}) = \mathbf{w}^T \frac{(\mathbf{m}_{ND} - \mathbf{m}_D)^T}{(\mathbf{w}\mathbf{S}\mathbf{w}^T)^{1/2}}, \tag{3.2.20}$$

where $\mathbf{m}_{ND}$ and $\mathbf{m}_D$ are sample means vectors for non-default and default companies, respectively, and $\mathbf{S}$ is the common variance-covariance matrix.

Differentiating (3.2.20) with respect to $\mathbf{w}$ and setting it to 0 derives the following equation:

$$\frac{\mathbf{m}_{ND}^T - \mathbf{m}_D^T}{(\mathbf{w}\mathbf{S}\mathbf{w}^T)^{1/2}} - \frac{(\mathbf{w}(\mathbf{m}_{ND} - \mathbf{m}_D)^T)(\mathbf{S}\mathbf{w}^T)}{(\mathbf{w}\mathbf{S}\mathbf{w}^T)^{3/2}} = 0 \tag{3.2.21}$$

$$(\mathbf{m}_{ND} - \mathbf{m}_D)^T(\mathbf{wSw}^T) = (\mathbf{Sw}^T)(\mathbf{w}(\mathbf{m}_{ND} - \mathbf{m}_D)^T). \tag{3.2.22}$$

Since $\frac{\mathbf{wSw}^T}{w(\mathbf{m}_{ND}-\mathbf{m}_D)^T}$ is a constant, (3.2.22) results in

$$\mathbf{w}^T \propto (\mathbf{S}^{-1}(\mathbf{m}_{ND} - \mathbf{m}_D)^T). \tag{3.2.23}$$

For a closer information please look [CET02] [B04] [JW97].

3.2.3 Advantages and Disadvantages of Discriminant Analysis

Advantages:

Discriminant analysis has the following advantages:

- dichotomous response variable,
- easy to calculate,
- yields the input needed for an immediate decision, and
- reduced error rates.

Disadvantages:

Discriminant analysis has the following disadvantages:

- normality assumption on variables,
- approximately equal variances in each group,
- assumption on equivalent correlation patterns for groups,

- problem of multi-collinearity, and
- sensitivity to the outliers.

.

3.3 Linear Regression

3.3.1 Introduction

The linear regression is a statistical technique for investigating and modelling the linear relationships between variables. The probability of default is defined by the following form of linear regression:

$$w_0 + w_1 X_1 + w_2 X_2 + \ldots + w_p X_p = \mathbf{w}^* \mathbf{X}^{*\mathbf{T}}, \tag{3.3.24}$$

where $\mathbf{w}^* = (w_0, w_1, w_2, \ldots, w_p)$ and $\mathbf{X}^* = (X_0, X_1, X_2, \ldots, X_p)$.

Suppose $p(\mathbf{x}_i)$ defines the probability of default for the i^{th} individual company, then,

$$p(\mathbf{x}_i) = w_0 + w_1 x_{i1} + w_2 x_{i2} + \ldots + w_p x_{ip} + \varepsilon_i. \tag{3.3.25}$$

The linear regression has some primary assumptions, namely:

- The relationship between probability of default and explanatory variables is linear, or at least it is well approximated by a straight line.
- The error term ε has zero mean.
- The error term ε has constant variance.

- The errors are uncorrelated.
- The errors are normally distributed.

Suppose n_D of the training set are default companies and n_{ND} ones are non-default companies. We denote the default and non-default companies by "1" and "0", respectively. That is, $p(\mathbf{x}_i) = 1$ when $i = 1, 2, ..., n_D$, and $p(\mathbf{x}_i) = 0$ when $i = n_D + 1, n_D + 2, ..., n_D + n_{ND}$, and $n = n_D + n_{ND}$.

Then, our aim is to find the best set of weights, i.e., the one which satisfies

$$\min_{w} \sum_{i}^{n} \varepsilon_i^2. \tag{3.3.26}$$

By (3.3.25), this results to minimize

$$\sum_{i=1}^{n_D} \left(1 - \sum_{j=0}^{p} w_j x_{ij}\right)^2 + \sum_{i=n_D+1}^{n_D+n_{ND}} \left(\sum_{j=0}^{p} w_j x_{ij}\right)^2, \tag{3.3.27}$$

where $x_{i0} = 1$ for $i = 1, 2, ..., n$.

By matrix notation,

$$\begin{pmatrix} \mathbf{1}_D & \mathbf{X}_D \\ \mathbf{1}_{ND} & \mathbf{X}_{ND} \end{pmatrix} \begin{pmatrix} w_0 \\ \mathbf{w}^T \end{pmatrix} = \begin{pmatrix} \mathbf{1}_D \\ \mathbf{0} \end{pmatrix} \tag{3.3.28}$$

or

$$\mathbf{X}\mathbf{w}^{\mathbf{T}} = \mathbf{y}^{\mathbf{T}}. \tag{3.3.29}$$

Here,

$$\mathbf{X} := \begin{pmatrix} \mathbf{1}_D & \mathbf{X}_D \\ \mathbf{1}_{ND} & \mathbf{X}_{ND} \end{pmatrix}$$

being an $(n_D \times (p+1))$-matrix,

$$\mathbf{X}_D = \begin{pmatrix} x_{11} & \cdots & \cdots & x_{1p} \\ x_{21} & \cdots & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n_D 1} & \cdots & \cdots & x_{n_D p} \end{pmatrix}$$

being an $(n_D \times p)$-matrix,

$$\mathbf{X}_{ND} = \begin{pmatrix} x_{n_D+11} & \cdots & \cdots & x_{n_D+1p} \\ \vdots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ x_{n_D+n_{ND}1} & \cdots & \cdots & x_{n_D+n_{ND}p} \end{pmatrix}$$

is an $(n_{ND} \times p)$ matrix, and

$$\mathbf{y^T} = \begin{pmatrix} \mathbf{1}_D \\ \mathbf{0} \end{pmatrix}.$$

Moreover, $\mathbf{1}_D$ and $\mathbf{1}_{ND}$ are the $(1 \times n_D)$- and $(1 \times n_{ND})$-vectors with entries 1, respectively.

Then, in matrix notation our problem in the equation (3.3.27) become the following:

$$\min \quad (\mathbf{Xw^T} - \mathbf{y^T})^T(\mathbf{Xw^T} - \mathbf{y^T}) \tag{3.3.30}$$

To treat the problem (3.3.30), we set the derivative of it with respect to $\mathbf{w}$ equal to 0, i.e.,

$$\mathbf{X^T}(\mathbf{Xw^T} - \mathbf{y^T}) = 0 \quad \text{or} \quad \mathbf{X^T X w^T} = \mathbf{X^T y^T}. \tag{3.3.31}$$

These equations are called *normal equations*. For solving them we study the system

$$\begin{pmatrix} \mathbf{1}_D & \mathbf{1}_{ND} \\ \mathbf{X}_D & \mathbf{X}_{ND} \end{pmatrix} \begin{pmatrix} \mathbf{1}_D & \mathbf{X}_D \\ \mathbf{1}_{ND} & \mathbf{X}_{ND} \end{pmatrix} \begin{pmatrix} w_0 \\ \mathbf{w}^T \end{pmatrix} = \begin{pmatrix} \mathbf{1}_D & \mathbf{1}_{ND} \\ \mathbf{X}_D & \mathbf{X}_{ND} \end{pmatrix} \begin{pmatrix} \mathbf{1}_D \\ \mathbf{0} \end{pmatrix} \tag{3.3.32}$$

and

$$\begin{pmatrix} n & n_D\mu_{\mathbf{D}} + n_{ND}\mu_{\mathbf{ND}} \\ n_D\mu_{\mathbf{D}}^{\mathbf{T}} + n_{ND}\mu_{\mathbf{ND}}^{\mathbf{T}} & \mathbf{X}_{\mathbf{D}}^{\mathbf{T}}\mathbf{X}_{\mathbf{D}} + \mathbf{X}_{\mathbf{ND}}^{\mathbf{T}}\mathbf{X}_{\mathbf{ND}} \end{pmatrix} \begin{pmatrix} w_0 \\ \mathbf{w}^T \end{pmatrix} = \begin{pmatrix} n_D \\ n_D{\mu_{\mathbf{D}}}^T \end{pmatrix}, \tag{3.3.33}$$

where $\mu_{\mathbf{D}}$ and $\mu_{\mathbf{ND}}$ indicate the mean vectors of explanatory variables for default and non-default companies, respectively.

Let us think that the learning set expectations as actual expectations and assumptions hold. Then, we get

$$\begin{aligned} \mathbf{X}_{\mathbf{D}}^{\mathbf{T}}\mathbf{X}_{\mathbf{D}} + \mathbf{X}_{\mathbf{ND}}^{\mathbf{T}}\mathbf{X}_{\mathbf{ND}} &= nE[X_iX_j] \qquad (3.3.34) \\ &= nCov(X_iX_j) + n_D\mu_{\mathbf{D}}\mu_{\mathbf{D}}^{\mathbf{T}} + n_{ND}\mu_{\mathbf{ND}}\mu_{\mathbf{ND}}^{\mathbf{T}}. \end{aligned}$$

Let $\mathbf{C}$ denote the learning sample covariance function. Now, (3.3.34) becomes

$$\mathbf{X}_{\mathbf{D}}^{\mathbf{T}}\mathbf{X}_{\mathbf{D}} + \mathbf{X}_{\mathbf{ND}}^{\mathbf{T}}\mathbf{X}_{\mathbf{ND}} = n\mathbf{C} + n_D\mu_{\mathbf{D}}\mu_{\mathbf{D}}^{\mathbf{T}} + n_{ND}\mu_{\mathbf{ND}}\mu_{\mathbf{ND}}^{\mathbf{T}}. \tag{3.3.35}$$

By using equation (3.3.35) in equation (3.3.33), we obtain

$$\begin{aligned} nw_0 + (n_D\mu_{\mathbf{D}} + n_{ND}\mu_{\mathbf{ND}})\mathbf{w}^{\mathbf{T}} &= n_D, \qquad (3.3.36) \\ (n_D{\mu_{\mathbf{D}}}^T + n_{ND}\mu_{\mathbf{ND}}^{\mathbf{T}})w_0 + (n\mathbf{C} + n_D\mu_{\mathbf{D}}\mu_{\mathbf{D}}^{\mathbf{T}} + n_{ND}\mu_{\mathbf{ND}}\mu_{\mathbf{ND}}^{\mathbf{T}})\mathbf{w}^T &= n_D\mu_{\mathbf{D}}^{\mathbf{T}}. \end{aligned}$$

Then, substituting the first equation of (3.3.36) into the second one gives

$$\begin{aligned}((n_D\mu_{\mathbf{D}}{}^T + n_{ND}\mu_{\mathbf{ND}}^{\mathbf{T}})(n_D - (\mu_{\mathbf{D}} + n_{ND}\mu_{\mathbf{ND}})\mathbf{w^T})/n) & \\ +(n_D\mu_{\mathbf{D}}\mu_{\mathbf{D}}^{\mathbf{T}} + n_{ND}\mu_{\mathbf{ND}}\mu_{\mathbf{ND}}^{\mathbf{T}})\mathbf{w^T} + n\mathbf{C}\mathbf{w^T} &= n_D\mu_{\mathbf{D}}^{\mathbf{T}}.\end{aligned}$$

Hence,

$$\left(\frac{n_D n_{ND}}{n}\right)(\mu_{\mathbf{D}} - \mu_{\mathbf{ND}})\mathbf{w^T} + n\mathbf{C}\mathbf{w^T} = \left(\frac{n_D n_{ND}}{n}\right)(\mu_{\mathbf{D}} - \mu_{\mathbf{ND}})^T;$$

thus

$$\mathbf{C}\mathbf{w^T} = a(\mu_{\mathbf{D}} - \mu_{\mathbf{ND}})^T, \tag{3.3.37}$$

where a is a constant. This relation (3.3.37) gives the *optimal vector* of weights $\mathbf{w} = (w_0, w_1, w_2, \ldots, w_p)$.

3.3.2 Advantages and Disadvantages of Regression

Advantages:

As very important advantages of regression, we note:

- The estimates of the unknown parameters obtained from linear least squares regression are the optimal. Estimates from a broad class of possible parameter estimates under the usual assumptions are used for process modelling.
- It uses data very efficiently. Good results can be obtained with relatively small data sets.
- The theory associated with linear regression is well-understood and allows

for construction of different types of easily-interpretable statistical intervals for predictions, calibrations, and optimizations.

Disadvantages:

As the disadvantages of regression we state:

- Outputs of regression can lie outside of the range [0,1].
- It has limitations in the shapes that linear models can assume over long ranges.
- The extrapolation properties will be possibly poor.
- It is very sensitive to outliers.
- It often gives optimal estimates of the unknown parameters.

.

3.4 Probit Regression

3.4.1 Introduction

Probit regression is a tool for a dichotomous dependent variable. The term "probit" was used firstly in the 1930's by Chester Bliss and implies a probability unit [Probit].

If we denote probability of default as $p(\mathbf{x}_i) = E(Y|\mathbf{x}_i)$. Then, probit model is defined as:

$$p(x_i) = \Phi(\mathbf{x_i w}) = \Phi(w_0 + w_1 x_{i1} + w_2 x_{i2} + \ldots + w_p x_{ip}) \qquad (i = 1, 2, ..., n), \quad (3.4.38)$$

where Φ is the standard cumulative normal probability distribution and $\mathbf{w}$ is the weight vector.

Here, $\mathbf{xw}$ has the normal distribution and y_i follows a Bernolli distribution. Then, the estimation of the coefficients of a probit regression model can be made with the help of the *maximum likelihood estimation (MLE)*. For MLE estimation we should firstly write the likelihood function of the model. Let us denote the likelihood function by L, then,

$$L(\mathbf{w}) = \prod_{i=1}^{n} p(\mathbf{x_i})^{y_i} \big(1 - p(\mathbf{x_i})\big)^{1-y_i}. \tag{3.4.39}$$

By putting (3.4.38) in (3.4.41), the following is obtained:

$$L(\mathbf{w}) := \prod_{i=1}^{n} \Phi(\mathbf{x_i w})^{y_i} \big(1 - \Phi(\mathbf{x_i w})\big)^{1-y_i}. \tag{3.4.40}$$

By MLE, the log-version of likelihood function i.e., *log-likelihood function* will be used to make calculations easy. Log-likelihood function is denoted by ℓ and shown as:

$$\ell(\mathbf{w}) := l(\mathbf{w}) = \ln(L(\mathbf{w})) = \sum_{i=1}^{n} \big\{ y_i \ln(\Phi(\mathbf{x_i w})) + (1 - y_i) \ln(1 - \Phi(\mathbf{x_i w})) \big\}. \tag{3.4.41}$$

The maximum likelihood requires to maximizing the log-likelihood function by taking derivative of it with respect to $\mathbf{w}$. That results in to solve following

equations:

$$\frac{\partial l(\mathbf{w})}{\partial \mathbf{w_0}} = \sum_{i=1}^{n} \big(y_i - \Phi(\mathbf{x_i w})\big) = 0 \tag{3.4.42}$$

$$\frac{\partial l(\mathbf{w})}{\partial \mathbf{w_j}} = \sum_{i=1}^{n} x_{ij}\big(y_i - \Phi(\mathbf{x_i w})\big) = 0 \qquad (j = 1, 2, ..., p). \tag{3.4.43}$$

3.4.2 Advantages and Disadvantages of Logistic Regression

Advantages:

- Scores are interpretable.
- Probabilities help for decisions.
- It is easy to compute.

Disadvantages:

- Normality assumption violations can not be hold.
- Over- or under-estimation problems can occur.

3.5 Logistic regression

3.5.1 Introduction

Logistic regression is a form of regression which is used when the dependent variable is a binary or dichotomous and the independents are of any type. In any regression analysis, the main feature is to find the expected value of the

dependent variable under the known explanatory variables, i.e., $E(Y|\mathbf{x})$, where Y and $\mathbf{x}$ denote the dependent and the vector of explanatory variables, respectively [HL00].

Let us use the notation $p(\mathbf{x}_i) = E(Y|\mathbf{x}_i)$ being the probability of default for the i^{th} individual company. Then, the form of the logistic regression model is

$$p(\mathbf{x}_i) = G(\mathbf{x}_i, \mathbf{w}) = \frac{e^{w_0+w_1x_{i1}+w_2x_{i2}+\ldots+w_px_{ip}}}{1+e^{w_0+w_1x_{i1}+w_2x_{i2}+\ldots+w_px_{ip}}} = \frac{e^{\mathbf{x_i w}}}{1+e^{\mathbf{x_i w}}}. \tag{3.5.44}$$

The *logit transformation* of it can be written as

$$\ln\left(\frac{p(\mathbf{x_i})}{1-p(\mathbf{x_i})}\right) = w_0 + w_1x_{i1} + w_2x_{i2} + \ldots + w_px_{ip} + \varepsilon_i = \mathbf{x_i w} + \varepsilon_i. \tag{3.5.45}$$

Here, the estimation of the coefficients of a logistic regression model is done with the help of the *maximum likelihood estimation (MLE)*. MLE primarily states that the coefficients are estimated in a way in which the *likelihood function* is minimized. In order to obtain a likelihood function, we should firstly introduce the probability mass function of y_i. Since y_i follows a Bernolli distribution, the probability mass function for the i^{th} company can be written as

$$p(\mathbf{x_i})^{y_i}\big(1-p(\mathbf{x_i})\big)^{1-y_i}. \tag{3.5.46}$$

If we assume that all observations are independently distributed, the likelihood function expression will be

$$L(\mathbf{w}) := \prod_{i=1}^{n} p(\mathbf{x_i})^{y_i}\big(1-p(\mathbf{x_i})\big)^{1-y_i}. \tag{3.5.47}$$

However, the computations by using (3.5.47) are difficult, so we use logarithmic version of it that is called *log-likelihood function* and defined as

$$l(\mathbf{w}) := \ln(L(\mathbf{w})) = \sum_{i=1}^{n} \big\{ y_i \ln(p(\mathbf{x_i})) + (1 - y_i) \ln(1 - p(\mathbf{x_i})) \big\}. \tag{3.5.48}$$

To find the values of coefficients that maximizes (3.5.48) we differentiate (3.5.48) with respect to $\mathbf{w}$. This gives:

$$\frac{\partial l(\mathbf{w})}{\partial \mathbf{w_0}} = \sum_{i=1}^{n} \big(y_i - p(\mathbf{x_i}) \big) = 0, \tag{3.5.49}$$

$$\frac{\partial l(\mathbf{w})}{\partial \mathbf{w_j}} = \sum_{i=1}^{n} x_{ij} \big(y_i - p(\mathbf{x_i}) \big) = 0 \qquad (j = 1, 2, ..., p). \tag{3.5.50}$$

Since these equations are nonlinear in $\mathbf{w}$, it is not possible to solve them directly. Therefore, the solution of it is usually made with the well-known nonlinear optimization method called *Gauss-Newton algorithm*. For a closer information please see [HL00].

3.5.2 Advantages and Disadvantages of Logistic Regression

Advantages:

- Scores are interpretable in terms of log odds.
- Constructed probabilities have chance of being meaningful.
- It is modelled as a function directly rather than as ratio of two densities.
- It is a good default tool to use when appropriate, especially, combined with

feature creation and selection.

Disadvantages:

- It invites to an over-interpretation of some parameters.

For a closer look we refer to [M04].

3.6 Classification and Regression Trees

Classification and regression tree (CART) is a nonparametric pattern recognition based statistical classification technique [FAK85]. CART is a tool for analyzing both categorical and continuous dependent variables. At this point, we can split CART methodology into two parts: For categorized dependent variables it gives the name *classification tree* and for continuous dependent variables its name becomes *regression tree* [YW99]. However, in both cases, it produces a *binary* classification tree.

In fact, CART algorithm divides the whole space into rectangular regions and assigns each rectangular region to one of the classes. The sample organization chart of CART algorithm can be shown in Figure 3.3.

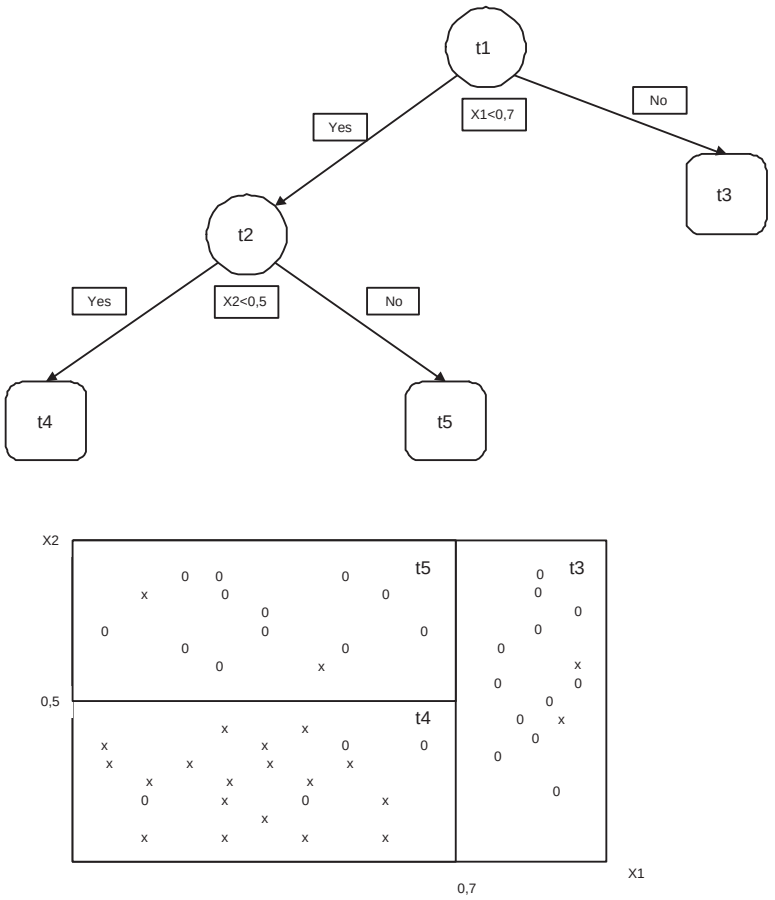

Figure 3.3: A Sample organization chart of classification and regression trees [BFOS84].

In our case, since we have a categorical dependent variable, we will only explain the classification trees.

A classification tree has three types of nodes. The top node is called the *root node* and contains all the observation in the sample. The second type of node consists of *terminal nodes* that are the end nodes assigned to one of the classes. The other nodes are called as *non-terminal nodes*.

The *CART algorithm* is a four-step classification *procedure*:

Step 1: *simple binary questions* of CART Algorithm,

Step 2: *goodness of split criterion*,

Step 3: *the class assignment rule* to the terminal nodes and resubstitution estimates,

Step 4: *selecting the correct complexity* of a tree.

To understand CART algorithm better, we firstly need to define some probabilities. The works [BFOS84], [YW99], [FAK85] give the following expressions:

Let N be the total learning sample size, and let N_{ND} be the number of non-default firms in our learning sample and N_D be the number of default firms.

Suppose CART has $t = 1, 2, ..., T$ nodes. In node t, there are $N(t)$ observations. Let $N_{ND}(t)$ represent the number of non-default firms in node t and similarly, $N_D(t)$ represents the number of default firms.

Let $p(ND, t)$ be the probability of a firm is non-default firm and falls into node t. Similarly, $p(D, t)$ be the probability of a firm is default firm and falls into node t. So, we can write these probabilities as

$$p(ND, t) = \delta_{ND} \frac{N_{ND}(t)}{N_{ND}}, \tag{3.6.51}$$

$$p(D, t) = \delta_D \frac{N_D(t)}{N_D}, \tag{3.6.52}$$

where δ_{ND}, δ_D stands for the prior class probabilities.

Moreover, the probability that an observation falls into node t is

$$p(t) = \sum_{j=ND}^{D} p(j, t). \tag{3.6.53}$$

By *Bayes* rule, the probability of a firm in the node t to be a non-default firm is denoted by $p(ND|t)$ and equal to

$$p(ND|t) = \frac{p(ND,t)}{p(t)}, \tag{3.6.54}$$

and the probability that a firm in the node t is a default firm is

$$p(D|t) = \frac{p(D,t)}{p(t)}. \tag{3.6.55}$$

The sum of (3.6.54), (3.6.55) probabilites satisfies the below equality:

$$\sum_{j=ND}^{D} p(j|t) = 1. \tag{3.6.56}$$

With the help of Bayes rule, we can observe the probabilities of $p(t|ND)$ and $p(t|D)$ from equations (3.6.51) to (3.6.56). Then,

$$p(t|ND) = \frac{p(ND,t)}{\delta_{ND}} = \frac{p(ND|t)p(t)}{\delta_{ND}}, \tag{3.6.57}$$

$$p(t|D) = \frac{p(D,t)}{\delta_{D}} = \frac{p(D|t)p(t)}{\delta_{D}}. \tag{3.6.58}$$

3.6.1 Simple Binary Questions of CART Algorithm

The tree-growing procedure of CART algorithm is based on the binary questions of type $\{\texttt{Is } x_i \leq c?\}$ for numerical values and $\{\texttt{Is } x_i = d?\}$ for the categorical values, where x_i is any variable in the feature vector $\mathbf{x} = (x_1, x_2, ..., x_p) \in X \subseteq \Re^n$. CART algorithm puts all observations into the root node and, then, with the help of these simple questions, it searches for the best split in order to

divide root node into binary nodes.

In CART, each split will be desired for only one variable. For this purpose, at each node, the algorithm tries all the variables $x_1, x_2, \ldots, x_n$. For each variable, it searches for the best split. Furthermore, selects the best split point and variable among these best splits for a particular node.

As an example according to [BFOS84],[JW97]; let x_5 be a categorical variable with 6 category form $1, 2, \ldots, 6$. Then, our set of questions will be `Is` $x_5 = 1$? ... `Is` $x_5 = 6$?. So, the CART should search $2^{6-1} - 1 = 31$ different splits for finding the best split for this single variable. For a numeric variable, let us say x_6 is in the range $[12, 34]$. Our possible questions will be `Is` $x_5 \leq 13$? ... `Is` $x_5 \leq 34$?. Herewith, if the number of questions is k then, the number of searched splits will be $2^{k-1} - 1$.

3.6.2 Goodness of Split Criterion

Introduction

The goodness of split criterion is applied to the each split point at a node in order to select best split point for each variables and then, for node. The goodness of split criterion is an index based on impurity functions.

Definition 3.1. [BFOS84] ***An impurity function*** is a function ϕ defined on the set of J classes with prior probability vector $\delta = (\delta_1, \delta_2, ..., \delta_J)$ satisfying $\delta_j \geq 0$, $(j = 1, 2, ..., J)$, $\sum_{j=1}^{J} \delta_j = 1$ with the following three properties:

(i) ϕ has a maximum only at the point $\delta = (\frac{1}{J}, ..., \frac{1}{J})$,

(ii) ϕ achieves its minimum only at the points $\delta = (1, 0, ..., 0)$, $\delta = (0, 1, 0, ..., 0)$, ..., $\delta = (0, 0, ..., 0, 1)$,

(iii) ϕ is a symmetric function of $\delta_1, \delta_2, ..., \delta_j$.

Definition 3.2. [BFOS84] Given an impurity function ϕ, define the ***impurity measure*** $i(t)$ of any node t as

$$i(t) := \phi(p(1|t), p(2|t), ..., p(J|t))$$

The CART basically for a split point divides sample left (l) and right (r) nodes, then applies a split criterion and looks the split's goodness.

In credit scoring literature, there are six most commonly used impurity measures can be found : *basic impurity index, gini index, Kolmogorov-Smirnov statistic, twoing index, entropy index* and *maximize half-sum of squares index.*

Basic Impurity Index

The basic impurity index is only based on the left(l) and right(r) node probabilities: $p(t_l)$ and $p(t_r)$ see (3.6.53) and impurities: $i(t_l)$ and $i(t_r)$.

Then, the change in the impurity at node t is defined as follows:

$$\Delta i(t, s) = i(t) - p(t_l)i(t_l) - p(t_r)i(t_r), \tag{3.6.59}$$

where s denotes the split and

$$i(j) = p(ND|j) \quad \text{if} \quad p(ND|j) \leq 0.5, \tag{3.6.60}$$

$$i(j) = p(D|j) \quad \text{if} \quad p(D|j) < 0.5. \tag{3.6.61}$$

If this change in the impurity is greater, then the node will be much more pure. That means we must select the split which maximizes this function [BFOS84], [CET02].

Gini Index

The Gini index is a quadratic impurity measure. The change in the impurity at node t is

$$\Delta i(t,s) := i(t) - p(t_l)i(t_l) - p(t_r)i(t_r), \tag{3.6.62}$$

where

$$i(j) := p(ND|j)p(D|j). \tag{3.6.63}$$

We refer to the [CET02], [BFOS84].

Kolmogorov-Smirnov Statistics

This impurity measure is based on the idea of maximizing the distance between probability distributions of non-default firms and default firms for a node.

Kolmogorov-Smirnov statistic is defined by

$$KS(s) := |p(t_l|D) - p(t_l|ND)| = \left| \frac{p(D|t_l)}{\delta_D} - \frac{p(ND|t_l)}{\delta_{ND}} \right|. \tag{3.6.64}$$

This implies that we select the split s which maximized the KS statistics [CET02].

Twoing Index

According to twoing Index, select the split s which maximizes the following measure [BFOS84]:

$$\Delta i(t,s) := \frac{p(t_l)p(t_r)}{4} \left(\sum_{j=D}^{ND} |p(j|t_l) - p(j|t_r)| \right)^2. \tag{3.6.65}$$

Entropy Index

The entropy index like Gini criterion is a nonlinear measure of impurity. The rule of entropy index is to

$$\max \quad \Delta i(t,s) := i(t) - p(t_l)i(t_l) - p(t_r)i(t_r), \tag{3.6.66}$$

where

$$i(j) := -p(ND|j)ln(p(ND|j)) - p(D|j)ln(p(D|j)). \tag{3.6.67}$$

Here, we refer to [CET02].

Maximize Half-Sum of Squares

Here, the index is

$$\Delta i(t,s) := n(t_l)n(t_r) - \frac{\big(p(ND|t_l) - p(ND|t_r)\big)^2}{n(t_l) + n(t_r)}, \tag{3.6.68}$$

where $n(t_l)$ and $n(t_r)$ are the total numbers of observations in the left and right nodes [CET02].

3.6.3 The Class Assignment Rule to the Terminal Nodes and Resubstitution Estimates

After selecting the variables with best splits for each node, a tree (let us say: T_{max}) will have been constructed. Let $\widetilde{T}$ present the terminal nodes.

Definition 3.3. [BFOS84] ***A class assignment rule*** assigns a class $j \in \{1, 2, ..., J\}$ to every terminal node $t \in \widetilde{T}$. The class assigned to node $t \in \widetilde{T}$

is denoted by $j(t)$.

The class assignment rule requires the minimization of cost of misclassification of a node one of the classes. In CART analysis, the observed expected cost of misclassification of each assignment is called the *resubstitution risk* [FAK85]. Let us denote the resubstitution risk of assigning a terminal node $t \in \widetilde{T}$ to j^{th} by $R_j(t)$.

Suppose that the misclassification cost of classifying a default firm as non-default be $c(ND|D) \geq 0$ and the misclassification cost of classifying a non-default firm as default be $c(D|ND) \geq 0$.

Then, the resubstitution risk of classifying the observation which falls into the terminal node t as a non-default firm can be obtained by

$$\begin{aligned} R_{ND}(t) &= c(ND|D)p(D,t) && (3.6.69)\\ &= c(ND|D)p(t|D)\delta_D && (3.6.70)\\ &= c(ND|D)\delta_D \frac{N_D(t)}{N_D} && (3.6.71) \end{aligned}$$

and, similarly,

$$R_D(t) = c(D|ND)\delta_{ND}\frac{N_{ND}(t)}{N_{ND}}. \quad (3.6.72)$$

Then, our class assignment rule $j(t)$ will be

$$j(t) = \begin{cases} ND & \text{if} \qquad R_D(t) \geq R_{ND}(t), \\ D & \text{otherwise.} \end{cases}$$

This means, if the resubstitution risk of assigning node t to the class of non-default firms is greater or equal to that of default firms assign node t to the class of default firms. Otherwise, we assign node t to the class of non-default firms.

3.6.4 Selecting the Correct Complexity of a Tree

The initial trees (T_{max}) includes more splits and more terminal nodes and so they are complex trees. Therefore, they have some problems: 1. Their resubstitution risks will always seen to be smaller, but they usually overfit the data. So, we are faced with poorer results of estimation with new data sets. 2. Another type of problem is interpretation of the complex trees. The larger the tree is, the more difficult to interpret it.

In order to faced with these problems, we would try to select the tree with correct complexity and smaller resubstitution risk.

In this part, we firstly consider all subtrees of the T_{max}. The resubstitution risk of the any tree can be computed by the formula

$$R(\widetilde{T}) = \sum_{t \in \widetilde{T}} R(t). \tag{3.6.73}$$

Then, for each tree T, we compute the following complexity measure

$$R(T) + K \times \quad (\text{number of terminal nodes of } T), \tag{3.6.74}$$

where K is a non-negative constant interpreted as a penalty for complex trees. Our *decision rule* is that for a given K, select the optimal tree T_{opt} which minimizes the complexity measure (3.6.74).

If $K = 0$, then, the optimal tree will be T_{max}. If $K > 0$, the optimal tree will be any subtree of T_{max}. And if K increases, then, the optimal tree's complexity will be less, but its resubstitution risk will be greater.

3.6.5 Advantages and Disadvantages of CART

Advantages:

- CART makes no distributional assumptions of any kind for dependent and independent variables.
- The explanatory variables can be a mixture of categorical and continuous.
- It is not at all affected by the outliers, collinearities, heteroskedasticity, or distributional error structures that affect parametric procedures. Outliers are isolated into a node and thus have no effect on splitting. Contrary to situations in parametric modeling, CART makes use of collinear variables in surrogate splits.
- CART has a built-in algorithm to deal with the missing values of a variable for a case, except when a linear combination of variables is used as a splitting rule.
- Furthermore, CART has the ability to detect and reveal variable interactions in the data set.
- It deals effectively with large data sets and the issues of higher dimensionality.
- It can handle noisy and incomplete data.
- CART is a user friendly method and it gives clear output.
- It is a simple procedure.

- The probability level or confidence interval associated with predictions derived from a CART tree could help to classify a new set of data.

Disadvantages:

- There can be errors in the specification of prior probabilities and misclassification costs.
- CART does not vary under a monotone transformation of independent variables.
- In CART, the relative importance of variables are unknown.
- It is a discrete scoring system.

For a closer information, we refer to [BO04], [YW99].

3.7 Semi-Parametric Binary Classification

Semi-parametric binary classification methods make *no* necessary assumption on the model and data sets like non-parametric models. However, at the same time, in contrast to non-parametric models, they allow extrapolations in some boundaries. Also they reduce the dimensionality of the parameter space to protect statistical accuracy of the model from sharp decreases.

Here, in this section, *generalized partial linear models* will be given.

In semi-parametric models *kernel density estimation* plays an important role; therefore, we firstly introduce it.

3.7.1 Kernel Density Estimation

Univariate Case

Suppose we have a univariate feature random variable X with n entities: X_1,X_2,...,X_n with an unknown continuous distribution.

Let h be the bandwidth. Then, the *kernel function* assigns weights to each observation X_i whose distance from a given x is not bigger than h.

A kernel function is denoted by $K(\cdot)$ and the density of x can be written as

$$\widehat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right), \tag{3.7.75}$$

please see Table 3.2 for some very important kernel functions.

Kernel	$K(u)$
Uniform	$\frac{1}{2}I(\lvert u\rvert \leq 1)$
Triangle	$(1 - \lvert u\rvert)I(\lvert u\rvert \leq 1)$
Epanechnikov	$\frac{3}{4}(1 - u^2)I(\lvert u\rvert \leq 1)$
Quartic Biweight	$\frac{15}{16}(1 - u^2)^2 I(\lvert u\rvert \leq 1)$
Triweight	$\frac{35}{32}(1 - u^2)^3 I(\lvert u\rvert \leq 1)$
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}u^2)$
Cosine	$\frac{\pi}{4}\cos(\frac{\pi}{2}u)I(\lvert u\rvert \leq 1)$

Table 3.2: Kernel functions [HMSW04].

The general form of the kernel density estimator of a probability density f, based on a random sample X_1,X_2,...,X_n from f, looks as follows [HMSW04]:

$$\widehat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h\Big(x - X_i\Big), \tag{3.7.76}$$

where

$$K_h(\cdot) = \frac{1}{n}K(\cdot|h). \tag{3.7.77}$$

The kernel density estimations of the car prices in the one of the *Matlab* data files were made by writing the code of (3.7.76) in *Matlab*. These density estimations in different bandwidths are shown in the Figures 3.4, 3.5, 3.6 and 3.7.

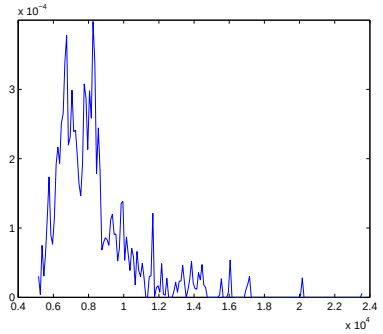

Figure 3.4: Kernel density estimation of car prices by *Matlab* ($h = 100$) with triangle kernel function.

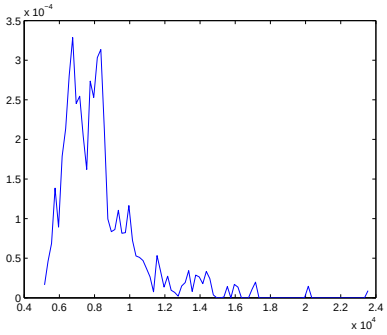

Figure 3.5: Kernel density estimation of car prices by *Matlab* ($h = 200$) with triangle kernel function.

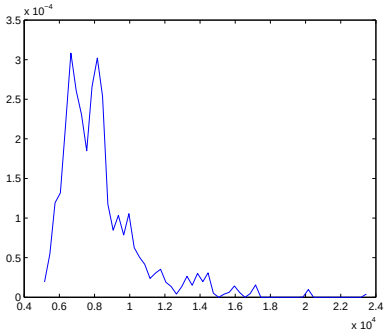

Figure 3.6: Kernel Density Estimation of car prices by *Matlab* ($h = 300$) with Triangle kernel function.

Statistical Properties of Kernel Density Functions

Since the kernel functions are usually probability density functions, they have the following main properties:

$$\int_{-\infty}^{\infty} K(s)ds = 1,$$

$$\int_{-\infty}^{\infty} sK(s)ds = 0,$$

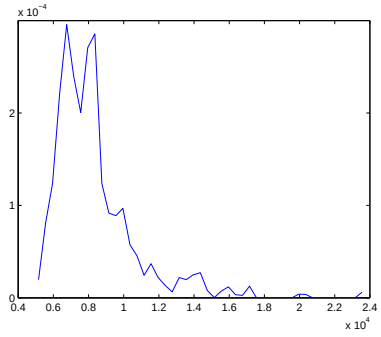

Figure 3.7: Kernel density estimation of car prices by *Matlab* ($h = 400$) with Triangle kernel function.

$$\int_{-\infty}^{\infty} s^2 K(s) ds < \infty,$$

and

$$\int_{-\infty}^{\infty} [K(s)]^2 ds < \infty.$$

For our investigation, we need some further measures of error and deviation:

A. Bias

$$\begin{aligned} \text{Bias} \quad \widehat{f}_h(x) &:= E\widehat{f}_h(x) - f(x) \\ &= \frac{1}{n}\sum_{i=1}^{n} EK_h(x - X_i) - f(x) \\ &= EK_h(x - X_i) - f(x) \\ &= \int \frac{1}{h} K\left(\frac{x-u}{h}\right) f(u) du - f(x). \end{aligned} \tag{3.7.78}$$

Let us put $s := \frac{u-x}{h}$, then, $du = hds$ and, by substitution rule,

$$\text{Bias} \quad \widehat{f}_h(x) = \int K(s) f(u) ds - f(x). \tag{3.7.79}$$

By using the Taylor expansion of $f(u)$ around x, the equation is as follows:

$$\begin{aligned}\text{Bias}\quad \widehat{f_h}(x) &= \int K(s)[f(x)+f'(x)(u-x)+ \\ &\qquad \frac{1}{2}f''(x)(u-x)^2+o(h^2)]ds - f(x) \\ &= f(x)\int K(s)ds + hf'(x)\int sK(s)ds + \frac{h^2}{2}f''(x)\int s^2K(s)ds - f(x) \\ &= \frac{h^2}{2}f''(x)\int s^2K(s)ds + o(h^2).\end{aligned} \tag{3.7.80}$$

Hence as $h \longrightarrow 0$, the bias will be removed. Therefore, we should take h as small as possible to reduce the bias.

B. Variance

$$\begin{aligned}\text{Var}\widehat{f_h}(x) = &\left\{\tfrac{1}{n}\textstyle\sum_{i=1}^{n} K_h(x-X_i)\right\} \\ &= \text{Var}\frac{1}{n^2}\sum_{i=1}^{n} Var(K_h(x-X_i)) \\ &= \frac{1}{n}Var(K_h(x-X_i)) \\ &= \frac{1}{nh}f(x)\int K^2(s)ds + o(\frac{1}{nh}).\end{aligned} \tag{3.7.81}$$

Multivariate Case

Suppose we have a p-dimensional feature random vector $\mathbf{X} = (X_1, X_1..., X_p)$. Each $\mathbf{X}$ have n observations.

Let us represent the i^{th} observation as

$$\mathbf{X}_i = \begin{pmatrix} X_{i1} \\ \vdots \\ X_{ip} \end{pmatrix} \qquad (i = 1, 2, ..., n).$$

Then, the multivariate kernel density estimator of the $\mathbf{X} := (X_1, X_2, ..., X_p)$ will be

$$\widehat{f}_{\mathbf{h}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_1...h_p}\aleph\left(\frac{x_1 - X_{i1}}{h_1}, ..., \frac{x_p - X_{ip}}{h_p}\right). \tag{3.7.82}$$

By using the *multiplicative kernel*

$$\aleph(u) = K(u_1)K(u_2)...K(u_p), \tag{3.7.83}$$

the estimator (3.7.82) becomes

$$\widehat{f}_{\mathbf{h}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{p}h_j^{-1}K\left(\frac{x_j - X_{ij}}{h_j}\right). \tag{3.7.84}$$

The kernel density estimations of the car prices with respect to house prices in the one of the *Matlab* data files were made by writing the code of (3.7.84) in *Matlab*. These density estimations in different bandwidths are shown in the Figures 3.8, 3.9, 3.10 and 3.11.

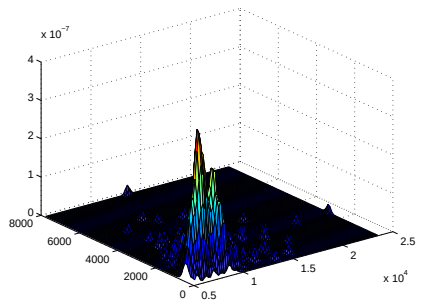

Figure 3.8: Kernel density estimation of car prices and house prices by *Matlab* ($h_1 = 200$, $h_2 = 100$) with Gaussian kernel function.

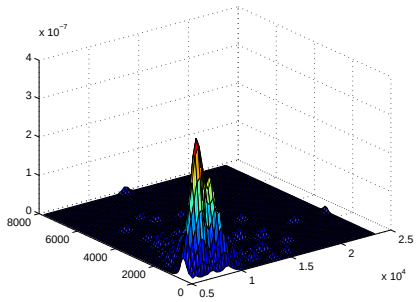

Figure 3.9: Kernel density estimation of car prices and house prices by *Matlab*($h_1 = 300$, $h_2 = 100$) with Gaussian kernel function.

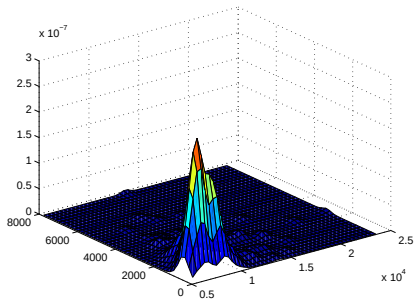

Figure 3.10: Kernel density estimation of car prices and house prices by *Matlab* ($h_1 = 400$, $h_2 = 200$) with Gaussian kernel function.

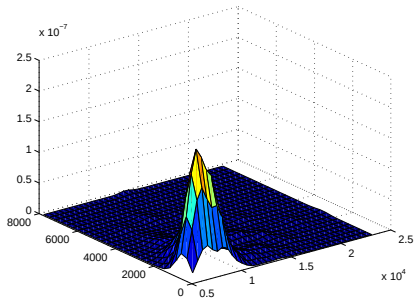

Figure 3.11: Kernel density estimation of car prices and house prices by *Matlab* ($h_1 = 500$, $h_2 = 300$) with Gaussian kernel function.

3.7.2 Generalized Partial Linear Models

Partial linear models are composed of two parts, a linear and non-parametric part. With a known link function G(•), a ***generalized partial linear model*** (**GPLM**) can be represented by

$$E(Y|\mathbf{U}, \mathbf{T}) = G(\mathbf{U}^T\beta + m(\mathbf{T})), \tag{3.7.85}$$

where $\beta = (\beta_1, \beta_2, ..., \beta_p)^T$ is a finite dimensional parameter and $m(\cdot)$ a smooth function.

The estimation of a generalized partial linear model is an two step procedure. Firstly, estimate β with an known $m(\cdot)$, then, find an estimator of $m(\cdot)$ with the help of a known β [Mu00].

To estimate the GPLM by semiparametric maximum likelihood method, the

following assumptions should be made:

$$E(Y|\mathbf{U},\mathbf{T}) = \mu = G(\eta) = G(\mathbf{U}^T\beta + m(\mathbf{T})), \tag{3.7.86}$$

$$Var(Y|\mathbf{U},\mathbf{T}) = \sigma^2 V(\mu). \tag{3.7.87}$$

Let $L(\mu, y)$ be the individual log-likelihood (or let the distribution of Y not be belong to an exponential family). Then, a *quasi-likelihood function* can be written as

$$L(\mu, y) := \frac{1}{\sigma^2}\int_{\mu}^{y} \frac{(s-y)}{V(s)} ds. \tag{3.7.88}$$

Based on the sample, the estimated scale parameter σ is

$$\widehat{\sigma^2} := \frac{1}{n}\sum_{i=1}^{n} \frac{(y_i - \widehat{\mu}_i)^2}{V(\widehat{\mu}_i)}, \tag{3.7.89}$$

where $\widehat{\mu}_i := G(\widehat{\eta}_i)$ and $\widehat{\eta}_i = x_i^T\widehat{\beta} + \widehat{m}(t_i)$.

Profile Likelihood Algorithm

The *profile likelihood method* is one of the methods to solve generalized partial linear models. This method distinguishes two parts in the estimation: a *parametric* and a *nonparametric* part. The method fixes the parameter β to estimate the most probably nonparametric function $m(\cdot)$, then, it uses the estimate $m(\cdot)$ to construct the profile likelihood for β. As a result of the profile likelihood method, the estimator $\widehat{\beta}$ is $\sqrt{n}$-consistent, asymptotically normal and efficient, and the estimator $\widehat{m}(\cdot) = \widehat{m_{\widehat{\beta}}}(\cdot)$ is consistent in sup mode.

Firstly, let $L(\beta)$ be the *parametric profile likelihood function* and defined by

$$L(\beta) := \sum_{i=1}^{n} L(\mu_{i,\beta}, y_i), \tag{3.7.90}$$

where $\mu_{i,\beta} := G(x_i^T\beta + m(t_i))$.

This likelihood is optimized to obtain an estimate for β.

Secondly, the *local likelihood* can be shown to look as follows:

$$L^H(m_\beta(t)) := \sum_{i=1}^{n} \aleph_H(t - t_i) L(\mu_{i,m_{\beta(t)}}, y_i), \tag{3.7.91}$$

where $\mu_{i,m_{\beta(t)}} := G(x_i^T\beta + m(t))$ and $\aleph_H(t - t_i)$ is the kernel weight, with $\aleph_H$ denoting the multidimensional kernel functions and H denoting the *bandwidth matrix*. Moreover, this is also optimized to estimate an estimator for $m_\beta(t)$ at t.

Then, the *individual quasi-likelihood* is

$$\ell_i(\eta) := L(G(\eta), y_i), \tag{3.7.92}$$

and the first and second derivative of it with respect to η are denoted by $\ell_i^{'}$ $\ell_i^{``}$.

For the estimation of the $m(\cdot)$ the maximization of local likelihood can be obtained by solving

$$\sum_{i=1}^{n} \aleph_H(t_i - t_j)\ell_i^{`}(x_i^T\beta + m_j) = 0 \tag{3.7.93}$$

with respect to m_j. To obtain the estimator of β the following derivative of the *quasi-likelihood* part requires to solve [MR99]

$$\sum_{i=1}^{n} \ell_i^{`}(x_i^T\beta + m_i)x_i + m_i^{`} = 0 \tag{3.7.94}$$

with respect to β.

A further differentiation of (3.7.93) results in [MR99]

$$m_j^{'} = -\frac{\sum_{i=1}^{n} \ell_i^{''}(x_i^T\beta + m_j)\aleph_H(t_i - t_j)x_i}{\sum_{i=1}^{n} \ell_i^{''}(x_i^T\beta + m_j)\aleph_H(t_i - t_j)}. \tag{3.7.95}$$

The ***algorithm*** solves in an iterative way and basically includes the following steps [Mu00]:

1. *updating step for* β

$$\beta^{new} = \beta - B^{-1}\sum_{i=1}^{n} \ell_i^{'}(x_i^T\beta + m_i)\widetilde{x_i} \tag{3.7.96}$$

with a Hessian type matrix

$$B = \sum_{i=1}^{n} \ell_i^{''}(x_i^T\beta + m_i)\widetilde{x_i}\widetilde{x_i}^T \tag{3.7.97}$$

and

$$\widetilde{x_j} = x_j + m_j^{'} = x_j - \frac{\sum_{i=1}^{n} \ell_i^{''}(x_i^T\beta + m_j)\aleph_H(t_i - t_j)x_i}{\sum_{i=1}^{n} \ell_i^{''}(x_i^T\beta + m_j)\aleph_H(t_i - t_j)}; \tag{3.7.98}$$

2. *updating step for* m_j

$$m_j^{new} = m_j - \frac{\sum_{i=1}^{n} \ell_i^{'}(x_i^T\beta + m_j)\aleph_H(t_i - t_j)}{\sum_{i=1}^{n} \ell_i^{''}(x_i^T\beta + m_j)\aleph_H(t_i - t_j)}. \tag{3.7.99}$$

3.7.3 Advantages and Disadvantages of Semi-parametric Methods

Advantages:

- The semi-parametric method preserves the degrees of freedom.
- It achieves greater precision than nonparametric models but with weaker assumptions than parametric models.
- By restricting $G(x)$, it reduces the effective dimension of x.
- The risk of the specification error is less than with a parametric model.

Disadvantages:

- The full functional form is not known with confidence.
- The risk of specification error is greater than for the fully nonparametric model.

For a closer information we refer to [Horrowitz].

CHAPTER 4

NONSTATISTICAL METHODS IN CREDIT SCORING

4.1 Neural Networks

4.1.1 Structure of Neural Networks

Like a human brain, ***neural network*** has an ability of learning, remembering and generalizing. The basic element of a neural network is called as a *neuron*. As seen in the Figure 4.1 a neuron has five components: *inputs, weights, combination part, activation part* and *output*. In the following, we give closer information about these components.

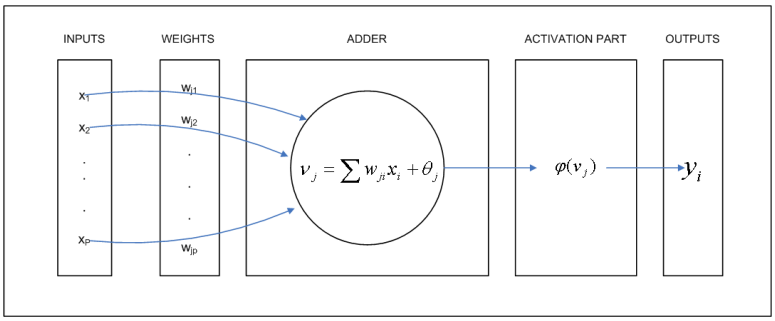

Figure 4.1: Structure of neural networks.

A. Inputs:

Inputs are denoted by x_i. The inputs get the information from the environment of the neuron. This input may be an initial input or an output of prior neurons. For a company, the initial input is $(x_1, x_2, ..., x_p)$.

B. Weights:

The weights for neuron j are comprised in the vector $(w_{j1}, w_{j2}, ..., x_{jp})$. The weights are some constants that determine the effects of inputs on neuron. The greater the weight of an input is, the greater is the impact of the input on the neuron.

C. Adder:

In this part, the input values are multiplied with weights and summed with the threshold level θ_j and, then, sent to the activation part.

D. Activation Part:

Activation function of a neuron specified the final output of a neuron at some activity level and denoted by $\varphi(\cdot)$. There are three main types of activity functions:

D.1. Threshold Function: The *threshold function* for neuron k is simply

$$\varphi(v_k) := y_k = \begin{cases} 0 & \texttt{if} \quad v_k \geq 0 \\ 1 & \texttt{if} \quad v_k < 0, \end{cases} \tag{4.1.1}$$

where $v_k = \sum_j w_{kj} x_j - \theta_k$.

The threshold function is shown in Figure 4.2.

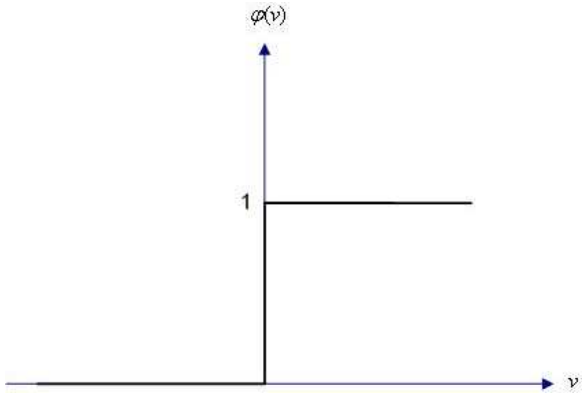

Figure 4.2: Threshold activation function [H94].

D.2. Piecewise-Linear Function: The *piecewise-linear function* is defined by

$$\varphi(v) := \begin{cases} 1, & \text{if} \quad v \geq 1/2 \\ v, & \text{if} \quad 1/2 > v > -1/2 \\ 0, & \text{if} \quad v \leq -1/2. \end{cases} \tag{4.1.2}$$

Figure 4.3 shows this piecewise-linear function.

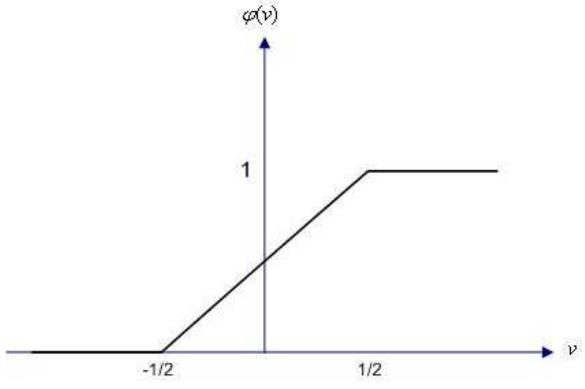

Figure 4.3: Piecewise-linear activation function [H94].

3. Sigmoid Function: The *sigmoid function* is the most widely used activation function in the applications of neural networks. All strictly increasing functions with smoothness and asymptotic properties can be put into this type. As an example of a sigmoid function, we can give the logistic function. The logistic function has the following form:

$$\varphi(v) = \frac{1}{1 + \exp(-\alpha v)}, \tag{4.1.3}$$

where α is the slope parameter.

Another example can be the hyperbolic tangent function given by

$$\varphi(v) = \tanh\left(\frac{v}{2}\right) \ \frac{1 - \exp(-v)}{1 + \exp(-v)}. \tag{4.1.4}$$

In Figure 4.4, the examples of sigmoid functions can be seen.

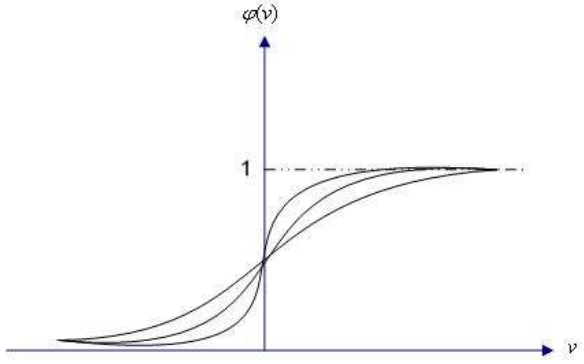

Figure 4.4: Sigmoid activation functions.

E. Output:

In this step, the *output* of the activation part diffused to the outer world or another neuron. Every neuron has only one output.

4.1.2 Learning Process

Introduction

The significance of neural networks comes from its ability to *learn* their *environment* and according to it, to improve its performance by means of *adjusting weights and thresholds.*

Let us consider the following network given in Figure (4.5):

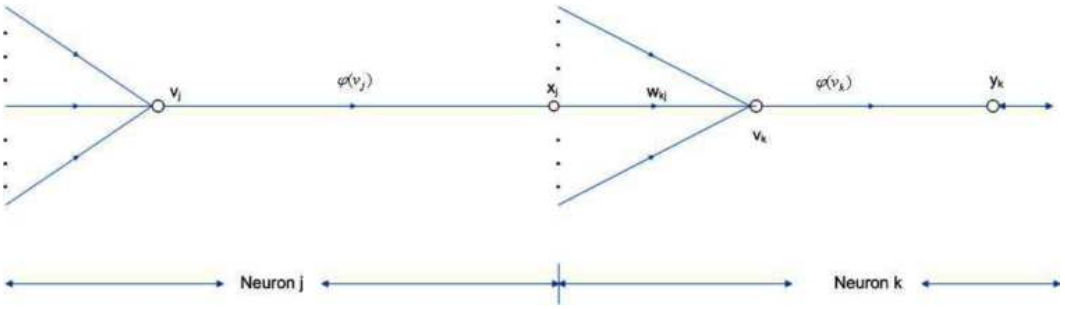

Figure 4.5: A neural network structure.

In this network, the x_j denote the output of neuron j and they are connected with the internal activity v_k of neuron k. Let $w_{kj}(t)$ represent the value of weight w_{kj} at time t. Then, to obtain the updated weights for time $t+1$, the adjustment $\Delta w_{kj}(t)$ is applied in the following way:

$$w_{kj}(t+1) := w_{kj}(t) + \Delta w_{kj}(t). \tag{4.1.5}$$

Some basic rules of the learning processes are shown in Figure 4.6.

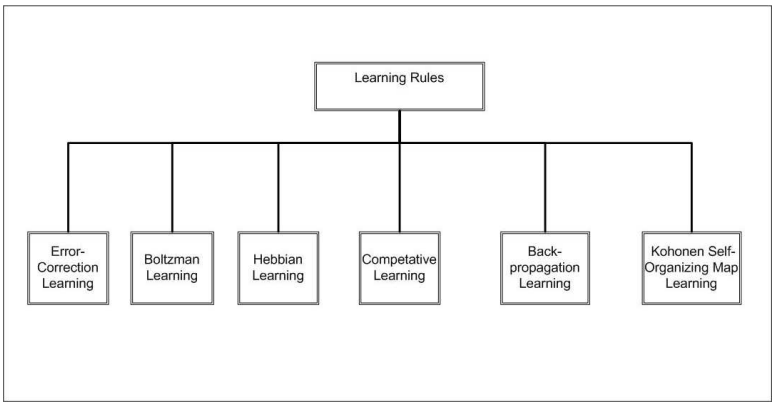

Figure 4.6: Diagram of learning process [H94].

Error Correction Learning

In Figure 4.7, d_j denotes the desired response or target output for neuron j. Furthermore, y_j denotes the output of neuron i. In error correction learning, the algorithm tries to abate the error between desired response and actual output. The error is simply

$$e_j = d_j - y_j, \tag{4.1.6}$$

where

$$y_j := \sum_{i=1}^{p} w_{ji} x_{ji}. \tag{4.1.7}$$

As a performance measure, the sum of mean squared error is used, i.e.,

$$J := \frac{1}{2} E(e_j). \tag{4.1.8}$$

Here, the factor 1/2 is included for convenience. Our problem is to find the best set of weights $(w_{j1}, w_{j2}, ..., w_{jp})$ which minimize (4.1.8).

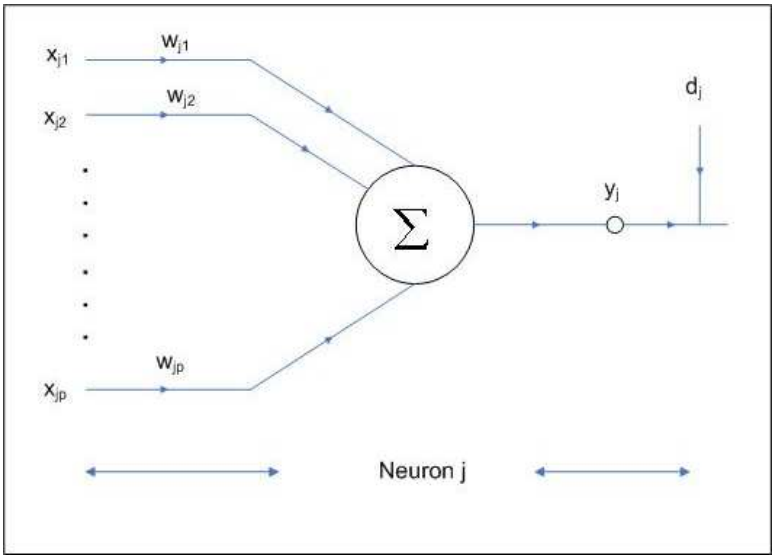

Figure 4.7: Error correction learning.

The solution of this problem is known to be represented as *Wiener-Hopf equations*. By substituting (4.1.6) and (4.1.7) into (4.1.8), we get

$$J = \frac{1}{2}E(d_j^2) - E\left(\sum_{i=1}^{p} w_{ji}x_{ji}d_j\right) + \frac{1}{2}E\left(\sum_{k=1}^{p}\sum_{i=1}^{p} w_{jk}w_{ji}x_{jk}x_{jk}\right) \quad (4.1.9)$$

$$= \frac{1}{2}E(d_j^2) - \sum_{i=1}^{p} w_{ji}E(x_{ji}d_j) + \frac{1}{2}\sum_{k=1}^{p}\sum_{i=1}^{p} w_{jk}w_{ji}E(x_{ji}x_{jk}). \quad (4.1.10)$$

Then, the gradient Δw_{kj} can be found as

$$\Delta w_{ji} = \frac{\partial J}{\partial w_{ij}} \quad (4.1.11)$$

$$= -E(x_{ji}d_j) + \sum_{k=1}^{p} w_{jk}E(x_{ji}x_{jk}) \qquad (i = 1, 2, ..., p). \quad (4.1.12)$$

Therefore, the optimality condition is defined by

$$\Delta w_{ji} = 0 \qquad (i = 1, 2, ..., p). \quad (4.1.13)$$

Boltzman Learning

This type of learning is a *stochastic* one. The *Boltzman machine* has basically following properties [H94]:

- processing units have binary values (-1 and 1),
- all the connections between units are symmetric,
- the units are picked at random and one at a time or updating,
- it has no self-feedback,
- it permits the use of hidden nodes,
- it uses stochastic neurons with a probabilistic firing mechanism,
- it may also be trained by supervision of a probabilistic form .

The *Boltzman machine* is described by the following *energy function*:

$$E = -\frac{1}{2}\sum_{i}\sum_{j \atop i \neq j} w_{ji} s_j s_i, \tag{4.1.14}$$

where s_i is the state of neuron i, and w_{ji} is the weight connecting neuron i to neuron j. The relation $i \neq j$ implies that none of the neurons in the machine has self-feedback. The machine operates by choosing a neuron at random - say, neuron j - at some step of learning process, and flipping the state of neuron j from state s_j to state $-s_j$ at some constant $C > 0$ with probability

$$W(s_j \longrightarrow -s_j) = \frac{1}{1 + \exp(-\Delta E_j / C)}, \tag{4.1.15}$$

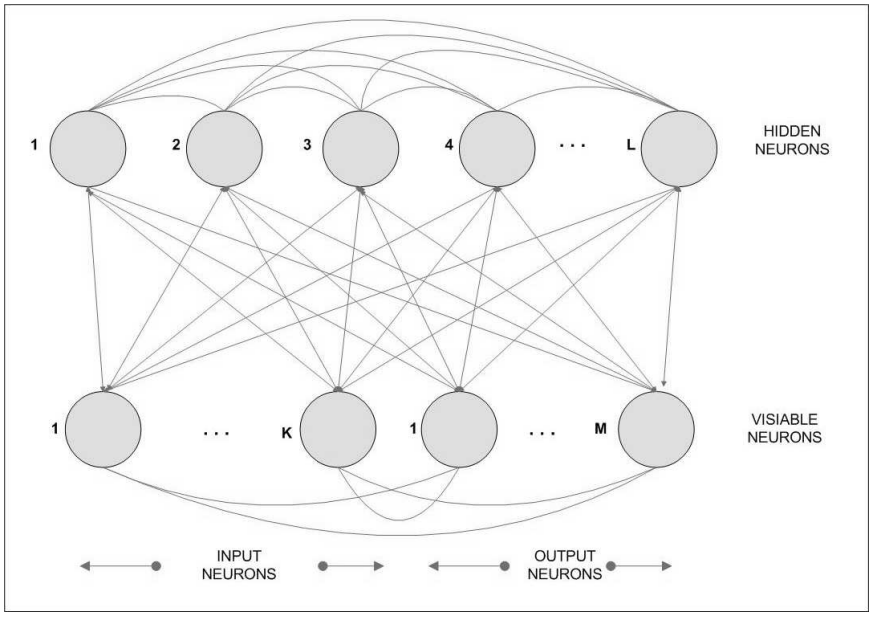

Figure 4.8: Boltzman machine [H94].

where ΔE_j is the change in the energy function of the machine.

The Boltzman machine has two types of stochastic neurons as seen in (4.8): *hidden neurons, visible neurons*, where the visible neurons work as a connection between the network and environment and on the other hand, hidden neurons work as constrains of the input vectors by taking the higher order correlations of the vectors into account. The machine has two types of operations:

- *clamped condition*: in this type of operation, the visible neurons are all clamped onto specific states determined by the environment;
- *free-running condition*: here, all types of neurons are permitted to operate freely.

Let the states of visible neurons be α and that of hidden neurons be β. We assume the network has L hidden neurons and K visible neurons, so α runs from

1 to 2^K and β runs from 1 to 2^L. Let P_α^+ indicate the probability that the visible neurons are collectively in state α, given that the network is operating in its clamped condition. Let P_α^- denote the probability that these same neurons are collectively in state α, given that the network is allowed to run freely, but no environment input. The signs $+$, $-$ indicate that the network is in clamped condition and running freely, respectively. Accordingly, the set of properties

$$\{P_\alpha^+ | \alpha = 1, 2, ..., 2^K\} \tag{4.1.16}$$

consists of the desired probabilities which represent the environment, and the set of properties

$$\{P_\alpha^- | \alpha = 1, 2, ..., 2^K\} \tag{4.1.17}$$

consists of the actual probabilities which are computed by the network.

Suppose ρ_{ji}^+ denotes the correlation between the states of neurons i and j, conditional on the network being in its clamped condition. Let ρ_{ji}^- denote the unconditional correlation between the states of neurons i and j. The correlations ρ_{ji}^+ and ρ_{ji}^- are given by

$$\rho_{ji}^+ := \sum_\alpha \sum_\beta P_{\alpha\beta}^+ s_{j|\alpha\beta} s_{i|\alpha\beta}, \tag{4.1.18}$$

$$\rho_{ji}^- := \sum_\alpha \sum_\beta P_{\alpha\beta}^- s_{j|\alpha\beta} s_{i|\alpha\beta}, \tag{4.1.19}$$

where $s_{i|\alpha\beta}$ denotes the state of neuron i, given that the visible neurons of the machine are in state α and the hidden neurons are in state β. Then, according to the Boltzman learning rule, the change Δw_{ji} applied to the weight w_{ji} from

neuron i to neuron j is defined by

$$\Delta w_{ji} := \eta(\rho_{ji}^{+} - \rho_{ji}^{-}), \qquad \forall i \neq j \tag{4.1.20}$$

where η is a *learning-rate parameter*.

Hebbian Learning

Hebb's rule of learning is the most well-known learning algorithm. Hebb proposed that weights are adjusted in proportional to the correlation between input and output of the network, respectively [HS95].

Consider the Figure 4.5 again. The weights, inputs and outputs of neuron k are denoted by w_{kj}, x_j, y_k, respectively. The Hebb learning has the following form of adjustment

$$\Delta w_{kj} := F(y_k(t), x_j(t)), \tag{4.1.21}$$

where $F(\cdot, \cdot)$ is a function of $y_k(t)$, and $x_j(t)$. A special case of equation (4.1.21) is:

$$\begin{aligned} \Delta w_{kj} &= \eta \mathrm{Cov}[y_k(t), x_j(t)] && (4.1.22) \\ &= \eta E[(y_k(t) - \overline{y}_k)(x_j(t) - \overline{x}_j)], && (4.1.23) \end{aligned}$$

where η is the *rate of learning*.

Competitive Learning

In *competitive learning*, output neurons which win the competition are activated. Competitive learning has three basic properties:

- A set of neurons that are all the same except for some randomly distributed weights, and which therefore respond differently to a given set of input patterns.

- A limit imposed on the "strength" of each neuron.

- A mechanism which permits the neurons to compete for the right to respond to a given subset of inputs, such that only one output neuron, or only one neuron per group, is active at a time. The neuron which wins the competition is called a *winner-takes-all neuron* [H94].

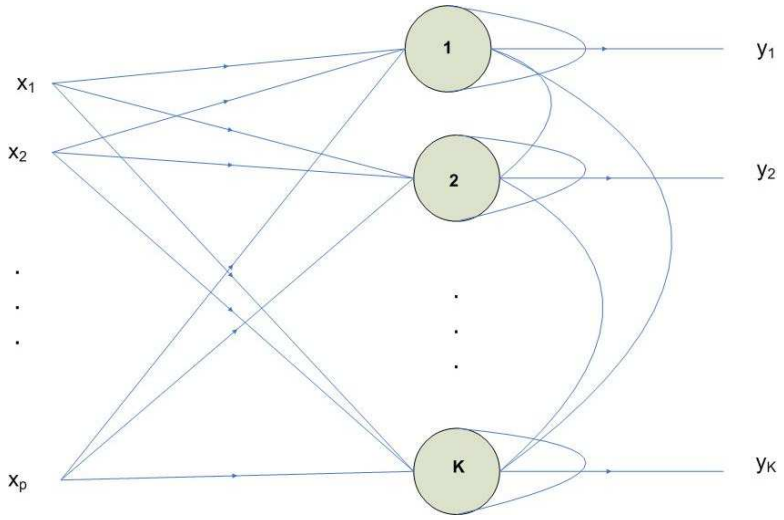

Figure 4.9: Single layer competitive network [H94].

Let neuron j be a winning neuron with the largest activity level v_j. Then, the output y_j of the winning neuron is set equal to one and the output of other neurons are set to zero. Furthermore, in competitive learning, a fixed amount of

weights are assigned to each neuron such that

$$\sum_i w_{ji} = 1 \qquad \text{for all } j. \tag{4.1.24}$$

The algorithm of competitive learning adjusts the weights by the way of

$$\Delta w_{ji} = \begin{cases} \eta(x_i - w_{ji}) & \text{if neuron j wins the competition} \\ 0 & \text{if neuron j loses the competition,} \end{cases} \tag{4.1.25}$$

where η is the learning-rate parameter.

Back Propagation Algorithm

The *back propagation algorithm* is a learning rule for *multi-layered* neural networks. The algorithm primarily work for adjusting the synaptic weights in order to minimize the network system's output and actual output.

Derivation of Back Propagation Algorithm

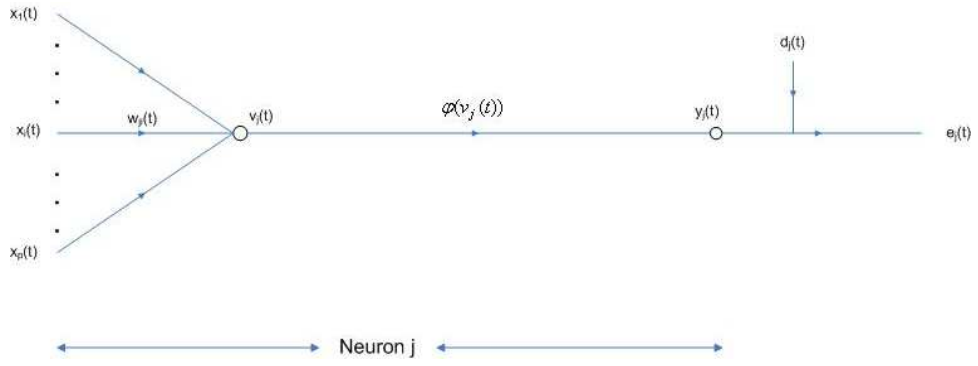

Figure 4.10: The feed-forward network [H94].

The error at the output of neuron j at iteration t can be defined by

$$e_j(t) := d_j(t) - y_j(t). \tag{4.1.26}$$

As the sum of squared errors of the network at time t we put

$$\xi(t) := \frac{1}{2}\sum_{j\in C} e_j^2(t) \tag{4.1.27}$$

Suppose that the iteration is finished at the T. We try to minimize the average sum of squared errors of the training set. In other words, the cost function of the training set learning performance, that is represented as follows and is to be *minimized*:

$$\min \xi_{av} = \frac{1}{T}\sum_{t=1}^{T} \xi(t). \tag{4.1.28}$$

In the minimization part, the back propagation algorithm uses the *least mean square algorithm (LMS)*. Let us define

$$v_j(t) := \sum_{i=0}^{p} w_{ji}(t)x_i(t), \tag{4.1.29}$$

where p is the total number inputs applied to neuron j. Furthermore, the output $y_j(t)$ at iteration t is

$$y_j(t) := \varphi_j(v_j(t)). \tag{4.1.30}$$

The back propagation algorithm improves its weights as proportional to the instantaneous gradient $\frac{\partial \xi(t)}{\partial w_{ji}(t)}$.

In the algorithm, the gradient $\frac{\partial \xi(t)}{\partial w_{ji}(t)}$ refers to a *sensitivity factor* which determines the direction of the search for weight w_{ji}. According to chain rule, the

gradient may be expressed in the following way:

$$\frac{\partial \xi(t)}{\partial w_{ji}(t)} = \frac{\partial \xi(t)}{\partial e_j(t)} \frac{\partial e_j(t)}{\partial y_j(t)} \frac{\partial y_j(t)}{\partial v_j(t)} \frac{\partial v_j(t)}{\partial w_{ji}(t)}. \tag{4.1.31}$$

Let us compute the gradient. Firstly, we differentiate both sides of (4.1.27) with respect to $e_j(t)$, then,

$$\frac{\partial \xi(t)}{\partial e_j(t)} = e_j(t). \tag{4.1.32}$$

Differentiating both sides of (4.1.26) with respect to $y_j(t)$, then,

$$\frac{\partial e_j(t)}{\partial y_j(t)} = -1. \tag{4.1.33}$$

Next, differentiating (4.1.30) with respect to $v_j(t)$ gives

$$\frac{\partial y_j(t)}{\partial v_j(t)} = \varphi_j'(v_j(t)). \tag{4.1.34}$$

Finally, differentiating (4.1.29) with respect to $w_{ji}(t)$ yields

$$\frac{\partial v_j(t)}{\partial w_{ji}(t)} = x_i(t). \tag{4.1.35}$$

Furthermore, by putting (4.1.32) to (4.1.35) into (4.1.31), we obtain

$$\frac{\partial \xi(t)}{\partial w_{ji}(t)} = -e_j(t)\varphi_j'(v_j(t))x_i(t). \tag{4.1.36}$$

Then, the correction $\Delta w_{ji}(t)$ applied to $w_{ji}(t)$ means an improvement process

which can be defined by the delta-rule

$$\Delta w_{ji}(t) := -\mu \ \frac{\partial \xi(t)}{\partial w_{ji}(t)}, \tag{4.1.37}$$

where μ is a constant learning rate. Use of (4.1.36) in (4.1.37) yields

$$\Delta w_{ji}(t) := -\mu \delta_j(t)(v_j(t))x_i(t), \tag{4.1.38}$$

where the *local gradient* $\delta_j(t)$ is defined by

$$\begin{aligned} \delta_j(t) &= -\frac{\partial \xi(t)}{\partial e_j(t)} \ \frac{\partial e_j(t)}{\partial y_j(t)} \ \frac{\partial y_j(t)}{\partial v_j(t)} \\ &= e_j(t)\varphi_j'(v_j(t)). \end{aligned} \tag{4.1.39}$$

In this step, we are faced with two situation. The first one is given by neuron j being an output node. The second one is the neuron j being an hidden node.

CASE I. **Neuron j is an output neuron:**

If neuron j is lying on the output layer of the network, there would exist a desired response. Furthermore, we may compute the error sum of squares by the formula (4.1.26) and correct synaptic weights with the help of (4.1.38).

CASE II. **Neuron j is a hidden neuron:**

If neuron j is lying on the hidden layer of the network, there is no desired response. So, the error rate may be determined in terms of the error rates of all neurons connected directly with that hidden neuron. Let us consider the situation in the below Figure (4.11):

In this figure, the j^{th} neuron represents the hidden node and the k^{th} neuron

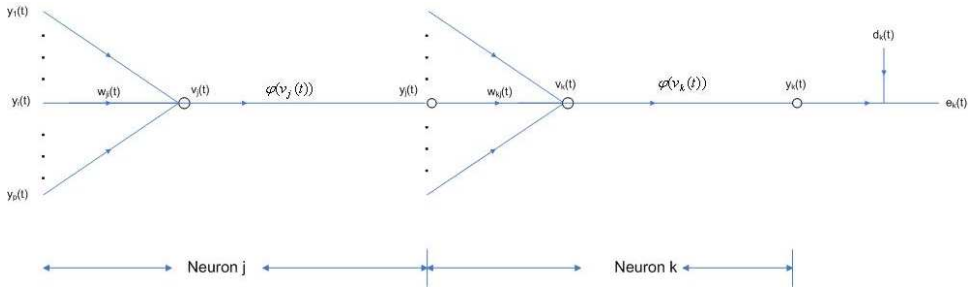

Figure 4.11: Two layer feed-forward network [H94].

represents output neuron, where

$$v_j(t) := \sum_i w_{ji}(t)x_i(t), \tag{4.1.40}$$

$$v_k(t) := \sum_j w_{kj}(t)y_j(t) \tag{4.1.41}$$

and

$$y_j(t) := \varphi_j(v_j(t)), \tag{4.1.42}$$

$$y_k(t) := \varphi_k(v_k(t)). \tag{4.1.43}$$

Then, the local gradient $\delta_j(t)$ for the hidden neuron j can be defined as

$$\begin{aligned} \delta_j(t) &:= -\frac{\partial \xi(t)}{\partial y_j(t)} \frac{\partial y_j(t)}{\partial v_j(t)} \\ &= -\frac{\partial \xi(t)}{\partial y_j(t)} \varphi_j'(v_j(t)). \end{aligned} \tag{4.1.44}$$

In order to obtain this correction factor for the hidden neuron, we must calculate $\frac{\partial \xi(t)}{\partial y_j(t)}$ by the following procedure:

For our case, the instantaneous sum of squared error existing in the output

neuron k can be written as follows:

$$\xi(t) = \frac{1}{2} \sum_{k \in C} e_k^2(t). \tag{4.1.45}$$

Firstly, we differentiate (4.1.45) with respect to the $y_j(t)$:

$$\frac{\partial \xi(t)}{\partial y_j(t)} = \sum_k e_k \ \frac{\partial e_k(t)}{\partial y_j(t)}. \tag{4.1.46}$$

Then, by chain rule, (4.1.46) becomes

$$\frac{\partial \xi(t)}{\partial y_j(t)} = \sum_k e_k \ \frac{\partial e_k(t)}{\partial v_k(t)} \ \frac{\partial v_k(t)}{\partial y_j(t)}. \tag{4.1.47}$$

Now, from Figure 4.11, we know that

$$e_k(t) = d_k(t) - y_k(t) = d_k(t) - \varphi_k(v_k(t)), \tag{4.1.48}$$

$$v_k(t) = \sum_j w_{kj}(t) y_j(t). \tag{4.1.49}$$

Therefore,

$$\frac{\partial e_k(t)}{\partial v_k(t)} = -\varphi_k(v_k(t)), \tag{4.1.50}$$

$$\frac{\partial v_k(t)}{\partial y_j(t)} = w_{kj}(t) \tag{4.1.51}$$

and, then,

$$\frac{\partial \xi(t)}{\partial y_j(t)} = -\sum_k e_k \ \varphi_k(v_k(t)) \ w_{kj}(t). \tag{4.1.52}$$

Accordingly, by using (4.1.52) in the equation (4.1.44), we obtain

$$\delta_j(t) = \sum_k e_k \;\; \varphi_k(v_k(t)) \;\; w_{kj}(t) \;\; \varphi_j'(v_j(t)). \tag{4.1.53}$$

From (4.1.31), we learn here:

$$\frac{\partial \xi(t)}{\partial w_{ji}(t)} = \delta_j(t) \;\; \frac{\partial v_j(t)}{\partial w_{ji}(t)}. \tag{4.1.54}$$

Putting

$$v_j(t) = \sum_i w_{ji}(t) x_i(t) \tag{4.1.55}$$

and (4.1.53) into the equation (4.1.54) yields

$$\frac{\partial \xi(t)}{\partial w_{ji}(t)} = x_i(t) \;\; \varphi_j'(v_j(t)) \;\; \sum_k e_k \;\; \varphi_k'(v_k(t)) w_{kj}(t). \tag{4.1.56}$$

Then, the correction (4.1.37) for the hidden neuron is obtained in the following form by using (4.1.54) in it:

$$\begin{aligned} \Delta w_{ji}(t) &= \mu \;\; \frac{\partial \xi(t)}{\partial w_{ji}(t)} \\ &= \mu \;\; \delta_j(t) \;\; x_i(t). \end{aligned} \tag{4.1.57}$$

4.1.3 Advantages and Disadvantages of Neural Networks

Advantages:

The neural network

- does not use pre-programmed knowledge base,

- suited to analyze complex pattern,
- have no restrictive assumptions,
- allows for qualitative data,
- can handle noisy data,
- can overcome autocorrelation,
- user-friendly: clear output, and
- robust and flexible.

Disadvantages:

The neural network

- requires high quality data,
- variables must be carefully selected a priori,
- risk of overfitting,
- requires a definition of architecture,
- long processing time,
- possibility of illogical network behavior, and
- large training sample required.

For a closer explanation we refer to the [BO04].

CHAPTER 5

Parameter Estimation Accuracy of Logistic Regression

5.1 Introduction and Methodology

Since the 1980's, the logistic regression is the primary tool for research. It is used for every type and size of data without any consideration on the accuracy of parameter estimation. Therefore, in this section, we tried to check the conditions in which logistic regression performs well. In our analysis, we made our analysis by ***monte carlo*** type simulation in ***Matlab***.

As we mention in previous sections, the logistic regression is defined as follows:

$$p(\mathbf{x}_i) = G(\mathbf{x}_i, \mathbf{w}) = \frac{e^{w_0+w_1x_{i1}+w_2x_{i2}+\ldots+w_px_{ip}}}{1+e^{w_0+w_1x_{i1}+w_2x_{i2}+\ldots+w_px_{ip}}} = \frac{e^{\mathbf{x_i w}}}{1+e^{\mathbf{x_i w}}}, \tag{5.1.1}$$

where w_i's $(i = 1, 2, \ldots, p)$ are weights, and x_{ji} is the i^{th} independent variable for the individual j.

For *Monte Carlo* type simulation, we follows *a five step procedure* that is:

Step 1 The set of independent variables, $\mathbf{x}$ is derived from various sets of dis-

tributions with three different dimensions.

Step 2 The variables are normalized in order to prevent from scale problems.

Step 3 The variables multiplied by some initial weights w_i's $i = 1, 2, \ldots, p$, and summed.

Step 4 According to following formulae y scores are obtained,

$$y^* = \mathbf{x}\mathbf{w} + \varepsilon, \tag{5.1.2}$$

where $\varepsilon \sim logistic(0, \frac{\pi}{\sqrt{3}})$ and we assign y as $\begin{cases} 0, & \text{if } y^* < 0 \\ 1, & \text{if } y^* \geq 0. \end{cases}$.

Step 5 Lastly, by using produced y and derived $\mathbf{x}$, weights are estimated from logistic distribution and compared with initials.

In *Matlab*, there is no tool for logistic random variable to produce error terms. Therefore, we derive it by making an inverse operation in the following way:

If z follows a logistic distribution with parameters α and, β, then,

$$F(z, \alpha, \beta) = \frac{\exp\left((z-\alpha)/\beta\right)}{1 + \exp\left((z-\alpha)/\beta\right)}. \tag{5.1.3}$$

This expression is equivalent to

$$F(z, \alpha, \beta) = 1 - \frac{1}{1 + \exp\left((z-\alpha)/\beta\right)}. \tag{5.1.4}$$

Bringing 1 to the other sides results in

$$1 - F(z, \alpha, \beta) = \frac{1}{1 + \exp\left((z-\alpha)/\beta\right)}. \tag{5.1.5}$$

Since the exponential term is always positive, we can take the -1^{st} power of both sides:

$$\frac{1}{1-F(z,\alpha,\beta)}-1 = \exp\left((z-\alpha)/\beta\right) \tag{5.1.6}$$

$$\frac{F(z,\alpha,\beta)}{1-F(z,\alpha,\beta)} = \exp\left((z-\alpha)/\beta\right). \tag{5.1.7}$$

Then, by taking logarithms of both sides, we obtain

$$\ln\left(\frac{F(z,\alpha,\beta)}{1-F(z,\alpha,\beta)}\right)=\frac{z-\alpha}{\beta}. \tag{5.1.8}$$

So,

$$z=\alpha+\beta\ln\left(\frac{F(z,\alpha,\beta)}{1-F(z,\alpha,\beta)}\right). \tag{5.1.9}$$

In order z to be a logistic random variable and since $F(\cdot)$ only takes values between 0 and 1, we change it with *uniform random variable*.Then, we obtain the following formula:

$$z=\alpha+\beta\ln\left(\frac{Unif(0,1)}{1-Unif(0,1)}\right). \tag{5.1.10}$$

In particular, we derive our error term, that is $\varepsilon \sim logistic(0, \frac{\pi}{\sqrt{3}})$, by

$$\varepsilon=\frac{\pi}{\sqrt{3}}\ln\left(\frac{Unif(0,1)}{1-Unif(0,1)}\right). \tag{5.1.11}$$

In our estimations, to check the estimation accuracy of logistic regression in different dimensions and sizes, we made our estimations by using *one, six*, and *twelve* independent variables for number of *250, 500* and 1000 *data* cases with *1000 simulations*.

To check the parameter accuracy in logistic regression, we took coefficient of

variation as a measure of bias. The coefficient of variation is

$$CV = \frac{\text{Standard deviation of } \widehat{w}}{\text{Mean of } \widehat{w}}. \tag{5.1.12}$$

Moreover, we have tested it by using different set of weights. The testing weight sets are selected among sequences with a sum of one or smaller (preserving the mean of the normalized data). For the 6 and 12 variables cases, we selected the weights in the following manner: the first ones having high variations among them, the second ones with relatively lower values and the third ones having relatively higher effects.

5.2 Results

5.2.1 One Variable Case

In this section, to examine the prediction accuracy of logistic regression in small dimensions of data, we used only a single variable that is generated from the uniform distribution with parameters 1 and 100. Furthermore, we made our calculations under the two different weights. One is a smaller and the other is the higher loading.

Smaller Initial Weight $w = 0.2$

Firstly, at the second step of the algorithm, we selected the initial weight as $w = 0.2$. Figure 5.1 shows the coefficient variation of the estimator. Contrary to what we expected, this graph figures out the direct relation between the percentage default in the data set and the bias. Herewith, according to the figure, if the data set includes more than 30% defaults in it, the bias will increase sharply.

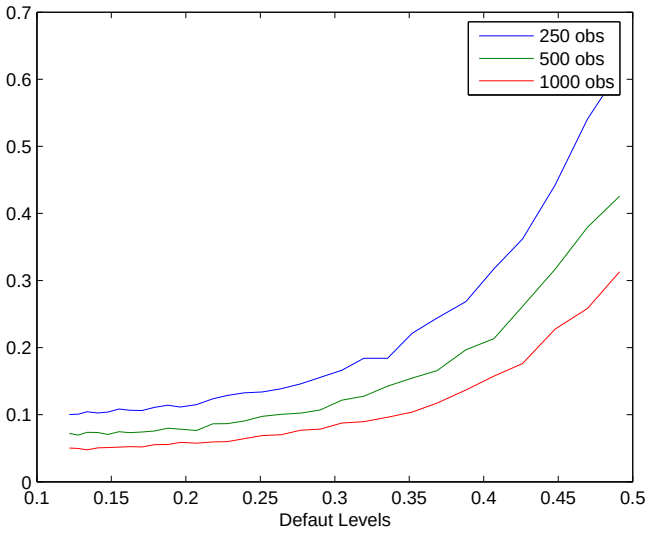

Figure 5.1: Coefficient of variation of estimator in different default levels for $w = 0.2$.

The coefficient estimate and the initial value $w = 0.2$ are compared in the Figure 5.2. Accordingly, a nearly positive linear relationship between the default level and the accuracy of the estimate can be observed from the graph. Since the bias rises after the 30% default cut-off, it can be reasonable to include 30% default in the data sets. Furthermore, the smaller bias and error in the estimation are observed with the long data sets.

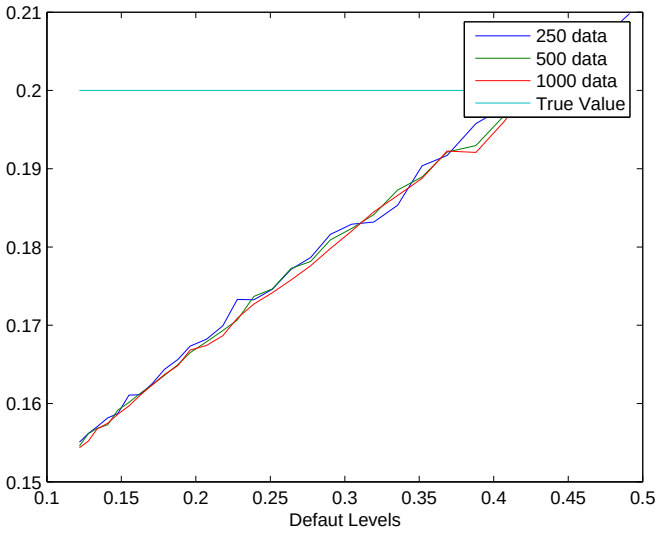

Figure 5.2: Coefficient estimate and their true value in different default levels for $w = 0.2$.

Higher Initial Weight $w = 0.6$

To see the effect of the variables with higher loadings, we selected initial weight as $w = 0.6$. Figures 5.3 and 5.4 are the graphs of coefficient of variation and coefficient estimations, respectively. Similar result with the above case $w = 0.2$ can be concluded. The first figure shows an increase in the bias before the 5% and 30% default cases in the data. From the other figure, positive relation can be observed between the estimation accuracy and the default level. Therefore, in fact, we noted for both cases of weights similar results.

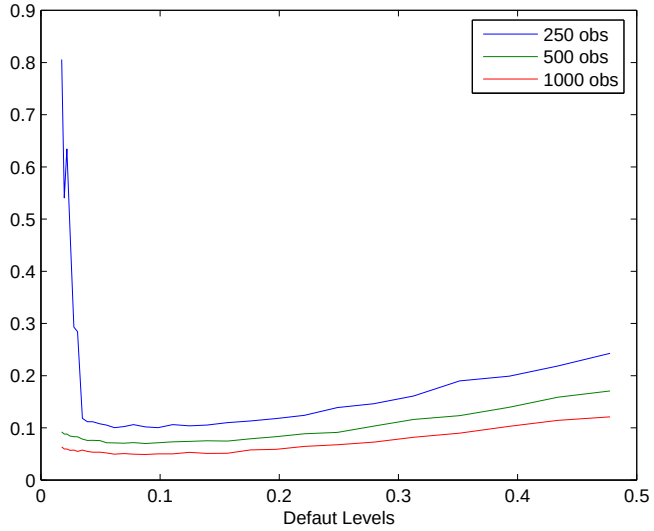

Figure 5.3: Coefficient of variation of estimator in different default levels for $w = 0.6$.

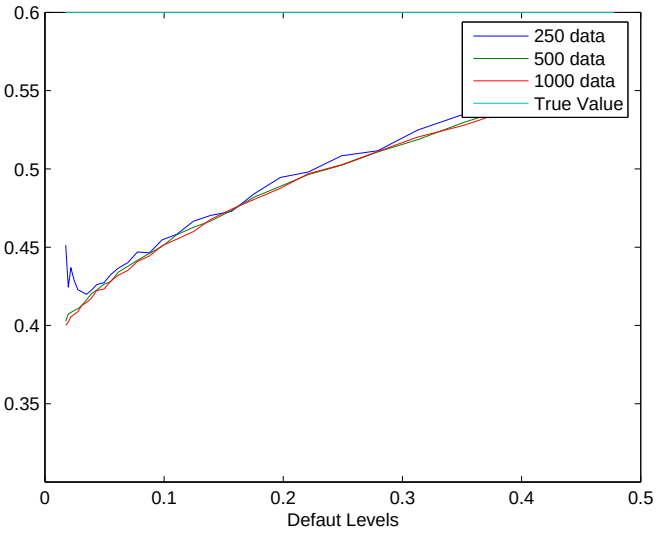

Figure 5.4: Coefficient estimate and their true value in different default levels for $w = 0.6$.

5.2.2 Six Variables Case

The data were generated from a set of random variables from the distributions:

1. $x_1 \sim uniform(1,100)$,
2. $x_2 \sim exponential(16)$,
3. $x_3 \sim normal(12,4)$,
4. $x_4 \sim weilbull(0.5,2)$,
5. $x_5 \sim chi(17)$,
6. $x_6 \sim beta(4,3)$.

First Set of Weights

Our first set of initial weights taken are $w_1 = 0.8$, $w_2 = -0.9$, $w_3 = 0.4$, $w_4 = 0.05$, $w_5 = 0.75$ and $w_6 = -0.1$.

This set includes high and low weights at the same time. The average coefficient of variations are shown in Figure 5.5. Accordingly, it is observed that when the data size is getting higher, the Monte Carlo simulations shows low bias in the estimation of parameters.

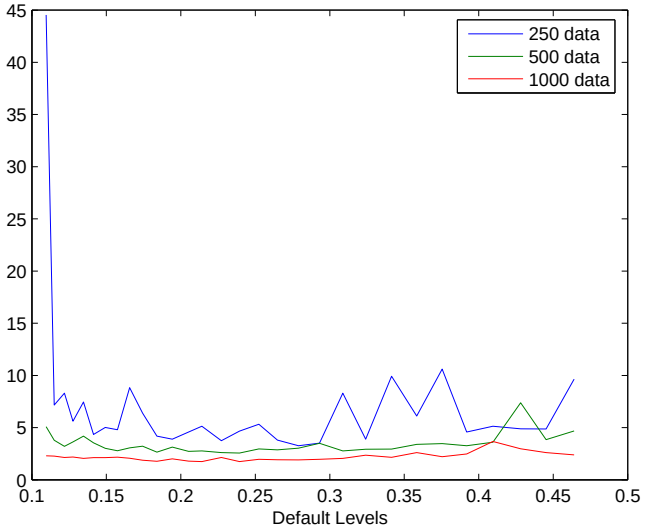

Figure 5.5: *Six variable case:* Average coefficient of variation of estimators in different default levels for the first set of weights.

In Figure 5.6, the true values of parameters and estimates of them for different sizes of data set are shown for each variable in an order from left to right. The first, second and fifth ones are drawn for high in the absolute value weights and it can be seen that the greater the number of defaults are in the data set, the closer are the estimates to the true values. Furthermore, the others are the smaller weights' graphs, and these represent nearly perfect estimations on the 30% default level.

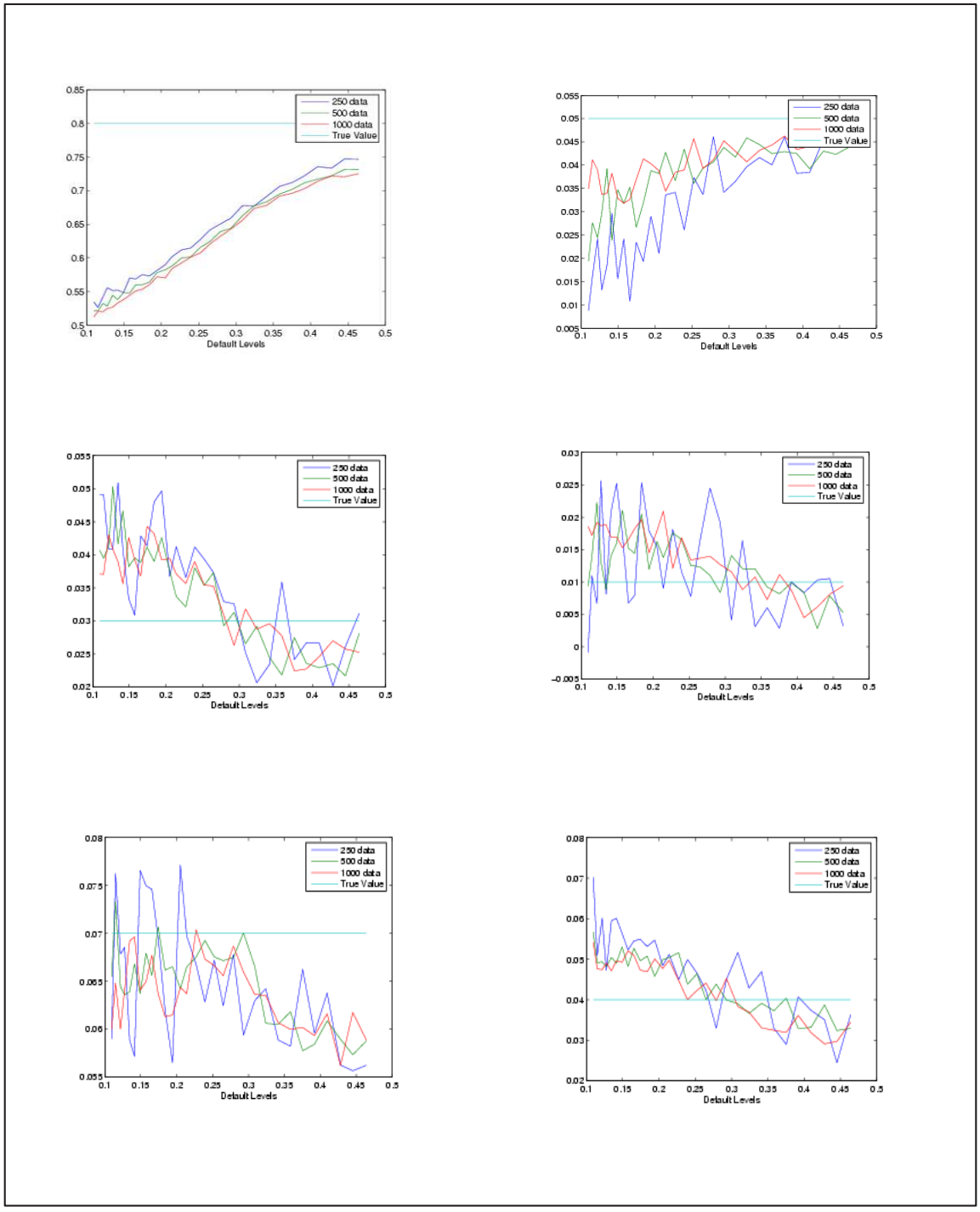

Figure 5.6: *Six variable case:* Coefficients' estimates and their true values on different default levels for the first set of weights.

Second Set of Weights

Our second set consists of $w_1 = 0.02$, $w_2 = 0.05$, $w_3 = 0.03$, $w_4 = 0.01$, $w_5 = 0.07$ and $w_6 = 0.04$.

This set assigns low weights to the variables. The coefficient of variation estimates of the data sets in different default levels shown in Figure 5.7 indicates that in the small samples, bias of estimators are much higher if the small number of defaults occurred in the sample. Moreover, the bias is getting smaller for each sample sizes after the point of 25% default level.

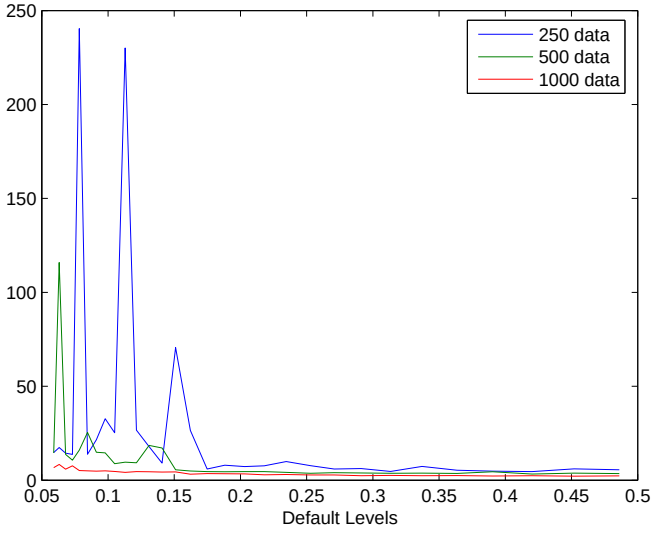

Figure 5.7: *Six variable case:* Average coefficient of variation of estimators in different default levels for the second set of weights.

Furthermore, Figure 5.8 shows that when the weights are very low the parameter estimation accuracy of logistic regression is very perfect, especially, in the sample with nearly 30% default in it.

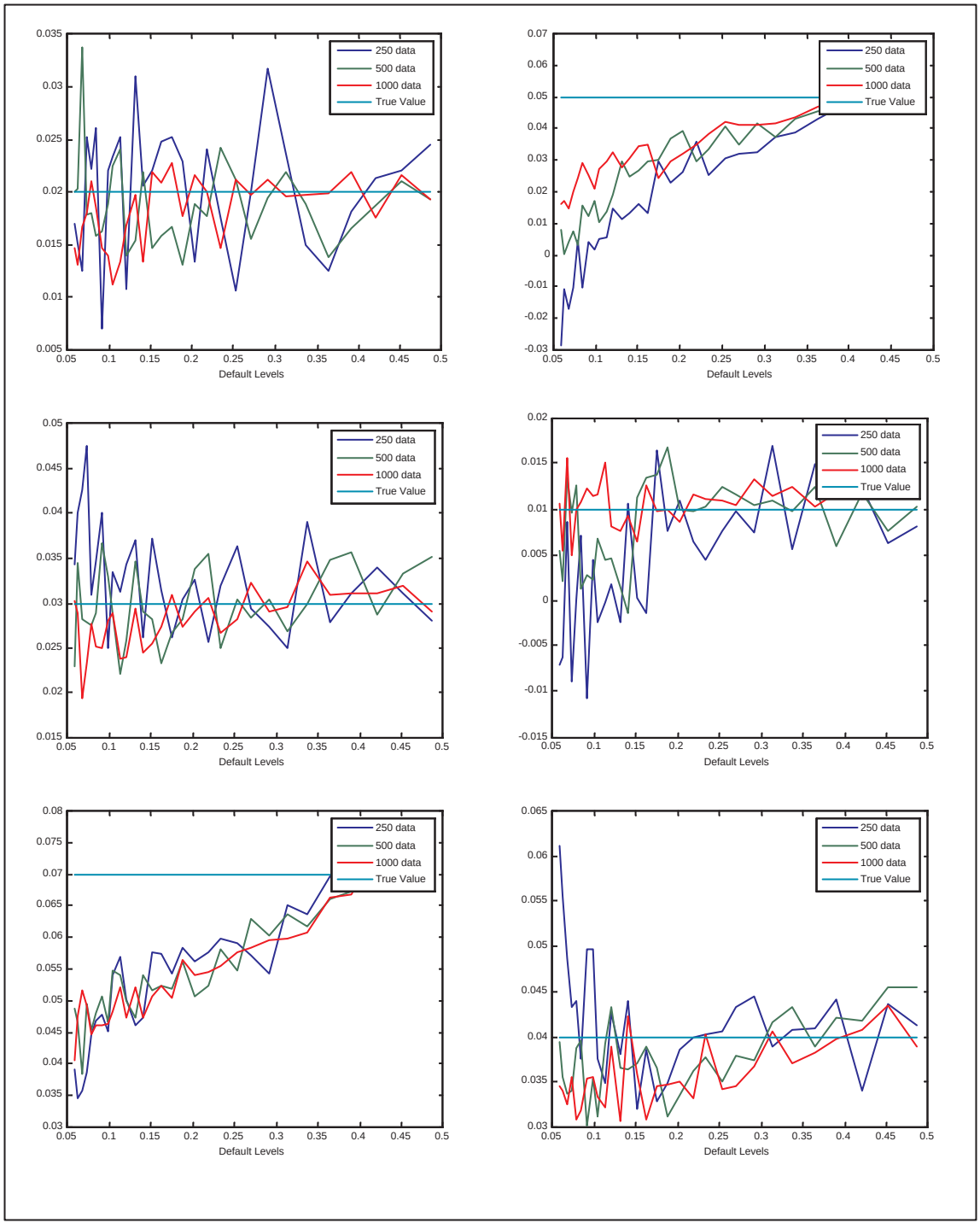

Figure 5.8: *Six variable case:* Coefficients' estimates and their true values in different default levels for the second set of weights.

Third Set of Weights

The initial weights taken in the analysis are $w_1 = 1.3$, $w_2 = 0.2$, $w_3 = -0.7$, $w_4 = -0.9$, $w_5 = 0.6$ and $w_6 = 0.5$

The third set of weights gives high loadings to the variables. To preserve the mean we used negative and positive high values together.

Figure 5.9 represent similar result the when sample size is increased, the bias of estimation decreased. Moreover, for these three sample sizes, it is valid that the bias shows nearly no change after the level of % 30 defaults in the sample.

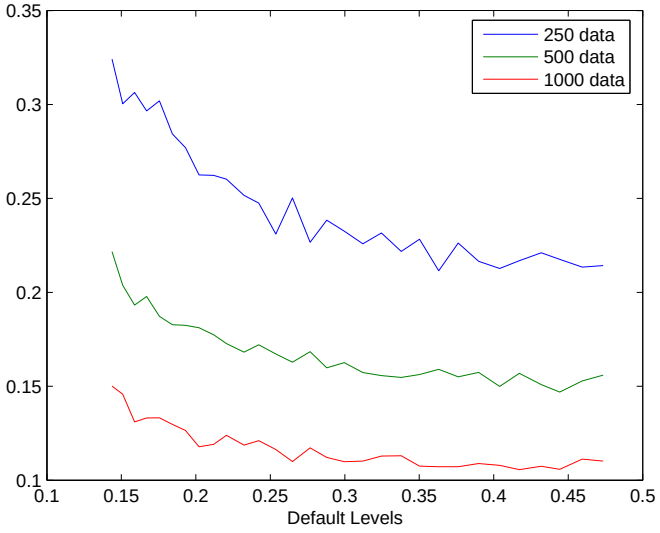

Figure 5.9: *Six variable case:* Average coefficient of variation of estimators in different default levels for the third set of weights.

As shown in Figure 5.10, the estimators are much more different than their true values. This failure of estimation of logistic regression cannot be eliminated much even on the high default levels.

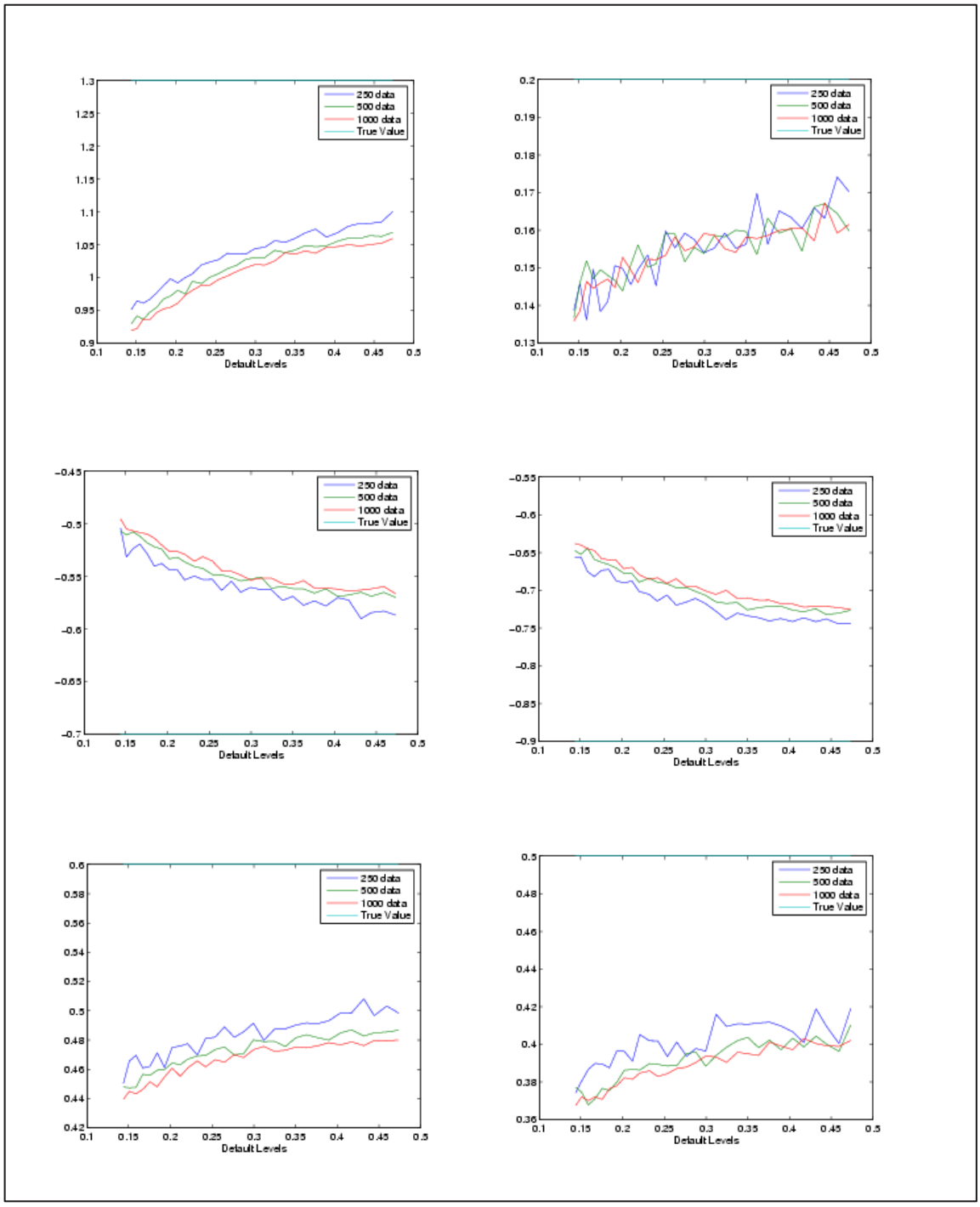

Figure 5.10: *Six variable case:* Coefficients' estimates and their true values in different default levels for the third set of weights.

5.2.3 Twelve Variables Case

The data were generated from a set of random variables from the distributions:

1. $x_1 \sim uniform(1,100)$,
2. $x_2 \sim exponential(16)$,
3. $x_3 \sim normal(12,4)$,
4. $x_4 \sim weilbull(0.5,2)$,
5. $x_5 \sim chi(17)$,
6. $x_6 \sim beta(4,3)$,
7. $x_7 \sim uniform(1,150)$,
8. $x_8 \sim exponential(11)$,
9. $x_9 \sim normal(32,5)$,
10. $x_{10} \sim weilbull(3,1)$,
11. $x_{11} \sim chi(51)$,
12. $x_{12} \sim beta(7,2)$.

First Set of Weights

Our first set of weights for these twelve variable has the elements $w_1 = 0.01$, $w_2 = 0.02$, $w_3 = 0.04$, $w_4 = 0.03$, $w_5 = 0.015$, $w_6 = 0.07$, $w_7 = 0.085$, $w_8 = 0.03$, $w_9 = 0.1$, $w_{10} = 0.24$, $w_{11} = 0.06$ and $w_{12} = 0.3$. These weights are selected to load low effects to variables and preserve the mean.

For the coefficient of variation Figure 5.11 shows a peak for the 250 observation data set which includes smaller than 5% default. Furthermore, until the level of 25% the bias of coefficient estimates are especially very high, but after that it is nearly stable.

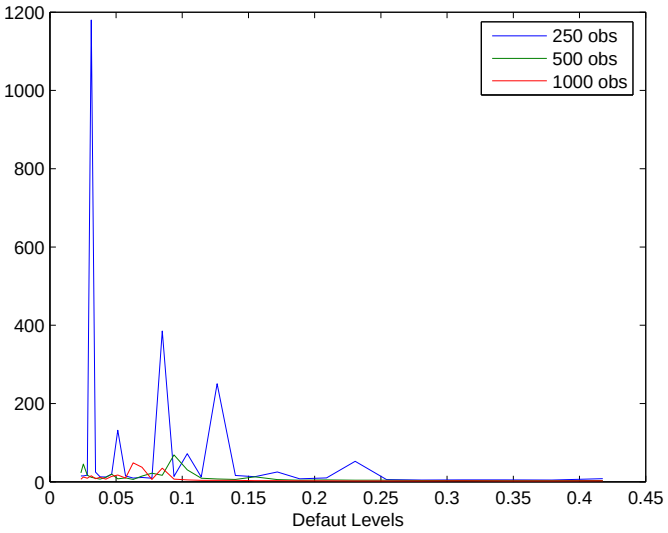

Figure 5.11: *Twelve variable case:* Average coefficient of variation of estimators in different default levels for the first set of weights.

Figures 5.12 and 5.13 represent the coefficient estimates with respect to their true values. Accordingly, it can be observed that for these low weights, the estimation accuracy for nearly all default levels and data sizes is perfect.

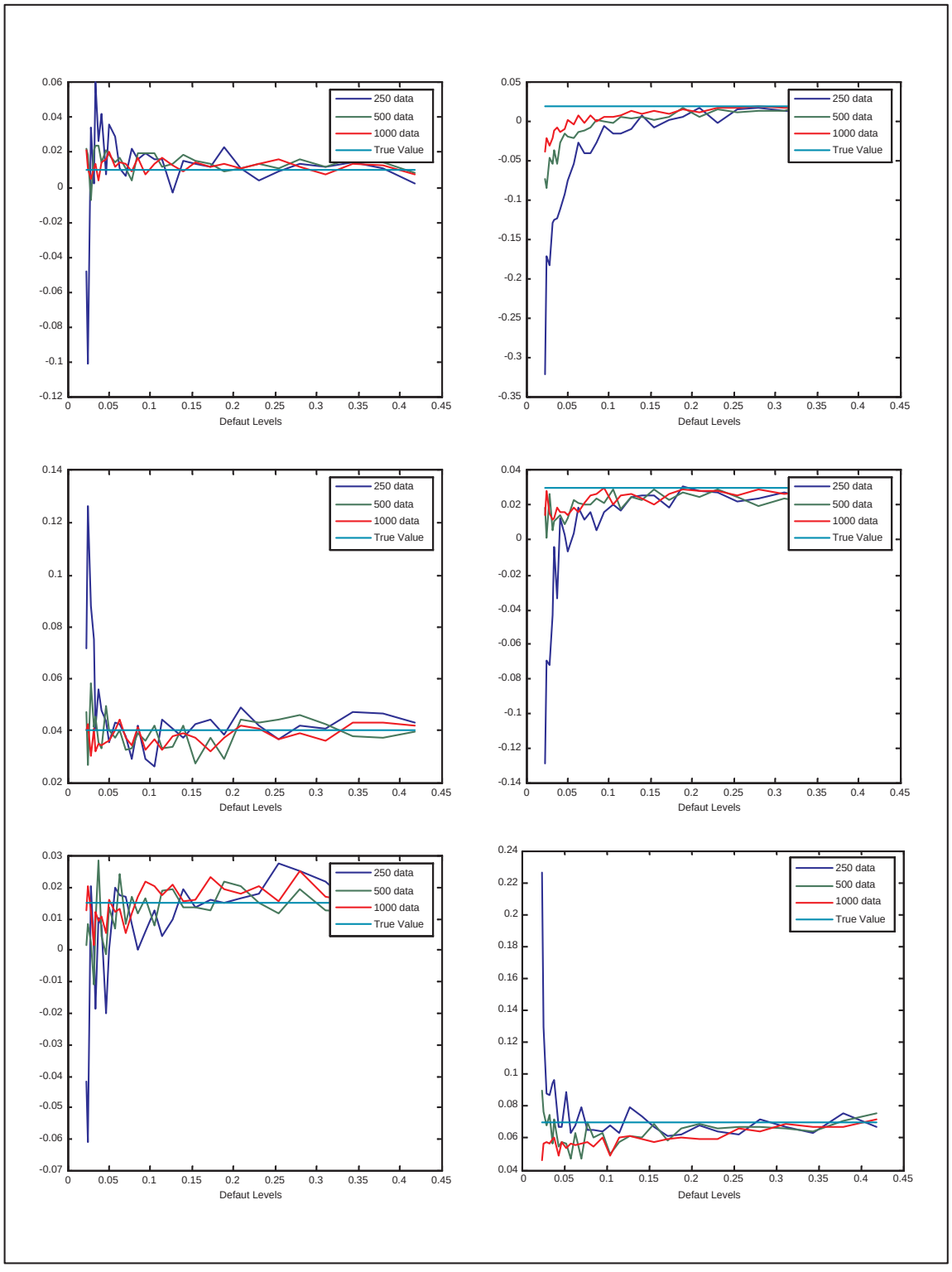

Figure 5.12: *Twelve variable case:* First six coefficients' estimates and their true values in different default levels for the first set of weights.

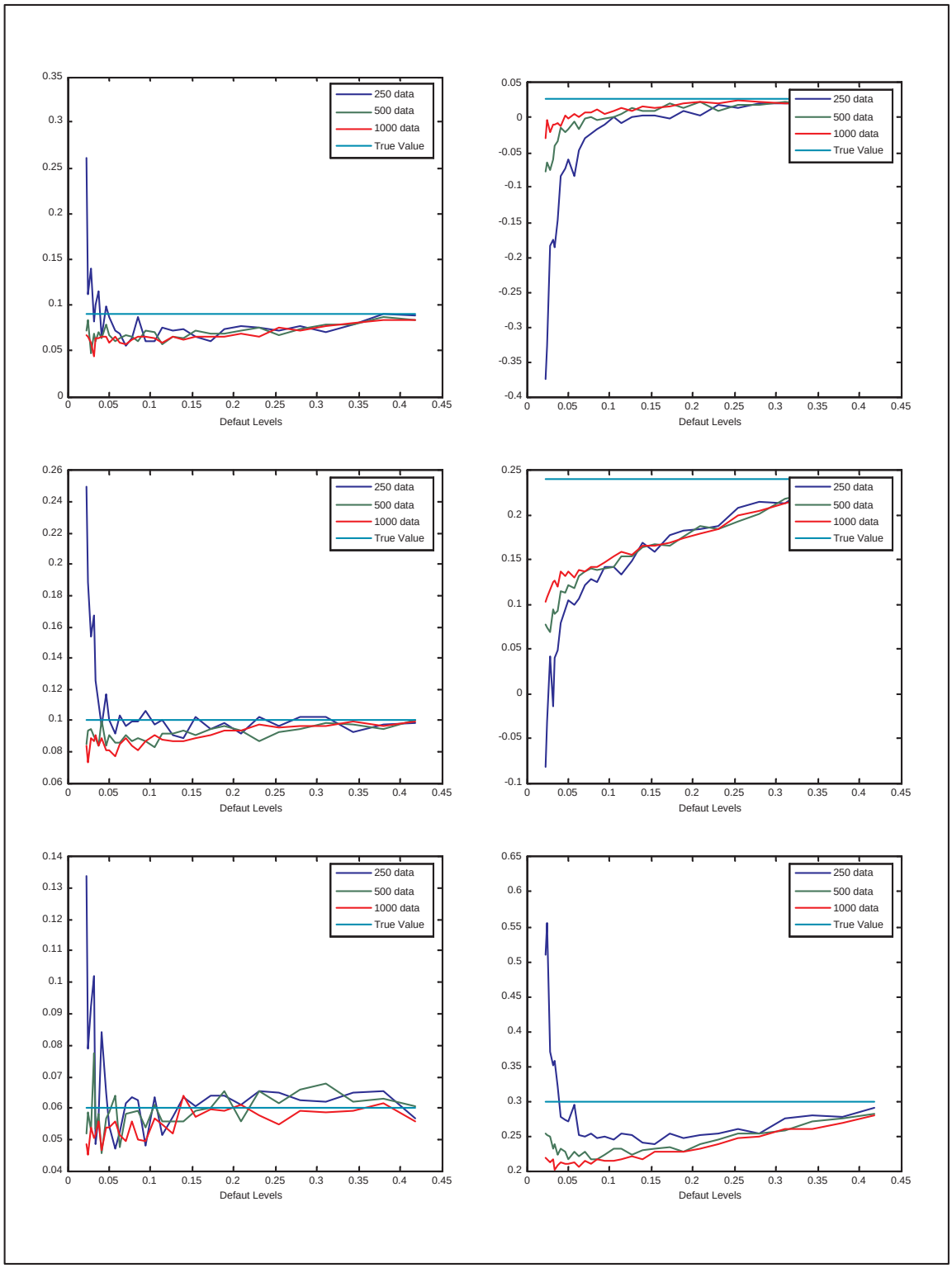

Figure 5.13: *Twelve variable case:* Second six coefficients' estimates and their true values in different default levels for the first set of weights.

Second Set of Weights

As second set of weights, we took: $w_1 = 0.9$, $w_2 = 1.2$, $w_3 = -0.7$, $w_4 = 0.85$, $w_5 = -1.1$, $w_6 = -1.1$, $w_7 = 0.65$, $w_8 = -0.5$, $w_9 = 1.3$, $w_{10} = 0.7$, $w_{11} = -1.7$ and $w_{12} = 0.8$. This set includes very high positive and negative weights. The mean is preserved also.

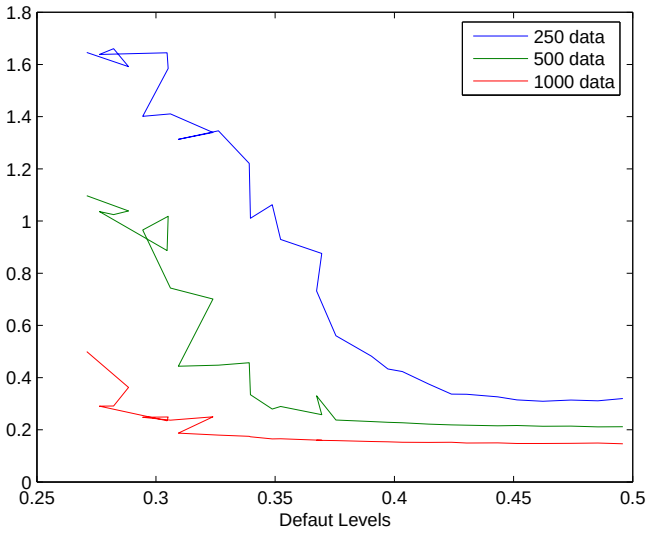

Figure 5.14: *Twelve variable case:* Average coefficient of variation of estimators in different default levels for the second set of weights.

From Figure 5.14, we observe that if the independent variables have high effects on dependent variables, the bias is fixed after the 30% default level for data sets with 500 and 1000 observations, i.e. for the larger data sets, and it is fixed after the 40% default level for smaller data sets.

Moreover, Figures 5.15 and 5.16 indicate very bad estimations of weights. The smaller data sets give more accurate estimation results. Furthermore, the best results are taken around the 30% default level for all variables.

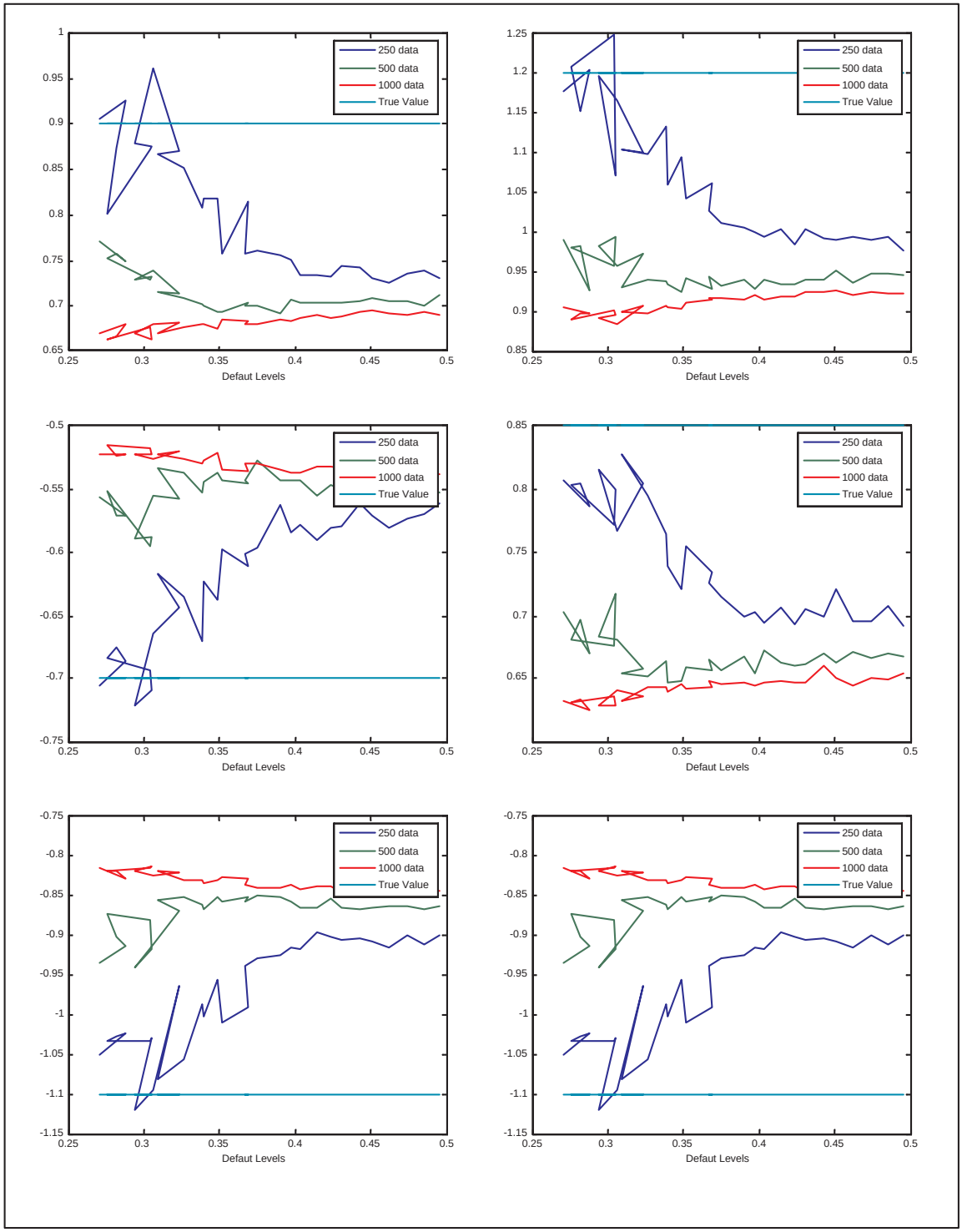

Figure 5.15: *Twelve variable case:* First six coefficients' estimates and their true values in different default levels for the second set of weights.

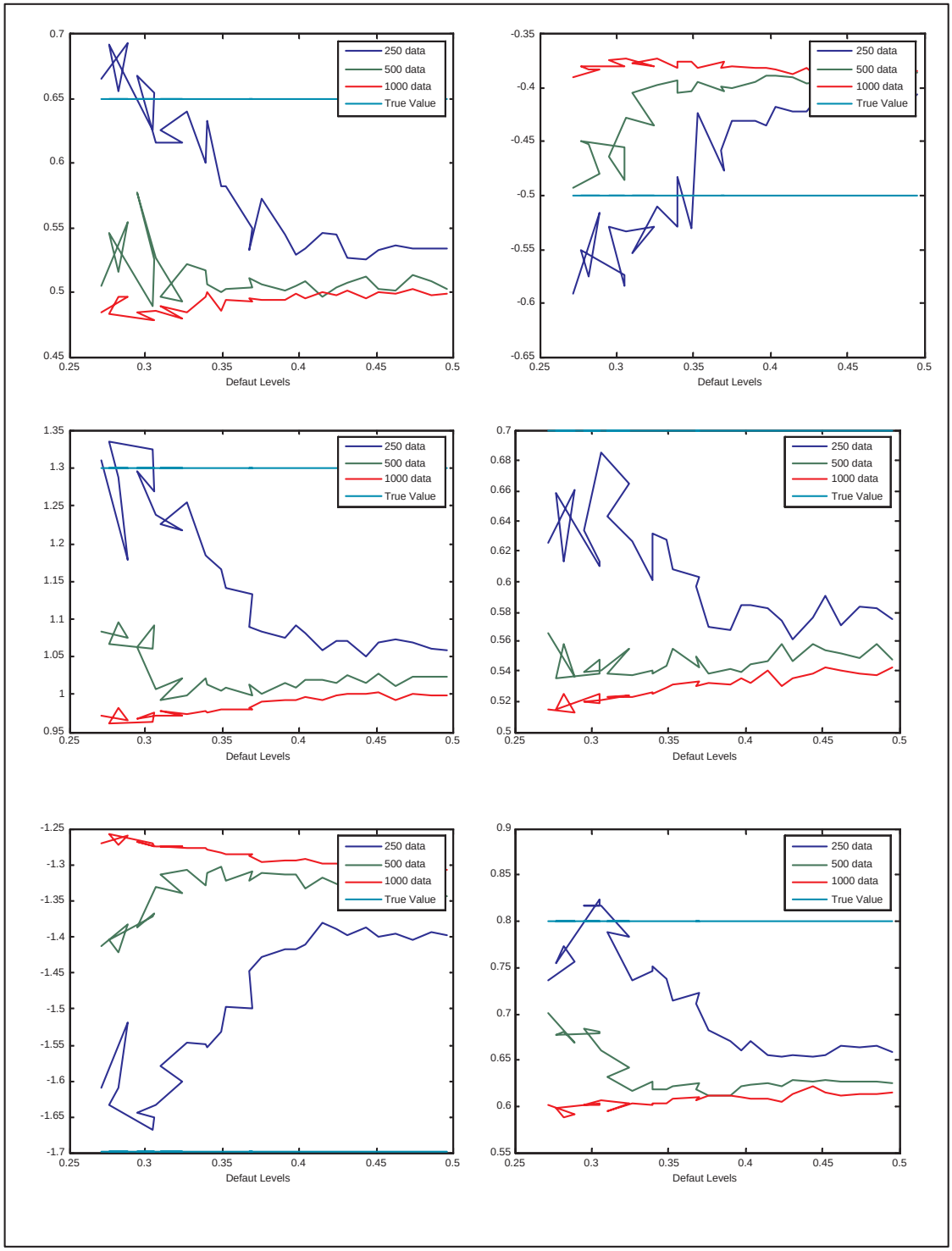

Figure 5.16: *Twelve variable case:* Second six coefficients' estimates and their true values in different default levels for the second set of weights.

Third Set of Weights

The third set of weights is the following: $w_1 = 0.1$, $w_2 = 0.2$, $w_3 = 0.4$, $w_4 = 0.3$, $w_5 = -0.2$, $w_6 = -0.5$, $w_7 = -0.25$, $w_8 = -0.15$, $w_9 = 0.35$, $w_{10} = 0.47$, $w_{11} = 0.18$ and $w_{12} = 0.1$. This set of weights is selected for showing the prediction accuracy in the data sets in which some variables have high, some have low effects, some variables have negative and some have positive effects.

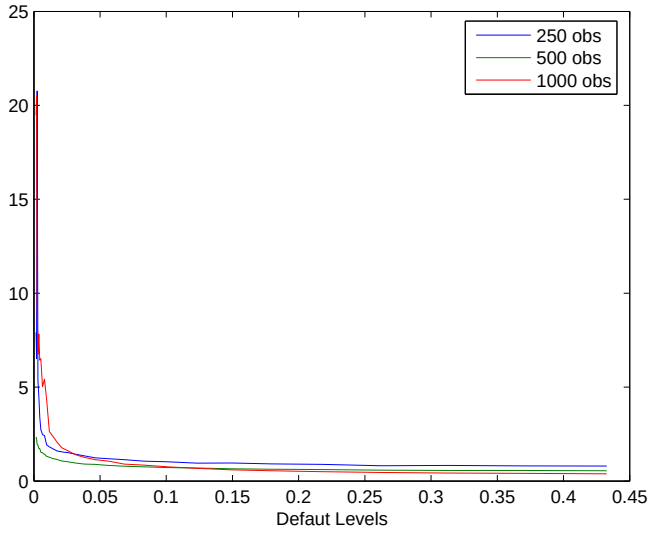

Figure 5.17: *Twelve variable case:* Average coefficient of variation of estimators in different default levels for the third set of weights.

According to above Figure 5.14, the average coefficient of variation of the variables has peak for all size of data sets which includes lower than 5% defaults in it. After the default cut-off point 25%, the bias follows a stable manner.

Figures 5.15 and 5.16 show estimators and their true values. If we made a comparison between the estimated and true value of all, we can say that the estimation accuracy of loadings is very good and when the number of default cases increases in the data set, the coefficient estimates converge to their true values. However, after the level of nearly 30 %, there are no big changes in the convergence.

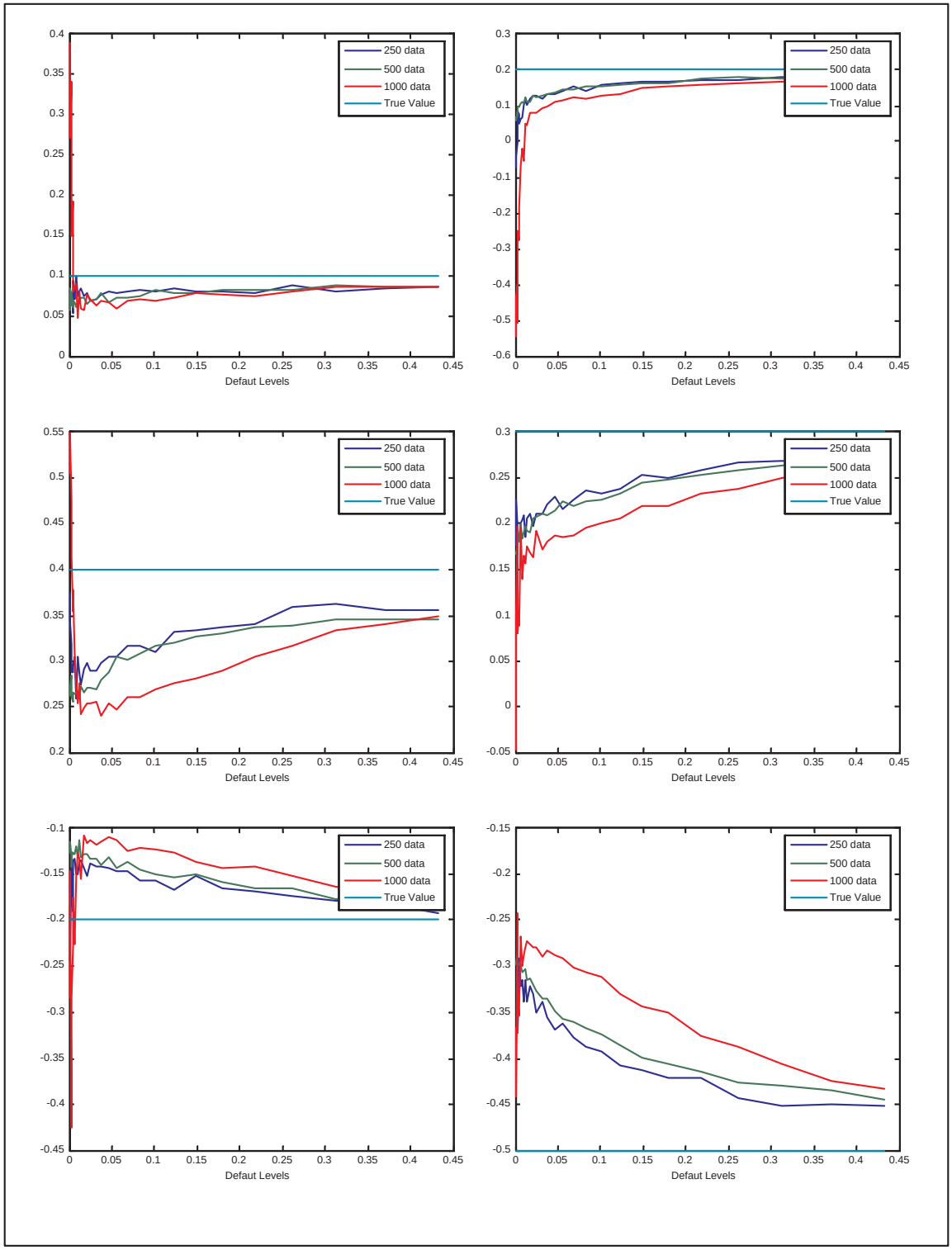

Figure 5.18: *Twelve variable case:* First six coefficients' estimates and their true values in different default levels for the third set of weights.

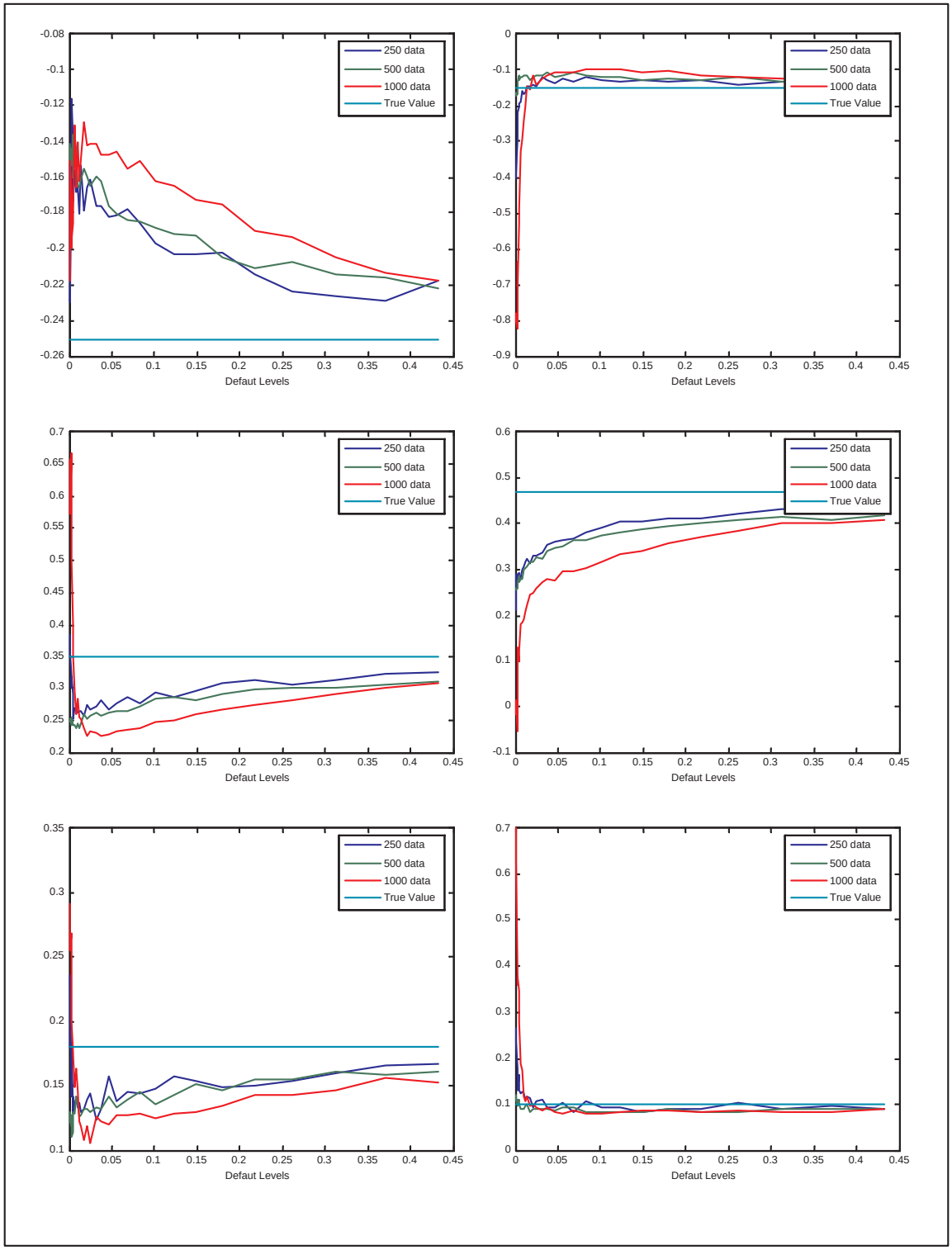

Figure 5.19: *Twelve variable case:* Second six coefficients' estimates and their true values in different default levels for the third set of weights.

5.3 Summary and Conclusion

As a whole, for all three cases: one variable, six variables and twelve variables, the results concluded are *very similar*. The higher loadings destroy the precision accuracy of logistic regression and precision is getting worse when the number of variables included increase. The more accurate results are taken with a small set of weights. Moreover, the larger sized data sets show lower bias and more convergent estimations of weights. Furthermore, if the data set includes nearly 30% default cases in it, the bias reaches its minimum for some cases or gets its optimal level. As a *result* of these, we can conclude that to imply the logistic regression it is good to select data sets with at least 30 % default cases in it, with minimum 500 observations and with a small number of variables or smaller effects on probability of defaults.

CHAPTER 6

ACCURACY RATIO AND METHOD VALIDATIONS

6.1 Introduction

In credit scoring, the most important part is the discriminative power of the methods. Since the methods are used to evaluate the credit worthiness for taking credit decisions and any classification errors can create damages to the resources of an credit institute [EHT02]. Therefore, in this chapter we focus on the validation techniques and validation of the methods which are introduced in the previous chapters.

6.2 Validation Techniques

In studies, various evaluation techniques can be found. However, the most popular ones are ***Cumulative Accuracy Profile*** and ***Receiver Operating Characteristic Curve***.

Let a method assign to each applicant a score out of k possible values $\{s_1, s_2, ..., s_k\}$ with $s_1 < s_2 < ... < s_k$. For example, let $k = 10$, then, $s_{10} = AAA$, $s_9 = AA$,... and $s_1 = D$. AAA is highest rating and D is the lowest.

Let us introduce the random variables, S_T, S_D and S_{ND} as model score distributions of all, defaulters and non-defaulters, respectively. The probability that a default has a score s_i where $i = 1, 2, ..., k$ is denoted by p_D^i, $\sum_{i=1}^{k} p_D^i = 1$. The probability that a non-default has a score s_i $(i = 1, 2, ..., k)$ is p_{ND}^i. Given the priori default probability π of all applicants, the probability p_T^i can be obtained

by the following way:

$$p_T^i = \pi p_D^i + (1-\pi) p_{ND}^i. \tag{6.2.1}$$

Then, the cumulative probabilities are

$$CD_D^i = \sum_{j=1}^{i} p_D^j \qquad (i = 1, 2, ..., k) \tag{6.2.2}$$

$$CD_{ND}^i = \sum_{j=1}^{i} p_{ND}^j \qquad (i = 1, 2, ..., k) \tag{6.2.3}$$

$$CD_T^i = \sum_{j=1}^{i} p_T^j \qquad (i = 1, 2, ..., k), \tag{6.2.4}$$

where CD_D, CD_{ND} and CD_T denote the cumulative distribution functions of the score values of the default, non-default and total applicants, respectively.

Let us understand these probabilities by an example.

Example 6.1. In a market, let 10 different rating classes. $s_{10} = AAA$, $s_9 = AA$, $s_8 = A$, $s_7 = BBB$, $s_6 = BB$, $s_5 = B$, $s_4 = CCC$, $s_3 = CC$, $s_9 = C$ and $s_1 = D$. The number of companies from each rating classes are summarized in Table 6.1.

Rating Class	Total	Number of default cases	Number of non-default cases
AAA	5	0	5
AA	12	1	11
A	20	3	17
BBB	32	9	23
BB	27	7	20
B	34	8	26
CCC	67	17	50
CC	42	20	22
C	17	9	8
D	12	10	2
Total	268	84	184

Table 6.1: The rating classes and total number of observations.

Then,

$$p_D^1 = 10/84,$$

$$p_D^2 = 9/84,$$

$$\vdots$$

$$p_D^{10} = 0/84$$

and

$$\sum_{i=1}^{10} p_D^i = 10/84 + 9/84 + ... + 0 = 1.$$

Similarly,

$$p_{ND}^1 = 2/184,$$

$$p_{ND}^2 = 8/184,$$

$$\vdots$$

$$p_{ND}^{10} = 5/184$$

and

$$\sum_{i=1}^{10} p_{ND}^i = 2/184 + 8/184 + ... + 5/184 = 1.$$

Furthermore, if we select the $\pi = 0.4$, the total probabilities will be

$$p_T^1 = 0.4p_D^1 + 0.6p_{ND}^1 = 0.0541,$$

$$p_T^2 = 0.4p_D^2 + 0.6p_{ND}^2 = 0.0689,$$

$$\vdots$$

$$p_T^{10} = 0.4p_D^{10} + 0.6p_{ND}^{10} = 0.0163.$$

The distribution functions are shown in Table 6.2. We assume $CD_D^0 = CD_{ND}^0 = CD_T^0 = 0$.

6.2.1 Cumulative Accuracy Profile (CAP)

The *cumulative accuracy profile* is basically defined as the graph of all points $(CD_T^i, CD_D^i)_{i=1,2,...,k}$ where the points are connected by linear interpolation [EHT02].

i	CD_T	CD_D	CD_{ND}
1	0,0541	0,119	0,0109
2	0,123	0,2261	0,0544
3	0,29	0,4642	0,174
4	0,534	0,6666	0,4457
5	0,6569	0,7618	0,587
6	0,7555	0,8451	0,6957
7	0,8734	0,9522	0,8207
8	0,9431	0,9879	0,9131
9	0,9837	0,9998	0,9729
10	1	1	1

Table 6.2: The cumulative probability functions.

The CAP of the Example 6.1 and the *random model* are shown in the following Figure (6.1).

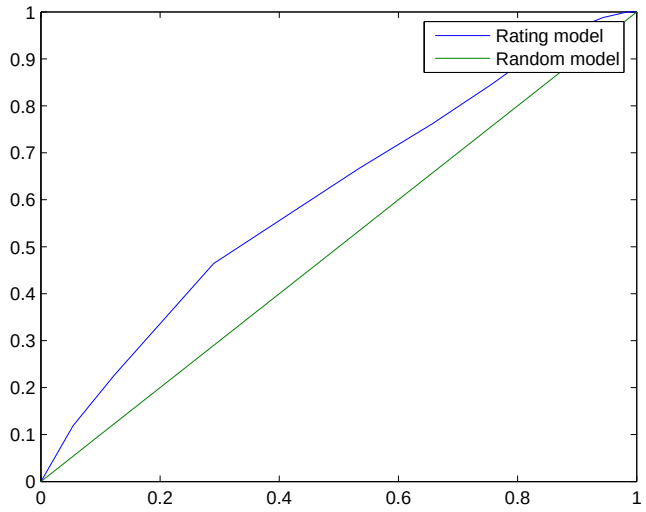

Figure 6.1: Cumulative accuracy profile curve of the Example (6.1).

In the random model of rating, the lowest α percentage of all companies in the research contains the α percentage of the all defaults.

The assignment of the lowest scores to the actual default companies is a measure of the quality of rating methods and called the *accuracy ratio (AR)*. The

accuracy ratio in fact is the monotone transformation of the method's strength and weakness to a one dimensional measure [SKSt00]. From the case in Figure 6.2, the accuracy ratio is defined as the ratio of the area α_R between the CAP of the rating model being applied and the CAP of the random model, and the area α_P between the CAP of the perfect rating model and the CAP of the random model, i.e.,

$$AR = \frac{\alpha_R}{\alpha_P}. \tag{6.2.5}$$

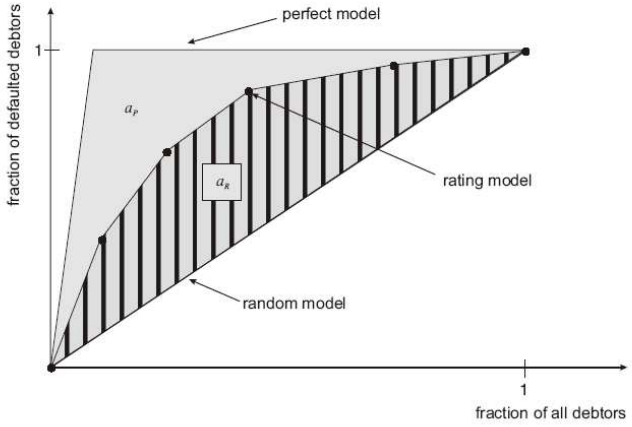

Figure 6.2: Cumulative accuracy profile [EHT02].

6.2.2 Receiver Operating Characteristic Curve (ROC)

Let us consider Figure 6.3. This figure depicts the distributions of rating scores for default and non-default companies. For a perfect rating model the distributions would not overlap like it instead they would be separate. The grey areas shows default companies and white areas shows the non-default companies. Furthermore, C denotes the cut-off point. Usually, the companies which have scores lower than C are classified as default and the companies with higher scores than C are classified as non-default.

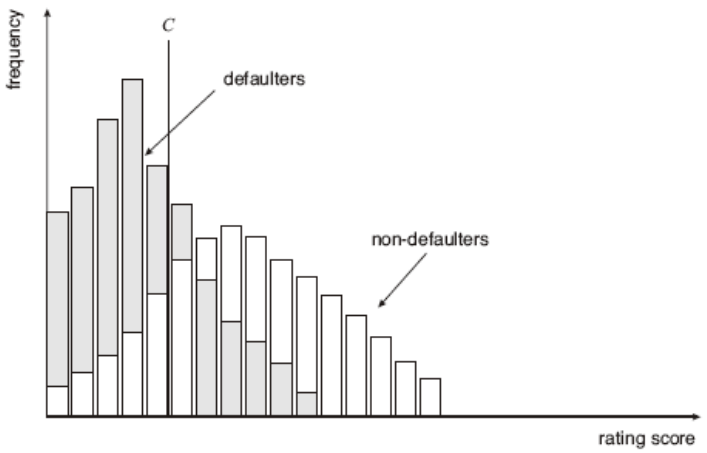

Figure 6.3: Distribution of rating scores for defaulting and non-defaulting debtors [EHT02].

		default	non-default
rating score	below C	correct prediction (hit)	wrong prediction (false alarm)
	above C	wrong prediction (miss)	correct prediction (correct rejection)

Table 6.3: Decision results given the cut-off value C [EHT02].

Table 6.3 summarizes all possible decision results. Accordingly, if the score of a company is below the C and the company default in the next period, the decision was correct and this situation is called an hit. Otherwise, the decision is wrong and called as a false alarm. Similarly, if the score is beyond C and the company does not default, the correct prediction is made. Otherwise the decision is wrong.

To construct the ROC curve let us firstly define the *hit rate (HR(C))* at C as

$$HR(C) = P(S_D \leq C), \tag{6.2.6}$$

and the *false alarm rate (FAR(C))* at C as

$$FAR(C) = P(S_{ND} \leq C). \tag{6.2.7}$$

Then, the ROC curve is defined as a plot of HR(C) with respect to FAR(C) for all values of C [EHT02]. Figure 6.4 shows an example of a ROC curve. This figure indicates that the rating model gives the results between the random model and the perfect model. The nearer the graph of the rating model is to the perfect one, the better are the predictions of the rating model.

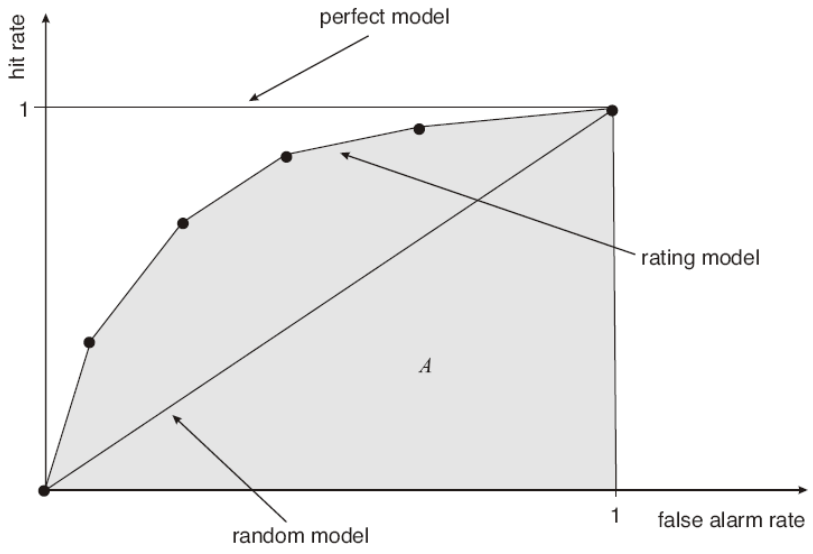

Figure 6.4: Receiver operating characteristic curve [EHT02].

For Example 6.1, let us construct a ROC curve. The distributions for default and non-default companies are shown in Figure 6.5. Accordingly, we can say that in our example rating system the perfect discrimination is not possible. the distributions overlap.

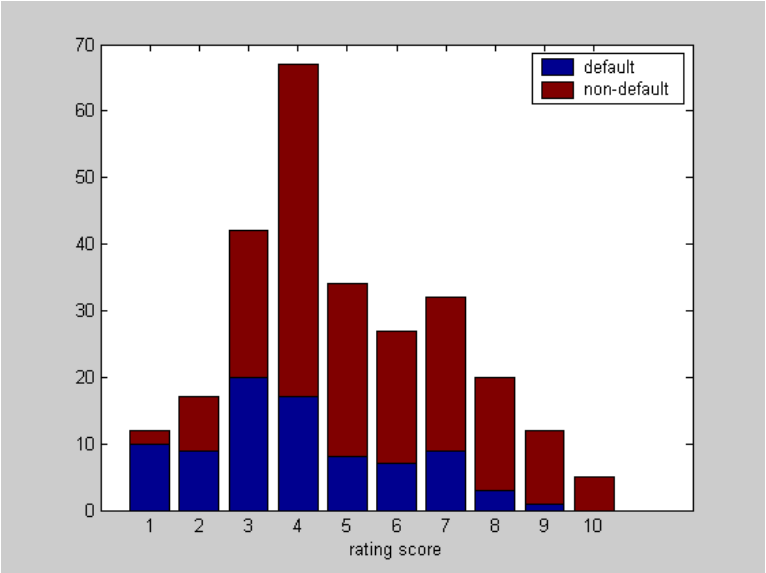

Figure 6.5: Distribution of rating scores for Example 6.1.

In our construction process, we can take the cut-off values as $C_1 = 1$, $C_2 = 2$, ... and $C_{10} = 10$. Then, the hit rate $\mathrm{HR}(C_i)$ $(i = 1, 2, ..., 10)$ from Table 6.1is

$$HR(C_1) = 10/84, \qquad FAR(C_1) = 2/184;$$

$$HR(C_2) = 19/84, \qquad FAR(C_2) = 10/184;$$

$$\vdots$$

$$HR(C_{10}) = 1, \qquad FAR(C_1) = 1.$$

Then, we can visualize the receiver operating characteristic curve in Figure 6.6.

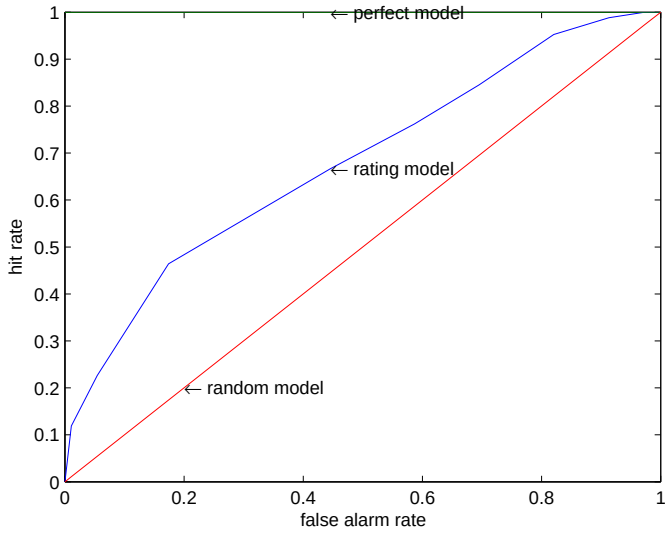

Figure 6.6: Receiver operating characteristic curve for Example 6.1.

6.3 Method Validations

6.3.1 Data and Methodology

Some statistical and non-statistical credit scoring methods with their advantages and disadvantages were mentioned in previous chapters. However, we did not consider which one is the best in scoring. In this section, the validation of methods mentioned in *Chapter 3* and *Chapter 4: discriminant analysis, linear regression, probit regression, logistic regression, semi-parametric logistic regression, classification and regression trees* and *neural networks* will be presented and applied.

The data used in this part were collected from the period of 1995-2005. The data set includes the financial situations of 1000 companies. Among of them, 247 are defaulters. The data set includes 17 explanatory variables:

I. Liquidity Ratios:

X_1: *current ratio*,

X_2: *liquidity ratio*,

X_3: *cash ratio*.

II. Activity Ratios:

X_4: *receivables turnover ratio*,

X_5: *inventory turnover ratio*,

X_6: *fixed-assets turnover ratio*,

X_7: *equidity turnover ratio*,

X_8: *(total liability)/(total assets)*,

X_{10}: *(total liability)/equidity* .

III. Profitability Ratios:

X_{11}: *gross merchandise margin*,

X_{12}: *net profit margin*,

X_{13}: *active profitability ratio*,

X_{14}: *equidity profitability ratio*.

IV. Growth Ratios:

X_{15}: *increase in sales*,

X_{16}: *active growth rate*,

X_{17}: *equidity growth rate*.

To obtain validation results, we follow a type of ***cross-validation simulations*** with *1000 simulations* for every method. In cross-validation methodology, we applied the following steps:

Step 1: The random data set of size 1000 are constructed from the actual data set with the help of the uniform random numbers.

Step 2: The random data sets of size 250 and 500 are selected from the selected random sample of size 1000.

Step 3: The 80% of the data sets are used as training samples and the other 20% are used as validation samples. The method is applied to the all three different sized data sets.

Step 4: The mean square errors (MSE) and classification accuracies are observed for each set.

Step 5: The steps 1, 2, 3 and 4 are repeated 1000 times and the MSE, and the classification accuracies are averaged.

6.3.2 Results

The average MSEs and accuracies of classification techniques are summarized in Table 6.4. The estimations were done in a way that the significance of coefficients were not compared. Because we tried to compare models instead of looking significance of the variables on credit scoring. Moreover, since the discriminant analysis and classification and regression trees (CART) give the outputs 0 or 1 only, we did not calculate the mean square errors for these methods. According to this table, the discriminant analysis gave nearly 96% accuracy in training samples and 95% accuracy in validation samples of small sizes, i.e., 250 and 500. However, its accuracy is dropping 4% for large samples. A similar situation is also valid for neural network. For small sample sizes, the accuracy of it is much higher. Furthermore, for neural networks, when sample size is getting larger, the MSEs are rising sharply.

The most interesting method is CART. It turns out to be a the perfect model in the sample of size 1000. However, for other sizes its accuracy in scoring is very low. It is about 60%.

In addition to CART, the semi-parametric regression method with logistic link is the other method which gave the worst results both in accuracy and MSEs. Its classification accuracy is between 70% and 80% for all sample sizes.

Moreover, among the all methods the logistic regression is the best scoring methods. Its accuracy is above the 98% for all sizes. The closer method is the probit. Its accuracy in estimation is also very high. However, for large samples its accuracy decreases.

To check the validation of the methods we constructed receiver operating characteristic curves for each method in sample of 1000. Figures 6.7 and 6.8 show the plotted ROC curves for validation and training samples for our analysis, respectively.

From these figures, we could not be able to observe any big differences in the validation and training sample results. The graphs shows us CART is the perfect scoring method and logistic regression is near to it. The probit is also near to the perfect model of classification. However, the graphs of semi-parametric regression

Method	data size	training sample		validation sample	
		MSE	accuracy	MSE	accuracy
Discriminant Analysis	250	-	96.64%	-	95.35%
	500	-	96.19%	-	95.67%
	1000	-	92.53%	-	92.00%
Linear Regression	250	11.663	93.20%	4.3522	91.59%
	500	24.712	92.56%	6.8408	91.84%
	1000	51.107	92.26%	13.419	91.72%
Probit Regression	250	2.8805	98.23%	1.4438	96.56%
	500	8.0458	97.42%	2.5722	96.72%
	1000	19.4081	96.96%	5.3519	96.62%
Logistic Regression	250	1.1472	99.51%	0.9881	97.99%
	500	3.3116	99.31%	1.2670	98.76%
	1000	7.6038	99.08%	2.2173	98.82%
CART	250	-	63.48 %	-	63.66%
	500	-	63.31%	-	63.43%
	1000	-	100%	-	100%
Semi-parametric regression	250	56	72%	10	80%
	500	90	78%	18	83%
	1000	182	78.13%	45	78.1%
Neural Networks	250	9.3362	98.5%	4.0003	96%
	500	52.8	95.42%	17.921	93.2%
	1000	101.08	93.5%	32.211	92.5%

Table 6.4: The MSE and accuracy results of methods.

are far from the perfect model so it is the worst model both in validation and training samples.

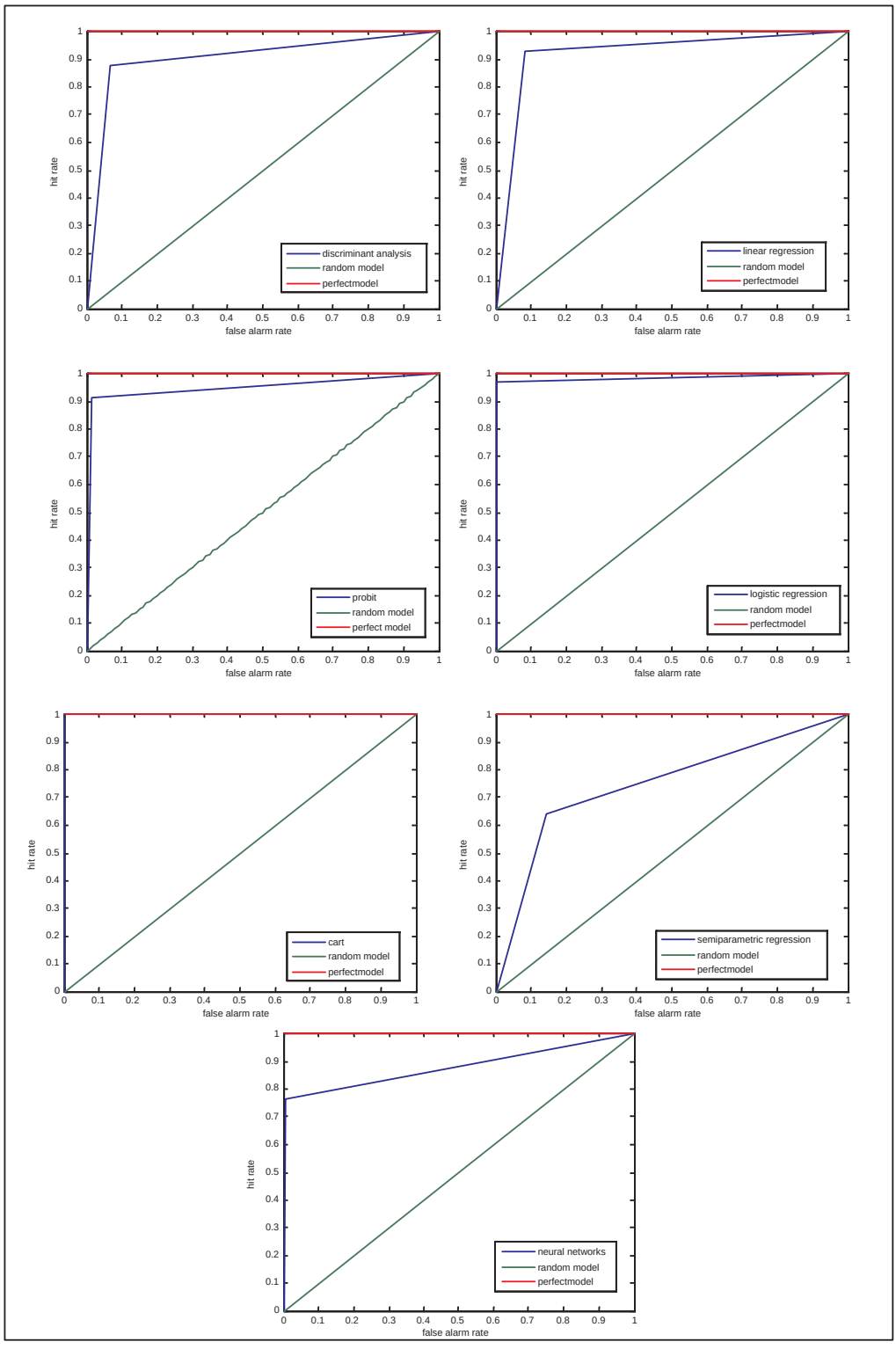

Figure 6.7: Receiver operating characteristic curves for *validation sample* of size 1000.

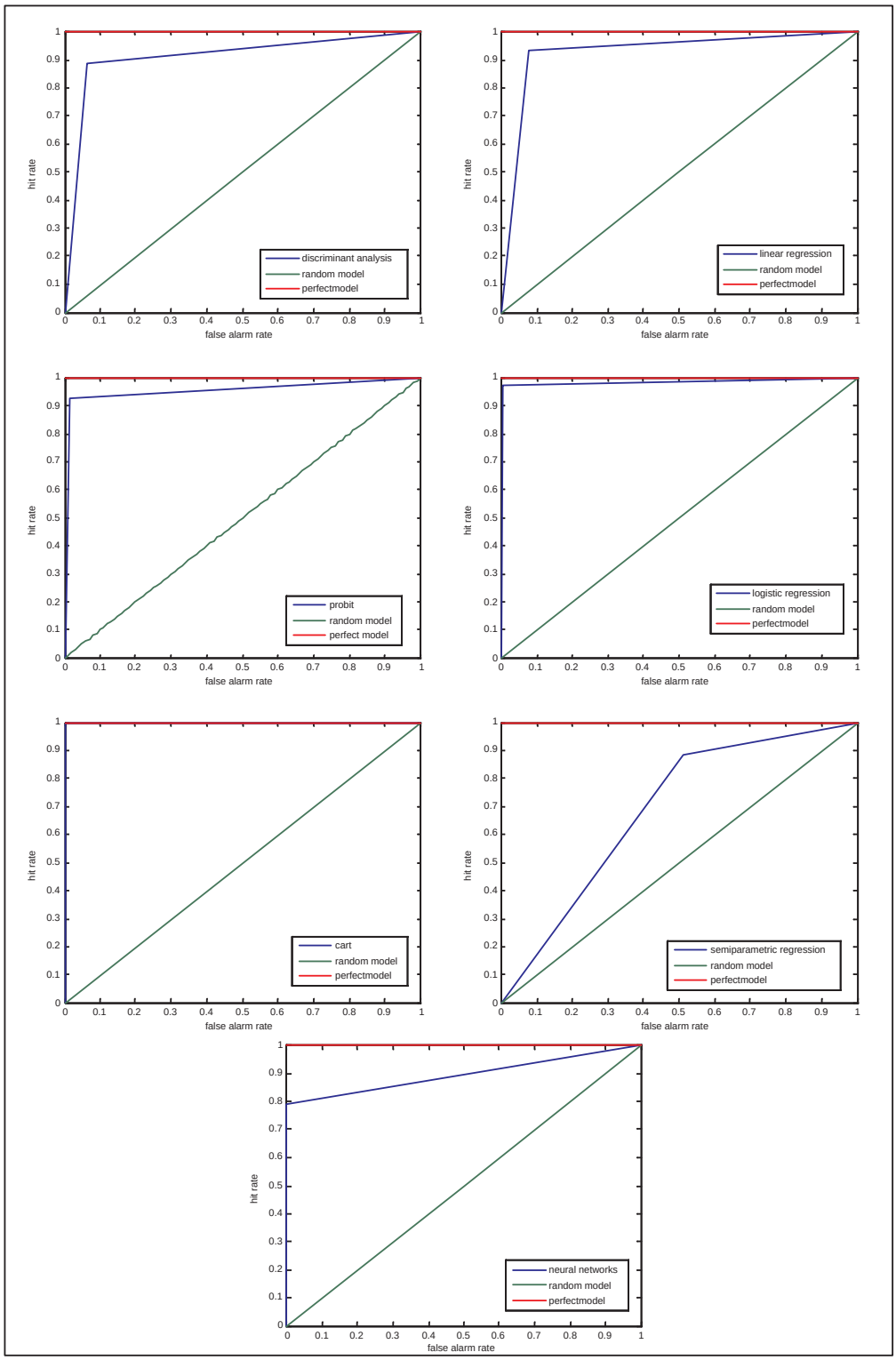

Figure 6.8: Receiver operating characteristic curves for *training sample* for size 1000.

6.3.3 Ratings via Logistic and Probit Regressions

In conclusion, results shows that the logistic regression provides the best accuracy results in validations. Therefore, for our data set, it is good to use it as a final conclusion and ratings. Moreover, to make a comment and comparison on its results, we also apply the probit regression.

In here, we also take 20 % of the data set as validation sample and remaining 80% of the set as training sample. Table 6.5 summarizes coefficient estimates and p-values. Accordingly, p-values of logistic regression show us the variables x_5, x_8, x_{11} and x_{15} are not significant on estimation. Similarly, the coefficients of x_3, x_5, x_8 and x_{14} are not significant at $\alpha = 0.05$. To make a comparison of these two regression models, we excluded only the variables x_5 and x_8 which are insignificant for both models and we fit the regressions with the remaining coefficients.

Variables	Logistic Regression		Probit Regression	
	coefficients	p-value	coefficients	p-value
x_0	-18.3391	0.0001	-7.0792	<0.0001
x_1	0.0664	<0.0001	0.0162	0.0045
x_2	-0.1272	<0.0001	-0.0327	<0.0001
x_3	-0.1031	0.0042	-0.0209	0.1081
x_4	0.0485	0.0001	0.0071	0.0251
x_5	-0.0710	0.2250	-0.0252	0.1037
x_6	-0.1343	<0.0001	-0.0479	0.0016
x_7	0.0332	<0.0001	0.0080	0.0005
x_8	0.0346	0.1945	0.0138	0.1174
x_9	0.0144	<0.0001	0.0061	<0.0001
x_{10}	0.1845	<0.0001	0.0646	<0.0001
x_{11}	0.0091	0.2933	0.0105	0.0251

Variables	Logistic Regression		Probit Regression	
	coefficients	p-value	coefficients	p-value
x_{12}	-0.0432	0.0182	0.0344	<0.0001
x_{13}	-0.1650	<0.0001	-0.0328	<0.0001
x_{14}	0.0455	0.0153	0.0116	0.1512
x_{15}	0.0032	0.1277	0.0015	0.0084
x_{16}	-0.0057	0.0016	-0.0031	0.0049
x_{17}	0.0262	<0.0001	0.0072	<0.0001

Table 6.5: The coefficients and p-values of logistic and probit regression.

Tables 6.6 and 6.7 present the classification errors of both models for both training and validation samples. Accordingly, the classification errors of non-default firms in training and validation samples are same for logistic and probit regression models. However, for default firms when logistic regression gives the no error, the probit regression gives important misclassification errors in both sample estimates at probability cut-off 0.5. In fact instead of this cut-off, the optimum cut-off should be used. However, to see some failures of probit we used 0.5. Furthermore, the results indicate the low probability assignment problem of probit regression to the default firms.

Variables	Logistic Regression		Probit Regression	
	default	non-default	default	non-default
default	198	0	170	28
non-default	7	595	7	595

Table 6.6: Training sample classification results.

The ratings are given to the responses according to estimated probabilities of both models. To select the optimum cut-off probabilities of rating categories we implement the following algorithm.

Variables	Logistic Regression		Probit Regression	
	default	non-default	default	non-default
default	49	0	42	7
non-default	2	149	2	149

Table 6.7: Validation sample classification results.

This algorithm firstly observes the repeated probabilities and according to them splits the data into several parts and thinks these probabilities as cut-offs. Then, algorithm tries different combinations of these cut-off probabilities and selects the optimum split in a way that optimum cut-off is the one which maximizes the area under the *receiver operating characteristic* curve. We run algorithm for 8, 9 and 10 rating categories in our analysis, and also compare the optimum number of categories.

The optimum cut-off probabilities for logistic and probit regression for 10 rating classes are summarized in Tables 6.8 and 6.9, respectively:

Ratings	Probability of defaults from logistic regression			
	validation sample		training sample	
	lower range	upper range	lower range	upper range
AAA	0.000	$2.3031*10^{-10}$	0.000	$6.27*10^{-12}$
AA	$2.3032*10^{-10}$	$1.0721110*10^{-8}$	$6.28*10^{-12}$	$1.03207*10^{-8}$
A	$1.072111*10^{-8}$	$1.4504844*10^{-7}$	$1.03207*10^{-8}$	$9.937391*10^{-6}$
BBB	$1.4504845*10^{-7}$	$3.86603848*10^{-6}$	$9.937392*10^{-6}$	0.0000014
BB	$3.86603849*10^{-6}$	0.0006369	0.0000015	0.0000232
B	0.0006369	0.007774	0.0000233	0.000139
CCC	0.007775	0.007806	0.004653	0.0001391
CC	0.0078067	0.073298	0.0001391	0.020088
C	0.073299	0.77553	0.020089	0.60002
D	0.77554	1.000	0.60003	1.000

Table 6.8: Optimum cut-off probability of defaults of logistic regression for 10 rating category.

From Tables 6.8 and 6.9 we observe that logistic regression is more sensitive to the probability of default and assigns high ratings for exactly low probabilities. However, probit regression classifies only very high probabilities to the low rating classes. Table 6.10

Ratings	Probability of defaults from probit regression			
	validation sample		training sample	
	lower range	upper range	lower range	upper range
AAA	0.000	$2.346*10^{-11}$	0.000	$2.346*10^{-11}$
AA	$2.346*10^{-11}$	$5.64459*10^{-7}$	$2.346*10^{-11}$	$2,07457*10^{-7}$
A	$5.64459*10^{-7}$	$4.20266*10^{-5}$	$2.07457*10^{-7}$	$1.21604*10^{-5}$
BBB	$4.20266*10^{-5}$	0.000692	$1.21604*10^{-5}$	0.000362
BB	0.000692	0.003843	0.000362	0.00231
B	0.0038434	0.013068	0.002317	0.00853
CCC	0.01306	0.158488	0.00853	0.065516
CC	0.15848	0.63818	0.06551	0.28311
C	0.63818	0.99752	0.28311	0.95799
D	0.99752	1.000	0.95799	1.000

Table 6.9: Optimum cut-off probability of defaults of probit regression for 10 rating category.

shows the number of companies in each rating categories and number of defaults in each category. Accordingly, while logistic regression assigns most of the defaults in the worst rating category "D", the probit regression assigns default observations in the categories "CC", "C" and "D".

Ratings	Logistic Regression		Probit Regression	
	validation	training	validation	training
AAA	20	67	25	69
AA	14	83	20	78
A	18	67	14	66
BBB	13	87	19	79
BB	18	73	18	92
B	14	74	18	73
CCC	17	71	21	60
CC	20	66	21(Def=7)	75
C	20 (Def=4)	15	19(Def=17)	59(Def=49)
D	45(Def=45)	198(Def=198)	25(Def=25)	149(Def=149)

Table 6.10: Ratings of companies for 10 rating classes (Def: number of defaults in rating categories).

The optimum cut-off probabilities for the cases 9 and 8 rating categories are given in Appendix.

Accordingly, the results shows that more accurate ratings of companies are obtained for the 10 rating category case. Furthermore, Table 6.11 demonstrates the areas under the ROC curves. It is seen that the area is maximized when 10 rating categories are used for both logistic and probit regression.

		Number of rating categories		
		8	9	10
Logistic	validation	0.9270	0.9835	0.9957
Regression	training	0.9327	0.9875	0.9958
Probit	validation	0.9470	0.9865	0.9899
Regression	training	0.9497	0.9917	0.9994

Table 6.11: The areas under the ROC curve for optimum cut-off probabilities.

CHAPTER 7

CONCLUSION

In this work, we gave an overview about the theoretical aspects of important statistical and nonstatistical methods and their applications in credit scoring. In our application part, we firstly focus on logistic regression which is the most widely used method in studies. By Monte Carlo simulations, we examined logistic regression under the aspect of its bias in parameter estimation by using both different data sets which includes various (%) defaults and different lengthes of variables. Our results show that the logistic regression performs well when the data sets include nearly 30% default cases. Moreover, if the independent variable set is very large and some variables have high effects on dependent variable, the coefficient estimates are much more different from their true values. Secondly, we checked the prediction accuracies of all methods mentioned in this work. In this part, cross-validation simulations on real Turkish credit data were made. The results of analyze show that the logistic regression is the best classification technique for Turkish credit data for small, medium and long sizes.

REFERENCES

[A68] Altman, E., Financial Ratios, Discriminant analysis and the prediction of corporate bankruptcy, The Journal of Finance (September 1968) 589 -609.

[AL76] Altman, E.I., and Loris, B., A financial early warning system for over-the counter broker-dealers, Journal of Finance, 31,4 (September 1976) 1201-1217.

[Ati01] Atiya, A.F., Bankruptcy prediction for credit risk using neural networks: A Survey and New Results, IEEE Transactions on Neural Networks, 12(July 2001) 929-935.

[BLSW96] Back B., Laitinen T., Sere K. and van Wezel M., Choosing bankruptcy predictors using discriminant analysis, logit analysis, and genetic algorithms, Turku Centre for Computer Science Technical Report 40 (September 1996).

[BO04] Balcean S. and Ooghe H., Alternative methodologies in studies on business failure: do they produce better results than the classical statistical methods?, Working Paper: Universiteit Gent, Fakulteit Economie, no:249 (2004).

[B66] Beaver, W., Financial ratios as predictors of failure, Empirical Research in Accounting: Selected Studies, 1966, supplement to vol.5, Journal of Accounting Research (1966) 71-111.

[Bl74] Blum, M. Failing company discriminant analysis, Journal of Accounting Research 1 (1974) 1-25.

[B04] Boik R.J., *Classical Multivariate Analysis*, Lecture Notes: Statistics 537, Department of Mathematical Sciences, Montana State University Bozeman, (2004).

[BFOS84] Breiman L., Frydman H., Olshen R.A., and Stone C.J., *Classification and Regression Trees*, Chapman and Hall, New York, London, (1984).

[CF93] Coats P. and Fant L.. Recognizing financial distress patterns using a neural network toll, Financial Management 22 (1993) 142-155.

[CET02] Crook J.N., Edelman D.B., and Thomas L.C., C*redit Scoring and Its Applications*, SIAM ,(2002).

[DK80] Dambolena, I.G., and Khoury, S.J., Ratio stability and corporate failure, The Journal of Finance, (September 1980) 1017-1026.

[D72] Deakin,E.B., A Discriminant analysis of predictors of business failure, Journal of Accounting Research, 10-1 (Spring 1972) 167-179.

[EHT02] Enleman, B., Hayden, E., and Tasche, D. . Measuring the discriminative power of rating systems (2002), www.defaultrisk.com.

[FAK85] Frydman H., Altman E.I., and Kao D-L., Introducing recursive partioning for financial classification: The Case of financial distress, The Journal of Finance XI-1 (March 1985).

[HM01] Hajdu, O. and Mikls, V., A Hungarian model for predicting financial bankruptcy, society and economy in central and eastern europe: quarterly., Journal of Budapest University of Economic Sciences and Public Administration, XXIII 1-2 (2001) 28-46.

[HMSW04] Härdle W., Müller M., Sperlich S., and Werwatz A., *Nonparametric and Semiparametric Models*, Springer Berlin, 2004.

[HS95] Hassoun M.H., Fundamentals of artificial neural networks. Cambridge, Mass., MIT Press, 1995.

[H94] Haykin S., *Neural Networks : a Comprehensive Foundation*, New York, Macmillan, 1994.

[HKGB99] Herbrich R., Keilbach M., Graepel T., Bollmann-Sdorra P., and Obermayer K., Neural networks in economics, http://www.smartquant.com/references/NeuralNetworks/neural7.pdf, (1999).

[Horrowitz] Horowitz, J.L., *Semiparametric Single-Index Models*, Lecture Notes: Department of Economics, Northwestern University

[H66] Horrigan, J. O.. The Determination of long-term credit standing with financial ratios, Journal of Accounting Research, Emprical Research in Accounting: Selected Studies 4 (1966) 44-62.

[HL00] Hosmer D.W., and Lemeshow Jr.S., *Applied Logistic Regression*, New York, Wiley, 2000.

[JW97] Johnson R.A., and Wichern D.W., *Applied Multivariate Statistical Analysis*, Prentice Hall, New Jersey.

[K98] Kiviluoto, K. Predicting Bankruptcies with the self-organizing map, Neurocomputing 21 (1998) 191-201.

[KGSC00] Kolari J., Glennon D., Shin H., and Caputo M., Predicting large U.S. Commercial bank failures, economic and policy analysis, Working Paper 2000-1 (January 2000).

[L99] Laitinen E.K., Predicting a corporate credit analysts risk estimate by logistic and linear Models, International Review of Financial Analysis, 8:2 (1999) 97121.

[LK99] Laitinen, T. and Kankaanpaa, M. Comparative analysis of failure prediction methods: the finnish case, The European Accounting Review 8:1 (1999) 67-92.

[Lib75] Libby, R., Accounting ratios and the prediction of failure: some behavioral evidence, Journal of Accounting Research, 13:1 (Spring 1975) 150-161.

[MG00] McKEE, T.E. and Greensten M. Predicting bankruptcy using recursive partitioning and a realistically proportioned data set. Journal of Forecasting (2000) 19: pp. 219-230.

[M66] Mears, P. K., Discussion of financial ratios as predictors of failures, Journal of Accounting Research, Emprical Research in Accounting: Selected Studies 4 (1966) 119-122.

[M04] Monsour C., Discrete predictive modeling, Casualty Actuarial Society, Special Interest Seminar on Predictive Modeling, Chicago(2004).

[Mu00] Muller, M., Semiparametric extensions to generalized linear models, Humboldt Universitat zu Berlin. Working Paper(2000).

[MR99] Muller M. and Ronz B. Credit scoring using semiparametric methods, Humboldt Universitaet Berlin in its series Sonderforschungsbereich 373 with number 1999-93 (1999).

[N66] Neter, J., Discussion of financial ratios as predictors of failures, Journal of Accounting Research, Emprical Research in Accounting: Selected Studies 4 (1966) 112-118.

[OS90] Odom, M.D., and Sharda, R., A Neural network model for bankruptcy prediction, Neural Networks, IJCNN International Joint Conference on 17-21, 2 (June 1990) 163-168.

[O80] Ohlson, J.A., Financial ratios and probabilistic prediction of bankruptcy, Journal of Accounting Research, 18 (Spring 1980) 109-131.

[Org70] Orgler, Y.E., Credit scoring model for commercial loans, Journal of Money, Credit and Bank 2: (1970) 435-445.

[PS69] Pogue T.F., and Soldofsky R.M., What is bond rating, Journal of Financial and Quantitative Analysis, 4:2 (June 1969) 201-228.

[PF97] Pompe, P.P.M, Feelders, A. J., Using machine learning, neural networks and statistics to predict corporate bankruptcy, Microcomputers in Civil Engineering, 12 (1997) 267-276.

[Probit] Probit regression, http://www.gseis.ucla.edu/ courses/ ed231a1/notes2/probit1.html

[RG94] Rosenberg,E., and Gleit, A., Quantitaive methods in credit management: a survey, Operations Research, 42:4(1994) 589-613.

[S75] Sinkey, J.F., A Multivariate analysis of the characteristics of problem banks, The Journal of Finance, 30: (March 1975) 21-36.

[SKSt00] Sobehart, J.R., Keenan, S.C., and Stein, M.S., Bencmarking quantitative default risk models: a validation methodology, Moody's Rating Methodology (2000).

[TK92] Tam Y. K. and Kiang Y.M. Managerial applications of neural networks: the case of bank failure predictions, Management Science, 38: (July 1992) 926-947.

[Wil71] Wilcox, J. W.. A Simple theory of financial ratios as predictors of failure, Journal of Accounting Research, 9:2 (1971) 389-395.

[Wil73] Wilcox, J. W., A Prediction of business failure using accounting data, Journal of Accounting Research, Emprical Research in Accounting: Selected Studies 11: (1973) 163-179.

[WBS97] Wong B.K., Bodnovich T.A., and Selvi Y., Neural network applications in business: a review and analysis of the literature (1988-95), Decision Support Systems 19 (1997) 301-320.

[YW99] Yohannes Y., and Webb P., *Classification and Regression Trees*,CART, A User Manual for Identifying Indicators of Vulnerability to and Chronic Food Insecurity, International Food Policy Research Institute, (1999).

CHAPTER 8

APPENDIX

Ratings	Probability of defaults from logistic regression			
	validation sample		training sample	
	lower range	upper range	lower range	upper range
AAA	0.000	$2.4201*10^{-10}$	0.000	$1.3673*10^{-10}$
AA	$2.4201*10^{-10}$	$2.76115*10^{-8}$	$1.3673*10^{-10}$	$1.07211*10^{-8}$
A	$2.76115*10^{-8}$	$2.20445*10^{-5}$	$1.07211*10^{-8}$	$3.62314*10^{-7}$
BBB	$2.20445*10^{-5}$	0.000139	$3.62314*10^{-7}$	$1.15619*10^{-5}$
BB	0.000139	0.007353	$1.15619*10^{-5}$	$8.10605*10^{-5}$
B	0.007353	0.007774	$8.10605*10^{-5}$	0.003895
C	0.007774	0.073298	0.003895	0.02008
D	0.073298038	1.000	0.02008	1.000

Table 8.1: Optimum cut-off probability of defaults of logistic regression for 8 rating category.

Ratings	Probability of defaults from probit regression			
	validation sample		training sample	
	lower range	upper range	lower range	upper range
AAA	0.000	$2.346*10^{-11}$	0.000	$2.346*10^{-11}$
AA	$2.346*10^{-11}$	$5.6445*10^{-7}$	$2.346*10^{-11}$	$2.0745*10^{-7}$
A	$5.6445*10^{-7}$	$4.2026*10^{-5}$	$2.0745*10^{-7}$	$1.2160*10^{-5}$
BBB	$4.2026*10^{-5}$	0.000692	$1.2160*10^{-5}$	0.000362
BB	0.000692	0.00384	0.000362	0.002317
B	0.00384	0.01306	0.002317	0.00853
C	0.01306	0.15848	0.00853	0.06551
D	1.000	0.15848	0.06551	1.000

Table 8.2: Optimum cut-off probability of defaults of probit regression for 8 rating category.

Ratings	Logistic Regression		Probit Regression	
	validation	training	validation	training
AAA	22	66	25	69
AA	17	83	20	78
A	21	67	14	66
BBB	16	87	19	79
BB	19	73	18	92
B	19	74	18	73
C	21	71	21	60
D	65	279	65	283

Table 8.3: Ratings of companies for 8 rating classes (Def: number of defaults in rating categories).

Ratings	Probability of defaults from logistic regression			
	validation sample		training sample	
	lower range	upper range	lower range	upper range
AAA	0.000	$2.4201*10^{-10}$	0.000	$1.367*10^{-10}$
AA	$2.4201*10^{-10}$	$2.76115*10^{-8}$	$1.367*10^{-10}$	$1.072*10^{-8}$
A	$2.76115*10^{-8}$	$1.0348*10^{-6}$	$1.072*10^{-8}$	$3.623*10^{-7}$
BBB	$1.0348*10^{-6}$	$2.33465*10^{-5}$	$3.623*10^{-7}$	$1.15619*10^{-5}$
BB	$2.33465*10^{-5}$	0.000279	$1.15619*10^{-5}$	$8.10605*10^{-5}$
B	0.000279	0.00777	$8.10605*10^{-5}$	0.00389
CC	0.00777	0.07329	0.00389	0.02008
C	0.07329	0.79767	0.02008	0.1882
D	0.79767	1.000	0.1882	1.000

Table 8.4: Optimum cut-off probability of defaults of logistic regression for 9 rating category.

Ratings	Probability of defaults from probit regression			
	validation sample		training sample	
	lower range	upper range	lower range	upper range
AAA	0.000	$2.346*10^{-11}$	0.000	$2.346*10^{-11}$
AA	$2.346*10^{-11}$	$5.644*10^{-7}$	$2.346*10^{-11}$	$2.074*10^{-7}$
A	$5.644*10^{-7}$	$4.202*10^{-5}$	$2.074*10^{-7}$	$1.216*10^{-5}$
BBB	$4.202*10^{-5}$	0.000692	$1.216*10^{-5}$	0.00036
BB	0.000692	0.00384	0.00036	0.00231
B	0.00384	0.01306	0.00231	0.00853
CC	0.01306	0.15848	0.00853	0.06551
C	0.15848	0.63818	0.06551	0.28311
D	0.63818	1.000	0.28311	1.000

Table 8.5: Optimum cut-off probability of defaults of probit regression for 9 rating category.

Ratings	Logistic Regression		Probit Regression	
	validation	training	validation	training
AAA	22	66	25	69
AA	17	83	20	78
A	21	67	14	66
BBB	16	87	19	79
BB	19	73	18	92
B	19	74	18	73
CC	21	71	21	60
C	22(Def=6)	66	21(Def=7)	75
D	43(Def=43)	213(Def=198)	44(Def=42)	208(Def=198)

Table 8.6: Ratings of companies for 9 rating classes (Def: number of defaults in rating categories).