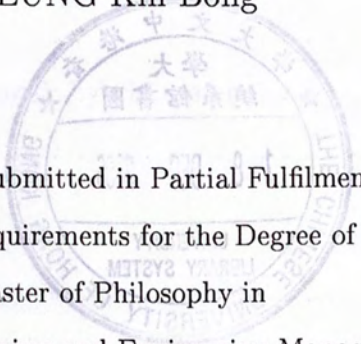


Modeling Financial Risk: From Uni- to Bi-directional

YEUNG Kin Bong



A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy in
Systems Engineering and Engineering Management

© The Chinese University of Hong Kong

August 2005

The Chinese University of Hong Kong holds the copyright of this thesis. Any person(s) intending to use a part or whole of the materials in the thesis in a proposed publication must seek copyright release from the Dean of the Graduate School.

Modeling Financial Risk: From Theory to Practice



© The Chinese University of Hong Kong

August 2006

The Chinese University of Hong Kong holds the copyright of this book. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the Chinese University of Hong Kong.

Contents

1	Introduction	1
1.1	Credit risk modeling	3
1.2	Uniqueness of bi-directional: hybrid system	4
1.3	Scope of the study	5
2	Literature Review	6
2.1	Statistical / Empirical approach	6
2.2	Structural approach	8
3	Background	10
3.1	Merton structural default model	10
3.2	Cross-sectional regression analysis (CRA)	15
3.3	Neural network learning (NN)	16
3.3.1	Single-layer network	17
3.3.2	Multi-layer perceptron (MLP)	20
3.3.3	Back-propagation network	22
3.3.4	Supervised, unsupervised and combine unsupervised-supervised learning	23
3.4	Weaknesses of uni-directional modeling	23

4	Methodology	26
4.1	Bi-directional modeling	26
4.2	Asset price estimation	31
4.3	Quantifying accounting data noise	33
5	Proposed Model	37
5.1	Core of the model	37
5.2	Feature selection	41
5.3	Bi-directional default neural system	44
6	Implementations	49
6.1	Data preparation	50
6.2	Experiment	51
6.3	Empirical results	61
6.3.1	Predicted spreads from the uni-directional models	61
6.3.2	Predicted spreads from the proposed bi-directional model	63
6.3.3	Performance comparison	64
7	Conclusions	67
	Bibliography	69

List of Tables

3.1	Merton's options-theoretic view of firms	12
3.2	Learning algorithm classification	24
5.1	Predictors for yield spread residual	42
6.1	Canadian firms and corresponding industries in the sample	51
6.2	Summary statistics on the bonds and issuers in the sample	51
6.3	Average risk-free rate in the sample period	52
6.4	Estimation of parameters	53
6.5	Results of network training obtained by using preprocessed asset price time-series and two-layer feed-forward network with 20 nodes in hidden layer (6-20-2 architecture).	57
6.6	Summary of training algorithm performance	59
6.7	Prediction errors of the chaotic part of the bi-directional modeling on training (in-sample) and testing (out-of-sample) sets.	59
6.8	Prediction percentage error of the Merton model: very large percentage errors and dispersion	61
6.9	Prediction percentage error of the bi-directional modeling: average percentage errors significantly improved	64

List of Figures

3.1	An illustration of the Merton model	12
3.2	Single-layer network - 2 classes	19
3.3	Single-layer network - c classes	19
3.4	Architecture of neural network	21
4.1	True system vs. artificial system	27
4.2	Merton model as an artificial "system" - System 1	28
4.3	First neural network module (System 2) - solving unobserved asset value by learning the relationship among equity prices and asset prices	29
4.4	Second neural network module (System 3) - solving credit risk underestimation by predicting the <i>risk residual</i> (i.e. the difference between the observed and predicted spreads)	30
4.5	The parameters θ of a pre-defined parametric model for the asset price distribution of \mathbf{x} are determined by the outputs of a neural network	33
4.6	Accounting Distance	36
5.1	Core of "True Nature"	38

5.2	Relationships of AD on yield spreads in the sample set of Canadian firms and bonds: the level of yield spreads versus AD in panel A shows the nonlinear nature (a smoothed trendline superimposed on data) of the relationship and how predictive the AD is (in a univariate sense).	43
5.3	Relationship between other variables and yield spreads (1): Yield spread inclines gradually and goes steeper at the end while leverage and equity volatility increase. The level of yield spreads versus leverage and volatility shows the highly nonlinear nature (<i>polynomial</i> trendlines fitted in data with the order of 3 and 2 respectively) of the relationship.	47
5.4	Relationship between other variables and yield spreads (2): For duration, yield spread reaches its highest point in the middle and then declines and stays flat after. Relationship between amount outstanding and yield spread is almost like normal distributed. Nonlinear nature is observed (<i>polynomial</i> trendlines fitted in data with the order of 5 and 4 respectively).	48
6.1	A sliding window extracts a sample set of input-output pairs from an asset price time-series.	55
6.2	Predicted temporal asset distribution (mean and std. dev.) for Hudson's Bay in 1990-2004 by neural network	56
6.3	An instance of yield spreads generated from the Merton equation for Sears Canada	58
6.4	The prediction results of risk residual on testing set of bi-directional modeling	60
6.5	Performance of the Merton structural model: extreme under- and over-estimation	62

6.6	Performance of the OLS regression (out-of-sample): extreme over-estimation	63
6.7	Performance of the bi-directional modeling: ability to generate high yield spreads and match the observed market.	64
6.8	Accuracy for the six tested models in the out-of-sample set: the bars depict the performance of each of the default models tested (less RMSE represents better prediction). Note the performance gap between NN, OLS (statistical) and Merton model (structural). Also note the performance gap between Merton model and our proposed bi-directional model. These gaps represent the gain in model accuracy from incorporating additional financial information for prediction and learning process.	65

Abstract

Traditional structural approach to model financial risk is basically based on modeling the underlying dynamics or relationships among related factors. It may ignore the noise of the data itself which would contribute significantly to the risk assessed. We propose a hybrid financial risk model using a combination of structural model and neural networks, which jumps from the framework of uni- to bi-directional approach. At the same time modeling the risk structurally and learning the risk from data statistically, it would greatly improve the accuracy of the credit risk prediction. Its performance is demonstrated by applying the framework on the Merton (1974) structural default model. Through the study of the error curves - the behavior of the root mean square error (RMSE) on the testing set, the improved performance is observed.

We apply neural network learning to two crucial problems of the Merton model: the unobserved data and severe risk underestimation. By parametric statistical estimation and considering the risk induced by data noise such as imperfect accounting reports, the uni-directional Merton model is improved by the proposed bi-directional hybrid neural system. An application to seven Canadian firms and corresponding thirteen bonds in the real market is presented. The empirical results show that the new proposed method outperforms existing financial models

撮要

傳統模擬金融風險的結構方法是根據有關風險因素的根本動態及關係而作出預測。這樣很多時忽略了數據本身的干擾而此也正正是風險的一重要部分。因此，我們提出了一應用結構方法及神經網絡的混合模型。有別於傳統單向的模型，這設計能夠在同一時間模擬結構風險以及學習數據中存在的風險，從而大大提升預測的準確度。為了試驗混合模型的表現，我們利用了Merton (1974) 結構破產模型為基礎作實驗研究。透過研究實驗中的誤差曲線 (RMSE)，我們便可以觀察到模型改善後的表現。

我們應用了神經網絡在Merton模型中的兩個核心問題上：無法觀察到的金融數據及嚴重的風險低估。研究發現在雙向的混合系統架構下，應用參數統計預測和考慮不完整會計報告風險可以有效改善單向模型的缺點。最後我們使用了七所在市場活躍的加拿大公司及對應的債券作試驗，結果顯示新提出方案的表現比現存的金融模型優勝。

Chapter 1

Introduction

There are two major approaches to financial risk modeling: structural and statistical. Traditionally, the structural approach is based on modeling the underlying dynamics or relationships among related factors, say interest rate and asset value, to derive the risk. Using the terminology of Cherkassky (1993), that is actually a top-down (“model-driven”) approach, which believes that mathematics can be used to represent or model any financial behavior perfectly. Unfortunately, researchers found that many market dynamics are too complicated or chaotic to model¹. Therefore, many derived models are found not quite consistent to the observed behaviors. This approach seemed to ignore the noises contained in the financial data that should be part of the risk to be assessed².

The statistical or empirical approach is that, instead of modeling the relationships directly, such relationships are learned from the historical data³. That’s why we also

¹Other than mathematical models, some types of nonlinear system modeling and learning methodologies like neural networks, multiple models, and chaotic pattern detection are used in financial applications instead. See Ljung (1999) and Jang et al. (1997) for details.

²for example, Duffie and Lando (2001) consider the imperfect of financial data as a risk factor and successfully improve the accuracy of risk prediction.

³See Beaver (1966), Altman (1968) and Ohlson (1980).

call it a bottom-up (“data-driven”) approach. Preprocessing and feature extraction in a data pool will be the critical step to support the relationship learning. However, if only statistical approach is used to learn the patterns from data, from the financial analysis point of view, it may rely on assumptions which may be too weak.

For example, in the case of credit risk modeling, statistical approach can only rely on sparse and noisy default data to generalize models, it is indeed a big challenge⁴. Without the backing of fundamental theoretic assumptions such as market efficiency and the boundary of default on debt obligations, modeling framework is very loose and *not convincing* at all.

We call the above stream of research, which uses only one approach, either pure structural or statistical, to dominate the whole modeling process, the *uni-directional modeling*. Although this stream is still active and popular in academia, disappointing accuracy from a variety of extensions and improvements drive us to find another stream out.

Motivated by the shortcomings of uni-directional modeling, our research is to propose a new approach to financial risk modeling with a hybrid process - the *bi-directional modeling*. The power of the new proposal is that financial risk is modeled structurally (top-down) and *at the same time*, relationship of the risk from data is learnt statistically (bottom-up). As a result, the shortcomings of each model can be compensated by each other and hence, the total risk can be estimated by such combination. The rigidity of structural modeling is improved by the flexibility from statistical modeling, which provides higher degree of freedom.

As a key contribution of this research and also a better illustration of the proposed

⁴Referred to Xu (2002), the key challenge of statistical learning is that learning is made on a finite size of samples but we want it to be applied to all or as many as possible new coming samples in the future.

framework, Merton (1974) structural default model will be implemented based on the data set of seven Canadian firms and corresponding thirteen bonds in the real market to show how it can improve the accuracy (by observing the RMSE on the testing set) of credit risk estimation and outperform the uni-directional modelings.

1.1 Credit risk modeling

Defined by Moody's Investors Service (2000), credit risk, or default risk, is defined as the potential that a borrower or counterparty will fail to meet its obligations in accordance with agreed terms. The failures may include a missed or delayed payment of interest or principal, a filing for bankruptcy or a distressed exchange. For financial engineers, it is always important to estimate the probability of default / bankruptcy (PD) and the expected loss of the counterparty (either individual or corporate) accurately. So that, investors or banks can manage the corresponding credit risk and make better financial decisions. A number of the world's largest financial institutions are still researching and developing sophisticated models in an attempt to aid institutions better quantifying, monitoring and managing the credit risk.

Statistical approaches based on historical data, which have the longest history (Beaver, 1966) and are the most frequently found in the literature of credit risk modeling. However, the completeness and quality of data affect the accuracy and success of the analysis significantly. Over the last decades, structural approach emerged after the seminal work of Black and Scholes (1973), and Merton (1974) created an enormous theoretical literature on credit risk modeling. Many researchers tried to find the main source of credit risk by studying the market information and corporate bond

yield spread so as to estimate the probability of default accurately. However, such structural models depend heavily on their assumptions to capture the true nature of the underlying dynamics and the accuracy of the model variables estimation. Therefore, their performance was not really promising as expected. Many empirical testing of Merton model found that it could not generate sufficiently high yield spreads to match those observed in the market, for example, the early study by Jones, Mason and Rosenfeld (1984) and Eom, Helwege and Huang (2001). That means those models severely underestimated the probability of default and the associated risk.

1.2 Uniqueness of bi-directional: hybrid system

In this research, we try to model credit risk with the bi-directional approach - which is a new methodology that includes both bottom-up (data-driven or statistical) and top-down (model-driven or structural) approaches. That is to develop a *hybrid system* that incorporates both structural model based on Merton model and statistical model learned from financial statement and corporate bond price. By integrating two dimensions of modeling, it is no longer relied on "data" or "model" but more importantly, it is "task-driven". Cherkassky (1993) indicated that future intelligent systems should be task-driven and their functionality can be enhanced by modular design using hybrid systems approach and multi-strategy learning.

One example of using hybrid system is a short-term default risk model developed by Moody's Corporation (2000). It created a system that merged both a contingent claims model⁵ and a statistical reduced form model by using a non-linear regression approach. This system not only takes the results of Merton model as the inputs, but

⁵Merton models credit risk by treating the financial stress situation as an option (or contingent claim) to price. That is called Merton's options-theoretic view of firms. See more detailed discussion in section 3.1.

also includes (1) credit agency rating, (2) company financial statement, (3) equity market information and (4) macroeconomic variables that reflecting the economy state to improve the accuracy. Though the model has proved to be useful as an early warning system to monitor corporate credit risk and outperformed a variety of models such as linear, contingent claims and logistic regression models in literature, the drawback is that a huge default database is needed. In our research, however, we focus on modeling based on a small-scale database⁶ as default data is somehow too scarce to be collected. Also, a detailed study about the strengths and weaknesses of Merton model is conducted to ensure that the corresponding tailor-made statistical model is the perfect match.

1.3 Scope of the study

In chapter 2, we briefly review both statistical and structural approaches in literature. We introduce three uni-directional models that would be intensively discussed in this study and their weaknesses in chapter 3. We show the strategy and methodology for the accuracy improvement problem in chapter 4. Chapter 5 proposes a bi-directional neural system to improve the Merton model. Finally, we do experiment to real data on seven Canadian firms and show empirical results in chapter 6 and conclude in chapter 7.

⁶Taking the default models developed by Moody's for example, the data set contains about 100,000 firm-year observations. Also, the sample of default events (up to 1400 cases) is included. It is always the best for data-driven models to have such huge and universal database. For more detail, see Moody's (2000).

Chapter 2

Literature Review

There are two main approaches to credit risk modeling: statistical and structural. The statistical approaches are discussed first, starting from the earliest simple linear analysis to the latest and most frequently applied neural networks. Then, the structural approach with profound Merton model and its variants are reviewed. Finally, we present Merton's empirical analysis in literature.

2.1 Statistical / Empirical approach

The earliest pioneers of the empirical approach are Beaver (1966), Altman (1968) and Ohlson (1980). Beaver is the first person to study the prediction of default / bankruptcy using financial statement data. Though the analysis was simple, it opened the empirical approach and led many researchers to this new direction. Altman and Ohlson try to classify healthy and unhealthy firms using linear models. More detailed studies have been conducted about the inputs of financial ratios and even today, that classic set of the ratios are still widely used in more sophisticated models. For the models, the classical multivariate discriminant analysis (MDA) by Altman and the

logistic regression approach (LR) by Ohlson are also widely studied in academia after.

Neural networks (NN) can be treated as a general case of LR. That is, a non-linear logistic regression achieved by a multi-layered network having either threshold or sigmoidal activation functions. Application of NN is broadly ranging from medical to environmental, financial application is one of the most active fields. In early 1990, researches about default risk prediction using NN have already started. One of the first studies was the work done by Odom and Sharda (1990), who use Altman's financial ratios as network inputs and compare its performance with MDA. In that study, NN achieved a Type I and Type II accuracy in a range up to 81.5% and 85.7% respectively. That significantly outperforms MDA. Tam and Kiang (1991, 1992) focused on the problem of bank default prediction. They compared the performance between several statistical methods such as MDA, LR, K-nearest neighbor (KNN), ID3 (a classification algorithm for decision tree), single-layer network and multilayer network. After all, the multilayer network has the best performance among.

Over the decade, we can see many researchers put intensive efforts on applying NN in the problem of default risk prediction and compare it with other models, for example, Salchenberger et al. (1992), Coats and Fant (1993), Kerling and Poddig (1994), Altman et al. (1994), Boritz and Kennedy (1995), Fernandez and Olmeda (1995), Alici (1995), Leshno and Spector (1996), Zhang et al. (1999), Martinelli et al. (1999) and Atiya (2001). The overall accuracy obtained by NN outperforms existing models. From Lee et al. (1996) onwards, hybrid NN models are being considered to be another research stream. They not only used the network for a single problem, but also tested the possibility of combining NN with MDA, ID3, self-organizing maps (SOM) and genetic algorithm (GA).

2.2 Structural approach

The seminal work of Black and Scholes (1973) and Merton structural default model (1974) is one of the most profound and broadly developed methods. An option pricing approach was used to price corporate liabilities so that the corresponding credit risk could be assessed. Merton assumes that the firm's asset value is governed by a geometric Brownian motion and hence the assumption of lognormality can be made. The success of the model is because of its using equity prices as the predictive index, which has never been the case before. Many variants have been further developed right after the novel Merton model, so that it would match the observed behavior in the market more¹.

Since the original Merton model can only deal with zero coupon bonds and constant interest rates, the extended version such as Longstaff and Schwartz (1995) tried to treat a coupon bond as *a portfolio of zero coupon bonds* so that each part can be priced as the original version. Also, the model allows stochastic interest rates that are described by the Vasicek (1977) model. Geske (1977) just solved the problem in different way by treating the coupon as a compound option. Collin-Dufresne and Goldstein (2001) extended the Longstaff and Schwartz model to allow deviation from target leverage ratio of the firm only over short run. Extensive empirical study of the Merton structural model can be found in Jones, Mason, and Rosenfeld (1984)². Moody's KMV Corporation (1993) has successfully put this model into a commercial product. For detailed reference of Merton/KMV approach please find Sundaram,

¹See Black and Cox (1976), Briys and de Varenne (1997), Ho and Singer (1982), Kim, Ramaswamy and Sundaresan (1993), Leland (1994, 1998), Titman and Torous (1989), Duffie and Lando (2001), Huang and Huang (2002)

²They apply the Merton model to a sample of firms with simple capital structures and secondary market bond prices during the 1977-81 period. The empirical implementation is found that the predicted prices from the model are too high by an average of 4.5% (i.e. the yield spreads are underestimated).

Rangarajan (2001).

Recently, many financial products for credit risk modeling have been developed other than the Merton/KMV model. J.P. Morgan's CreditMetrics (1997) is a tool for assessing *portfolio risk* due to changes in debt value caused by changes in obligor credit quality. The mechanism is based on modeling the "rating migration" so that the PD can be estimated, within a given time horizon, which is often taken arbitrarily as one year. Not only by possible default events, CreditMetrics but also include changes in value caused by upgrades and downgrades in credit quality. However, three major limitations come from three critical assumptions: (1) *Same* rating class have the *same* default rate, (2) actual default rate is equal to the historical average default rate and, (3) no market risk is considered (i.e. the interest rates are assumed to evolve in a deterministic fashion only). Credit-VaR of a portfolio is then derived by CIBC in a similar fashion as CreditMetrics for market risk. It is simply the percentile of the distribution corresponding to the desired confidence level.

CreditRisk+ developed by Credit Suisse Financial Products (CSFP) (1997) is based on modeling default for individual bonds, or loans as a Poisson process. The default risk is only defined by default losses. CreditRisk+ does not explicitly model the credit migration risk. Instead, it allows for stochastic default rates which partially account, although not rigorously, for migration risk. Jarrow and Turnbull (1995) develop another structural approach to model default as a point process with the time-varying hazard function for each credit class. Estimation is based on the credit spreads.

Crouhy et al. (2000) and Eom et al. (2004) give a detailed review and comparative analysis of current structural credit risk models³.

³See also Lyden and Saraniti (2000), Wei and Guo (1997), Anderson and Sundaresan (2000), and Ericsson and Reneby (2001).

Chapter 3

Background

In this chapter, we briefly introduce how the mechanism of Merton model goes, especially on the situation of yield spread prediction for risky debts which is the core that we are going to study and improve by bi-directional modeling. After studying the strengths and weaknesses of the model, an industrial practice of yield spread prediction - cross-sectional regression analysis is presented. Finally, the neural network learning is introduced. We can see how the neural network of statistical approach plays the role of improving Merton structural model and learns the lesson from normal practice.

3.1 Merton structural default model

In Merton (1974) model, firms are assumed to have a very simple capital structure. That is, each firm at time t , with asset value V_t , is financed by equity with market price E_t and a zero-coupon risky debt priced at D_t with face value F maturing at

time T . By assuming that V_t is always governed by a geometric Brownian motion

$$dV_t = \mu V_t dt + \sigma_{V_t} V_t dW_t \quad (3.1)$$

where μ and σ_{V_t} are the *drift* and *volatility* respectively, an option valuation approach¹ could be used to price corporate liabilities so as to the corresponding credit risk. The risk-free rate r is assumed to be a constant. A simple illustration is shown in Figure 3.1 and Table 3.1.

The bond value D_t is given as

$$V_t \Phi(-d(V_t, t, \sigma_{V_t})) + F e^{-r(T-t)} \Phi(d(V_t, t, \sigma_{V_t}) - \sigma_{V_t} \sqrt{T-t}) \quad (3.2)$$

where $\Phi(\bullet)$ is the standard normal distribution function and

$$d(V_t, t, \sigma_{V_t}) = \frac{\ln(V_t/F) + (r + \sigma_{V_t}^2/2)(T-t)}{\sigma_{V_t} \sqrt{T-t}} \quad (3.3)$$

and the corresponding probability of default (PD)

$$P_t = P[V_T < F] = \Phi\left(\frac{\ln(F/V_t) + (\mu - \sigma_{V_t}^2/2)(T-t)}{\sigma_{V_t} \sqrt{T-t}}\right) \quad (3.4)$$

can be obtained. Following the formula for the bond value, the (credit) yield spread² can be derived directly as

$$C_t = -\frac{\ln(D_t/F)}{T-t} - r. \quad (3.5)$$

This model provides great insight because it is the first model using forward

¹Equity-holder's payoff at maturity T is $\max\{V_t - F, 0\}$. The residual claim of equity-holder is simple the payoff from holding a long position in a call option on the firm asset value V_t with a strike price F maturing at time T . For the debt-holder, the payoff would be $\min\{V_t, F\}$.

²Credit risk measurement: probability of default(PD) \times loss given default(LGD)

Table 3.1: Merton's options-theoretic view of firms

	Asset Value	Debt-holder's Payoff	Equity-holder's Payoff
$t = 0$	V_0	D	E_0
$t = T$	$V_T > D$	D	$E_T = V_T - D$
$t = T$	$V_T < D$	V_T	$E_T = 0$

Obviously, input variables are needed in order to implement Merton's model. In particular, four critical variables are concerned, as pointed out by KM (1982): (a) initial values of assets, (b) volatility of asset values, (c) "shape" of the distribution of asset values, and (d) face values of obligations requiring servicing. Depending on the "quality" of assets, it is extremely crucial for the implementation as asset values are usually unobservable.

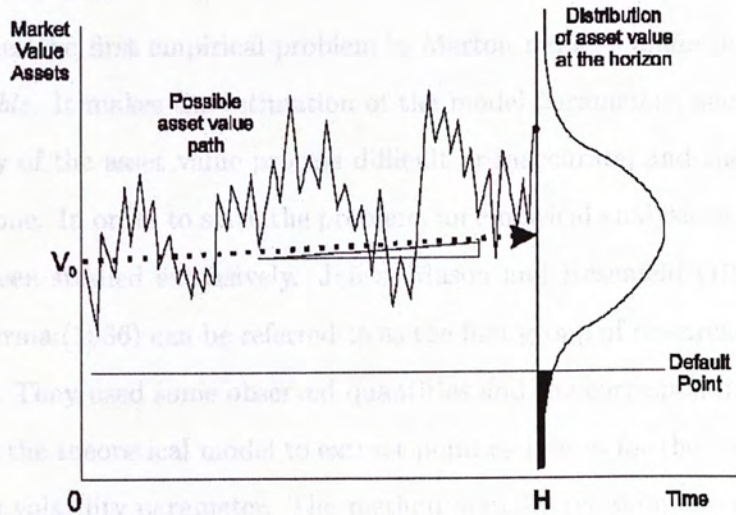


Figure 3.1: An illustration of the Merton model

An iterative method used by KM is another approach to solving the estimation problem. By starting with an initial guess of the parameters, the

looking and going concern approach in default prediction analysis. Although its theory is really breakthrough and theoretically strong at that time, there are many empirical shortcomings during implementation. Many researchers therefore proposed many extended variants based on Merton model, for example, Longstaff and Schwartz (1995), Briys and de Varenne (1997), Madan and Unal (2000) and Collin-Dufresne and Goldstein (2001).

Obviously, input variables are needed in order to implement Merton model empirically. Four critical variables are concerned, as pointed out by KMV: (a) market values of assets, (b) volatility of asset values, (c) "shape" of the distribution of asset values, and (d) face values of obligations requiring servicing. Regarding the "market values" of assets, it is extremely crucial for the implementation as asset values are actually unobservable.

Here comes the first empirical problem in Merton model - *underlying asset value is unobservable*. It makes the estimation of the model parameters, such as, the drift and volatility of the asset value process difficult or inaccurate, and therefore bias or errors are prone. In order to solve the problem, an empirical analysis of the structural model has been studied extensively. Jones, Mason and Rosenfeld (1984) as well as Ronn and Verma (1986) can be referred to as the first group of researchers to conduct such studies. They used some observed quantities and the corresponding restrictions derived from the theoretical model to extract point estimates for the underlying asset value and its volatility parameter. The method actually relies on the two equations: one relating asset value to equity value and the other relating asset volatility and equity volatility.

An iterated method used by KMV is another approach solving the parametric estimation problem. By starting with an initial guess of the asset volatility, the

invested asset values corresponding to the observed time series of equity prices are then obtained through the equity pricing equation. That is, the values for the firm's assets and volatility are implied from equity prices. Vassalou and Xing (2003) used the KMV method to obtain a default likelihood indicator.

Under the framework of statistical approach, maximum likelihood estimation (MLE) is one of the methodologies to estimate parameters, which is based on the Bayesian learning. Duan (1994, 2000) proposes a likelihood function based on the observed equity values derived by employing the transformed data principle in conjunction with the equity pricing equation. Ericsson and Reneby (2001) use this method in corporate bond pricing model.

The second problem in the Merton model is that *credit risk is severely underestimated*. That is, the model cannot consistently represent the actual equity and bond price dynamics, even by substituting any well-estimated asset price and volatility. It results in the estimated credit spreads that differ greatly from those observed empirically. As a result, credit risk is being under-estimated seriously. The inconsistency and insufficiency of the Merton model indicate the shortcoming of uni-directional modeling: It ignores the risk induced by data noise such as imperfect accounting reports. This consideration is crucial for the proposed bi-directional modeling and, it also makes perfect sense from the viewpoint of operational risk: Issues like fraud, opaque accounting practices and incomplete data source are *risky*.

Another minor empirical problem is that Merton model assumes a zero-coupon debt, however, most corporations have much more complex liability structure. Also, the amount of debt determination for Merton's framework is quite an arbitrary among many implementations. The proportion of short- and long-term liabilities is usually treated as one of the variables to be estimated.

3.2 Cross-sectional regression analysis (CRA)

In normal industrial practice, yield spread is predicted using *cross-sectional regression analysis*. Researchers such as Sengupta (1998) and Yu (2005) have used similar approaches for yield curve estimations³. In general, companies will first select input variables that are statistically significant to account for the major portion of the cross-sectional variation in the yield spread (YS), for example, leverage (LEV), equity volatility (VOL), and bond maturity (MAT). Then, for each time point i in the sample period, companies use the simplest *ordinary least squares (OLS)* to estimate the following regression

$$YS_i = \beta_0 + \beta_1 LEV + \beta_2 VOL + \beta_3 MAT + \varepsilon_i \quad (3.6)$$

Because the regression can only capture linear function, *piecewise linear function* separating the whole set into subsets of different maturities needs to be introduced in order to model a nonlinear term structure of yield spreads. Then, by separating different interested groups, say high and low equity volatility, effect of particular variables on the level of yield spreads is studied⁴.

Piecewise function is one of the approaches to handle nonlinearity in yield spread term structure. We however will see that neural networks, which actually derived from simple linear regression, do act as a much more general framework in function modeling including nonlinearity in the next section.

³See also Lang and Lundholm (1993), and King and Khang (2002).

⁴For the detailed discussion of the piecewise linear function construction, see Yu (2005)

3.3 Neural network learning (NN)

The field of learning theory emerges while the computational power and data storage volume keeps increasing. This stream of theory in fact is under the root of statistical / empirical approach. It concerns how computational methods automatically improve with experience. Normally, it applied to large-scale problems with high complexity. Many successful applications have been developed ranging from pattern recognition, to dimension reduction, to function approximation. Computational finance is one of them. Time-series prediction via AR, ARMA models and neural networks are the cases that often encountered in literature. The key algorithms and methodologies that form the core of learning theory, as summarized by Mitchell (1997), are as follows: (a) decision tree learning, (b) artificial neural networks, (c) evaluating hypotheses, (d) Bayesian learning, (e) computational learning theory, (f) instance-based learning, (g) genetic algorithms, (h) learning sets of rules, (i) analytical learning, (j) inductive-analytical learning and (k) reinforcement learning.

Neural network is a *non-linear* regression (nested) model based on a combination of logistic regression. Its design has been inspired by the biological learning systems built of very complex webs of interconnected neurons in human brain. In this computer analogy, network is built out of a densely interconnected set of simple units, where each unit takes a number of real-valued inputs (possibly the outputs of other units) and produces a single real-valued output (which may become the input to many other units). It is a so-called *universal approximator*, which provides a robust approach to approximating real-valued, discrete-valued and vector-valued target functions, say, the relationship between probability of default and Altman's financial ratios as mentioned in previous chapter. Usually, predetermination of the relationship between inputs and outputs with the exact functional form is not necessary.

Advantages for neural network learning in finance are:

1. non-linear relationships can be easily captured (normally financial data are in higher order non-linear relationships),
2. it learn in real-time and adapt to changes,
3. it makes reasonable decisions based on incomplete information (normally financial information is never complete), and
4. it is shown to outperform existing traditional financial models in literature.

Neural network is normally designed as proposed by Bishop (1995): a two-layer feed-forward network, because many literatures in the past have proved that any continuous functional mapping can be represented to arbitrary accuracy once, sigmoidal hidden units are used and sufficiently large number of hidden units are provided.

Referring to Smolensky (1996), NN would also be perfect to estimate model's parameters (for example, the asset value, the drift and the diffusion coefficient in Merton model). Other than the architecture of the network, the cost / error functions must be considered. As a matter of fact, different error function settings suit for different natures of problems. For the objective function to probability modeling, Bishop and Atiya (2001) suggests that neural network with cross-entropy error function can indeed achieve the estimation of probability of default and asset price distribution.

To begin with the detailed discussion of neural network, we should start with the most fundamental element - *single-layer network* first.

3.3.1 Single-layer network

Our earliest pioneers of empirical research in credit risk, Altman (1968) and Ohlson (1980), have come across with the methods of multivariate discriminant analysis

(MDA) and the logistic regression approach (LR). They are in fact the very early stage of more complex multi-layer networks.

In the early 60s, Altman of course has chosen the simplest discriminant function consisting of just a linear combination of the input variables to determine two classes: healthy and unhealthy firms. The outcome of such *two-class classification problem* can be represented in terms of a discriminant function y which takes the value greater than 0 if the vector x is classified as C_1 , and the value less than 0 if it is classified as C_2 . In general, the mapping is modeled in terms of mathematical function y which contains a number of adjustable parameters, whose values are determined *with the help of a data set of examples*. The function can be written as

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 \quad (3.7)$$

where \mathbf{w} is denoted as the *weight vector* and the parameter w_0 as the *bias / threshold*. If we represent this simple multivariate linear discriminant function as a diagram of neural network in figure 3.2, we can see that each component in the network is referring to a variable. The bias is simply considered as a weight parameter with an extra input x_0 which is always set to positive 1.

We can image if the network is extended to the case of several classes, the network diagram will become more complex as in figure 3.3.

By considering not just a simple linear function of all the input variables, as what Ohlson (1980) has done in his research, linear discriminant function can be generalized by using a non-linear function $g(\bullet)$

$$y(\mathbf{x}) = g(\mathbf{w}^T \mathbf{x} + w_0). \quad (3.8)$$

In order to bias the linear decision boundary in a classification problem, we may add the *class-conditional densities* $p(x|G_k)$ and assume Gaussian. By using Bayes' theorem, we will find that the discriminant function g can be written as

$$F(G_k|x) = g(\mathbf{w}^T \mathbf{x} + w_0) = \frac{1}{1 + \exp(-a)} \quad (3.1)$$

The function $g(a)$ is called the *sigmoid* or *logit* activation function, which allows the outputs of the discriminant to be interpreted as probabilities. It indeed curbers the possibilities of a discriminant function. In section 4.2, we will see how fundamental is the network output being interpreted as probabilities in asset price trading.

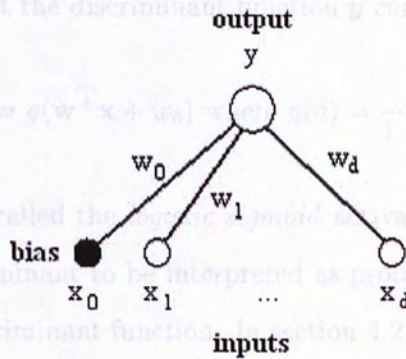
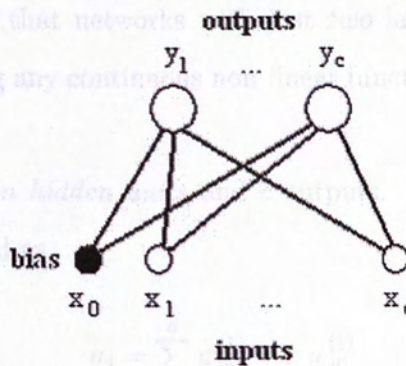


Figure 3.2: Single-layer network - 2 classes

3.3.2 Multi-layer perceptron (MLP)

Due to the limitation that single-layer networks can only solve 'linearly separable problems', networks with several layers are considered. We generally call the multi-layer networks having sigmoidal activation functions *multi-layer perceptrons (MLP)*. It is surprisingly proved that networks with two layers of weights are already capable of approximating any continuous non-linear function. An example of MLP is shown in figure 3.4.

There are d inputs, m hidden units and c outputs. Firstly, the j^{th} hidden unit output can be formulated as



⁴Once the first layer of processing units are designed and fixed to address that network, single-layer networks can in principle solve all linearly separable problems. In fact, this is not true.

Figure 3.3: Single-layer network - c classes

In order to loose the linear decision boundary in a classification problem, we consider the *class-conditional densities* $p(\mathbf{x}|C_k)$ and assume Gaussian. By using Bayes' theorem, we will find that the discriminant function y can be written as

$$P(C_k|x) = g(\mathbf{w}^T \mathbf{x} + w_0) \text{ where } g(a) = \frac{1}{1 + \exp(-a)}. \quad (3.9)$$

The function $g(a)$ is called the *logistic sigmoid* activation function, which allows the outputs of the discriminant to be interpreted as probabilities. It indeed furthers the possibilities of a discriminant function. In section 4.2, we will see how important is the network output being interpreted as probabilities in asset price modeling.

3.3.2 Multi-layer perceptron (MLP)

Due to the limitation that single-layer networks can only solve linearly separable problems⁵, networks with several layers are considered. We generally call the multi-layer networks having sigmoidal activation functions *multi-layer perceptrons (MLP)*. It is surprisingly proved that networks with just *two* layers of weights are already capable of approximating any continuous non-linear function. An example of MLP is shown in figure 3.4.

There are d inputs, m *hidden units* and c outputs. Firstly, the j^{th} hidden unit output can be formulated as

$$a_j = \sum_{i=1}^d w_{ji}^{(1)} x_i + w_{j0}^{(1)} \quad (3.10)$$

⁵Once the first layer of processing units are designed and fixed in advance (non-adaptive), single-layer networks can in fact solve a particular linearly separable problem, but not in general.

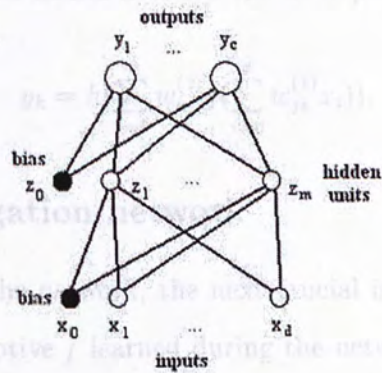


Figure 3.4: Architecture of neural network

or simply include the bias with extra input x_0 always set to 1 as

$$a_j = \sum_{i=0}^d w_{ji}^{(1)} x_i \quad (3.11)$$

where $w_{ji}^{(1)}$ corresponds to a weight from input i to hidden unit j in the *first* layer. Then, a logistic sigmoid activation function $g(\bullet)$ is used for the activation of hidden units as

$$z_j = g(a_j). \quad (3.12)$$

Finally, the output unit k of the network are obtained by a linear combination of all the outputs of the hidden units as

$$b_k = \sum_{j=0}^m w_{kj}^{(2)} z_j. \quad (3.13)$$

We can further add one non-linear activation function for the output units, say h .

After all, the function of the neural network can be depicted mathematically as

$$y_k = h\left(\sum_{j=0}^m w_{kj}^{(2)} g\left(\sum_{i=0}^d w_{ji}^{(1)} x_i\right)\right). \quad (3.14)$$

3.3.3 Back-propagation network

After the formulation of the network, the next crucial issue is that how the weight parameters \mathbf{w} can be adaptive / learned during the network training process. That is, how an error function being minimized with respect to the weights in the network. The most popular and powerful algorithm is called *error back-propagation*.

For simplicity, we first focus on the particular pattern n in the training set and to find the derivative of the error E^n with respect to weight w_{ji} . In the first layer of the network, since E^n depends on the weight w_{ji} only via a_j to hidden unit j , we can write the following partial derivatives

$$\frac{\partial E^n}{\partial w_{ji}} = \frac{\partial E^n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} \quad (3.15)$$

where we denote the *error* as

$$\delta_j = \frac{\partial E^n}{\partial a_j}. \quad (3.16)$$

In the second layer, we can similarly evaluate the error of the output unit k as

$$\delta_k = \frac{\partial E^n}{\partial b_k} \quad (3.17)$$

and the hidden units as

$$\delta_j = \frac{\partial E^n}{\partial a_j} = \sum_k \frac{\partial E^n}{\partial b_k} \frac{\partial b_k}{\partial a_j}. \quad (3.18)$$

Therefore, by combining the above equations, the general back-propagation formula

for propagating the error backwards from all k output units to a particular hidden unit j is

$$\delta_j = g'(a_j) \sum_k w_{kj} \delta_k. \quad (3.19)$$

3.3.4 Supervised, unsupervised and combine unsupervised-supervised learning

Normally learning algorithm can be categorized into three groups: supervised learning, unsupervised learning and combine unsupervised-supervised learning (combine learning). Neural network learning actually can be classified as either supervised or combine learning. Within these two main categories, there are also several sub-categories under network learning while back-propagation network and radial basis function (RBF) are the most frequently used. The detailed classification of NN and other learning, feature selection algorithm is presented as table 3.2 below⁶.

3.4 Weaknesses of uni-directional modeling

Throughout this chapter, we have introduced three modeling methods for default risk prediction that can be also called uni-directional modeling. Merton model is purely structural and model-driven, and CRA and NN are statistical and heavily data-driven. We can observe the shortcomings as follows.

For the Merton structural approach, the weakness mainly comes from how "closely" its assumptions and structure can capture the true world dynamics as well as the accuracy of the estimated parameters in the model. Especially, the Merton model relies

⁶For the detailed discussion of each of the learning algorithms, please refer to Bishop (1995) and Mitchell (1997).

Table 3.2: Learning algorithm classification

	Supervised Learning	Combine Unsupervised-Supervised Learning	Unsupervised Learning (Clustering)
Neural Network Algorithms	Back-Propagation (MLP), Hypersphere, Classifier, Perceptron	Radial Basis Function (RBF), Incremental RBF, Learning Vector Quantizer (LVQ), Nearest Cluster Classifier, Fuzzy ARTMap Classifier	
Statistical and Machine Learning Algorithms	Gaussian Linear, Discriminant, Gaussian Quadratic, K-Nearest Neighbor (KNN), Binary Decision Tree, Parzen Window, Histogram, Naive Bayes, Support Vector Machine (SVM)	Gaussian Mixture Classifier: Diagonal/Full Covariance, Tied/Per-Class Centers	K-Means Clustering, EM Clustering, Leader Clustering, Random Clustering
Feature Selection Algorithms	Linear Discriminant (LDA), Forward/Backward Search		Principal Components (PCA)

heavily on theories about market efficiency. That is assumptions about the comprehensiveness of the information contained in market data when used within the model structure. However, knowledge of market information *alone* in fact does not inform an investor directly as to a borrower's creditworthiness. Some cases like liquidity problems, and information reflection from market data⁷ have been simply ignored.

For the statistical approach like CRA and NN, since this method is heavily data-driven, the weakness mainly comes from the *fitness of data*. That is actually a tradeoff of the generalization error in terms of the model size. If the model is too simple, it loses its power. Though if the model size is increased the generalization error decreases because a larger model has less bias and fits the data better, at some points the model becomes too large - *overfitting* occurs. In that case, even the error on the training set is driven to a very small value, but when new data is presented to the model the

⁷For detailed discussion, see Sobehart and Keenan (1999).

error is very large. The reason is that the model is fitting the data noise also. It is nearly inevitable when using historical financial data⁸.

As a summary, uni-directional weaknesses are in two fold. For the pure structural models, they have ignored the whole picture of the true nature. That is actually a combination of rational and irrational. Though, on the surface, corporate are behaving as in Merton's eyes, many irrational, chaotic or hidden complicated behaviors are indeed happening below. For the pure statistical models, they have lost the initial orientation of the problem. Obviously, in some statistical literature, experiments being carried out may merely let the data speak for themselves (i.e. highly unsupervised). The problem formulation is sometimes without sitting on any theoretical ground. Optimization of fitted model size should be taken much care.

In order to have an optimal modeling, we think, it is always the best choice to let structural modeling plays the rational part and statistical modeling plays the irrational. Therefore, a hybrid system is proposed.

4.) Bi-directional modeling

⁸Based on the accounting principles, reported financial statements are just disclosed in a mandatory or voluntary manner and *not* necessarily reflecting the complete financial picture of the firm. See recent cases of accounting scandals, such as Enron, Authur Andersen, Worldcom, Adelphia, Global Crossing, Tyco, and Xerox for references.

Chapter 4

Methodology

After many studies conducted in literature, we do understand the strengths and weaknesses of both Merton model and NN. The key issue in this research is how to match their rigidity and flexibility perfectly. In this chapter, we first describe how the bi-directional modeling merges both the top-down and bottom-up models based on their characteristics. Then, two key methodologies: asset price estimation and data noise quantification are presented.

4.1 Bi-directional modeling

To better illustrate the bi-directional modeling, we first imagine any kind of modeling as a "system" which is characterized by several variables. Basically, variables are categorized into two types: (1) independent variables / predictor variables / input variables and (2) dependent variables / responses / output variables, depending on the field of study. Referred to Friedman (1994), the goal of any modeling or system M , undoubtedly, is to establish a relationship between the inputs and the outputs observed in the true world X , so as to determine / predict / estimate values for all

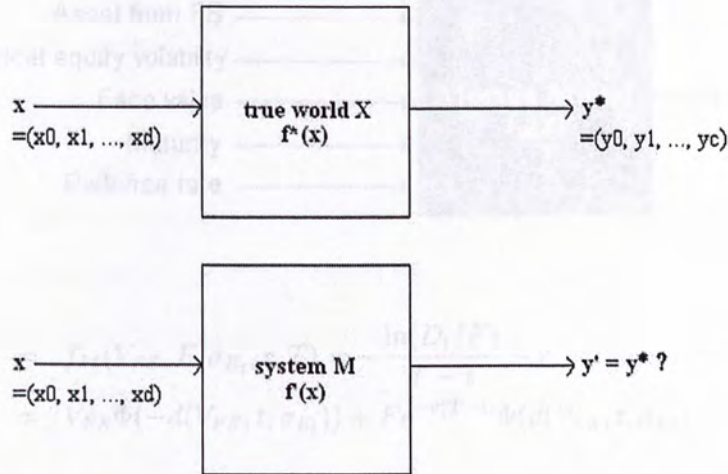


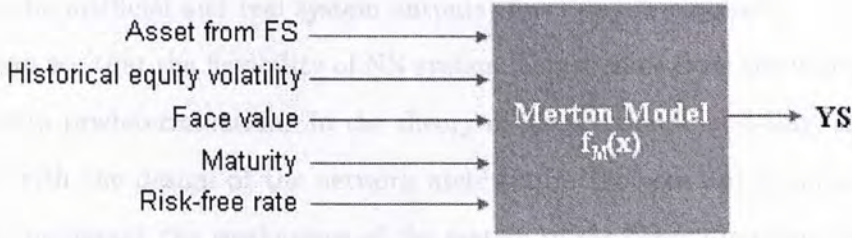
Figure 4.1: True system vs. artificial system

the new coming output variables given only the values of the input variables.

System is represented by a square box and, the input is represented as an arrow on the left side of the box and output as an arrow on the right. Moreover, the inputs can be categorized into two sets, either observed (measured) or unobserved. We can imagine if there is a true system $f^*(\mathbf{x})$ governing the true nature, our learning / artificial system is trying to find a function $f'(\mathbf{x})$ as close as possible to the true function $f^*(\mathbf{x})$. In this research, we are interested in the following true system

$$YS^* = f^*(V_t, F, \sigma_{V_t}, r, T \dots) \tag{4.1}$$

where YS^* denotes the observed corporate yield spread. We believe that the input vector \mathbf{x} should include all the typical variables like those in Merton model but, *should be more*.



$$YS = f_M(V_{FS}, F, \sigma_{E_t}, r, T) = -\frac{\ln(D_t/F)}{T-t} - r \quad (4.2)$$

$$\text{where } D_t = V_{FS}\Phi(-d(V_{FS}, t, \sigma_{E_t})) + Fe^{-r(T-t)}\Phi(d(V_{FS}, t, \sigma_{E_t}) - \sigma_{E_t}\sqrt{T-t})$$

Figure 4.2: Merton model as an artificial "system" - System 1

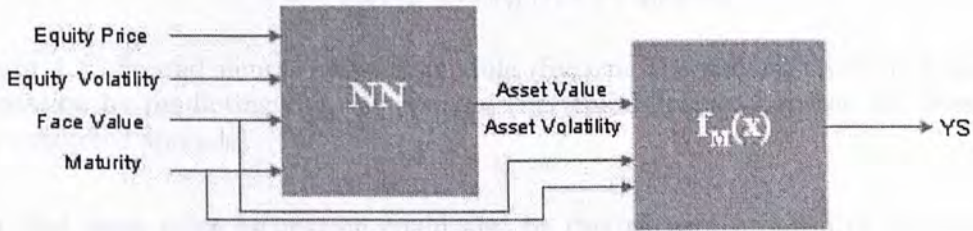
First, the artificial system being studied is the Merton model. The inputs have the asset value V , the face value F , the asset volatility σ_V , the risk-free rate r and maturity T and the output is the yield spread one-year-ahead. The goal of course is to predict the probability of default of the corporate from the knowledge of the input variables without having to actually wait for a year. It is always important for financial institutions to predict credit quality before any loan decision.

Figure 4.2 shows a diagrammatic representation of Merton model. Two input variables (asset value, asset volatility) are actually unobserved and so replaced by estimations (asset value from financial statement (FS), historical equity volatility). Because this system is structural, the relationship f_M between the values of inputs and outputs are determined before prediction.

NN is another example of a system. The only difference is that, as it is statistical, the determination of the relationship beforehand is not necessary. The goal of NN is then to learn a useful approximation to that function by example. Therefore, a "training" sample and a learning process in response to the error (i.e. differences

between the artificial and real system outputs) function are required.

We can see that the flexibility of NN system mainly come from the unnecessary of relationship predetermination. In the theory of bi-directional modeling, as a result, we play with the design of the network architecture (bottom-up) in an attempt to solve or counteract the weaknesses of the system in the other direction (top-down), say, the Merton Model in our case. Two of its critical empirical problems are the *unobserved input variables (asset value and asset volatility)* and the *under-estimation of credit spreads*. Hence, two neural network modules are assigned to each of the problems.

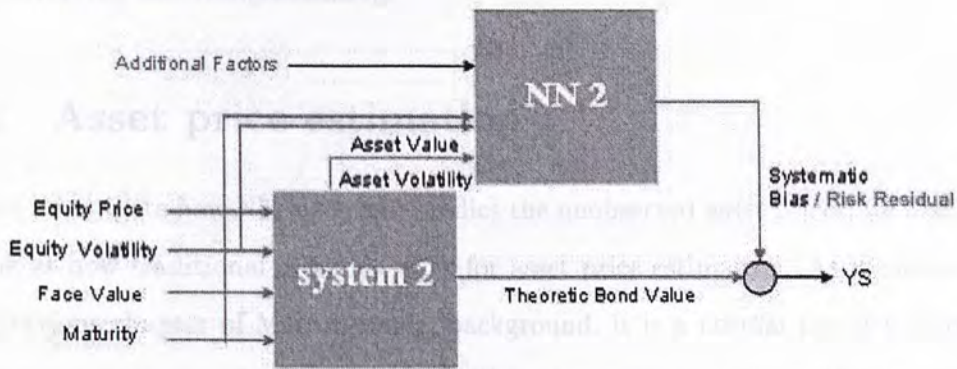


$$YS = f_M(V_t^{nn}, F, \sigma_{V_t}^{nn}, r, T) \tag{4.3}$$

$$\text{where } \theta = (\mu^{nn}, \sigma_{V_t}^{nn}) = h\left(\sum_{j=0}^m w_{kj}^{(2)} g\left(\sum_{i=0}^d w_{ji}^{(1)} \hat{V}_i\right)\right)$$

Figure 4.3: First neural network module (System 2) - solving unobserved asset value by learning the relationship among equity prices and asset prices

In figure 4.3, one network system in charge of the unobserved variable is placed in front of the Merton system. By considering the relationship among equity prices, unobserved asset prices and asset values from financial statement (FS), the goal of the network system is to learn that relationship by real life financial data so as to give the best estimation for those unobserved variables. In the next section, we will



$$YS = f_M(V_t^{nn}, F, \sigma_{V_t}^{nn}, r, T) + g_{NN}(y) \quad (4.4)$$

Figure 4.4: Second neural network module (System 3) - solving credit risk underestimation by predicting the *risk residual* (i.e. the difference between the observed and predicted spreads)

find that asset price estimation could also be treated as a problem of distribution estimation.

Yet this modular design is not quite enough for the second weakness of Merton model. Consequently, one more network system is placed.

In order to address the underestimation, we hope the network system can help us to predict the *risk residual* suffered by Merton's estimation. This time, however, the input variables of the system are not as clear as before. It is essential for us to study what key factors y can address such residual. For a detailed discussion refer to "quantifying accounting data noise" and chapter 5.

Based on structural models with the use of fairly flexible statistical models, bi-directional modeling is shown to be a modular design of a large / hybrid "system" in figure 4.4. After detailed and systematic analysis, top-down and bottom-up directions

are interacting and complementing.

4.2 Asset price estimation

Before investigate how NN learn and predict the unobserved asset prices, we first take a look at how traditional methods work for asset price estimation. As mentioned in the previous chapter of Merton model background, it is a normal practice to relate asset prices to equity prices and asset volatilities to equity volatilities. Having successfully implemented Merton model and develop it as a successful commercial product, Moody's KMV model is indeed doing a good work on this relationship and worth studying. If we refer the equation of Merton model from the equity-holder viewpoint, it is already relating the asset prices and asset volatilities to equity prices as

$$E = VN(d_1) - e^{-rT}FN(d_2). \quad (4.5)$$

However, one equation is not sufficient for two unknowns (the asset prices and asset volatilities). That is the fundamental problem in implementing Merton model. To solve that, Moody's KMV adds one more equation to relate asset volatilities and equity volatilities so that the two unknowns can be solved.

$$\sigma_E = \frac{V}{E}N(d_1)\sigma_V \quad (4.6)$$

The example of KMV shows that the relationship of asset and equity prices is very close. From the point of view of neural networks, it is possible to estimate asset price as a probability distribution. We can either treat the network output as a probability density function directly or, create a network in which the outputs are

taken as determining free parameters of a parametric family of probability densities¹.

For the network's objective function to be density function directly, Kullback and Leibler's (1951) cross-entropy error function can serve the purpose. Suppose P and Q are the candidate probability density functions, we would like to determine a posterior density $p(x)$ where $q(x)$ is a prior density for the random variable X . Since we want the guessed distribution as objective as possible, we use *maximum entropy principle* and its solution is based on

$$-\int p(x) \log p(x) dx \quad (4.7)$$

and it is identical to the solution of *minimum cross-entropy* as given by

$$I(p : q) = \int p(x) \log \left(\frac{p(x)}{q(x)} \right) dx \quad (4.8)$$

where $I(p : q)$ is called the Kullback-Leibler information criterion². It is an information theoretic measure of the "surprise" experienced when we believe X is described by q (prior knowledge) and are then informed that it is in fact described by p . To train the network, we associate a prior density $q(x)$ with any network output indexed by weights θ and denoted by q_θ . A log-normal distribution with approximated drift and volatility might be considered as an initial prior asset price density. Therefore, we would like to find an information theoretically optimal network weight θ^* that minimize $I(p : q_\theta)$. That is equivalent to find an optimal network weight that solves the problem

$$\max_{\theta} \frac{\sum_{i=1}^n \log q_{i\theta}}{n}. \quad (4.9)$$

¹The "shape" of the asset value distribution is usually assumed to be log-normal for facilitating the Merton's model implementation.

²See Buchen and Kelly (1996) to estimate asset distribution from option prices by similar entropic principles.

Similar experiment conducted by Atiya (2001) claims that cross-entropy error function may not be very favorable. As motivated by the new entropic approaches from Edelman (2004), local cross-entropy (LCE) may be a new direction for the error function of network training. Edelman claims that LCE is a slightly smoothed version of cross-entropy and the performance for modeling should be improved.

For the network in which the outputs are taken as determining free parameters of a assumed parametric family, we can approximate the asset price distribution as a Gaussian distribution (Rao, 1973, Lo, 1986, Duan et al., 2003). Therefore, it is perfect to use the squared error function for network training. Free parameters such as mean and standard deviation of the pre-defined density can then be determined.

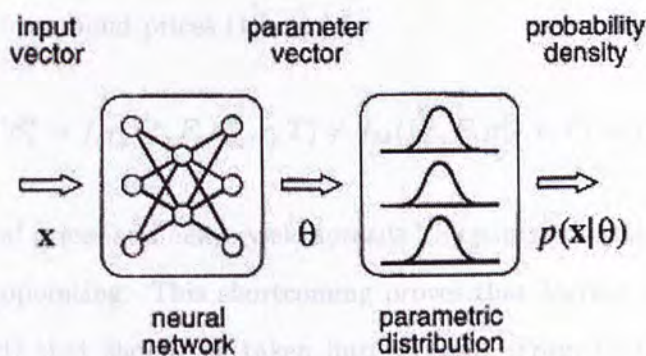


Figure 4.5: The parameters θ of a pre-defined parametric model for the asset price distribution of x are determined by the outputs of a neural network

4.3 Quantifying accounting data noise

In order to input suitable variables for the NN system account for the risk residual prediction, some literature such as Yu (2003) are studied and showing that the

presence of a sizable credit spread due to accounting transparency are found. That can attenuate some of the empirical problems associated with structural credit risk models.

In the last section, although the most "entropic" implied asset distribution above is estimated by one NN system, there is still one critical problem waiting to be solved - the underestimation of credit spreads. Following related literature, Kim, Ramaswamy and Sunderasan (1993), and Wei and Gou (1997) do suggest that Merton model itself is insufficient no matter how well the estimation was made for the underlying asset. The model is in fact not a consistent representation of actual equity and bond price (D_t^*) dynamics by the substitution of implied asset value and volatility. It is shown that the firm's assets and volatility implied from equity ($\hat{V}_t^e, \hat{\sigma}_{V_t}^e$) often disagree with those obtained from bond prices ($\hat{V}_t^b, \hat{\sigma}_{V_t}^b$) by

$$YS_t^e = f_M(\hat{V}_t^e, F, \hat{\sigma}_{V_t}^e, r, T) \neq f_M(\hat{V}_t^b, F, \hat{\sigma}_{V_t}^b, r, T) = YS_t^*. \quad (4.10)$$

That is why bond prices and *hence* yield spreads YS estimated by the Merton function f_M always disappointing. This shortcoming proves that Merton model does ignore the risk (spread) that should be taken into account. From bi-directional point of view, it should be induced by noisy data. This consideration is somehow consistent with the theory of discretionary disclosure, incomplete accounting information model of Duffie and Lando (2001) as well as accounting transparency and credit spreads regression analysis conducted by Yu (2003). In these papers, they find that the imperfect observation of firm value is one of the major components of the credit spread, while nearly most of the existing structural credit risk models (including Merton model) have ignored this issue and assumed everything is perfectly measured.

Now the question is how we can know that voluntary disclosed data are trustful.

Here the noise of data given by corporate is denoted as the discrepancy δ between the true asset values $p^*(x|\theta_o)$ and announced asset values V . Since the truth is always unknown, δ is usually estimated based on values V and the structural features of $p(x|\theta)$, as summarized by Xu (2002).

One important concept for accounting risk quantification is that - *distribution is actually temporal*. Therefore, we compare estimated distribution $p(x|\theta)$ by neural network with realized accounting data V at each time point. Given information up to current time t , we have the best estimated asset distribution (one-period-ahead) $p_{t+1}(x|\theta)$ up to time t . Suppose an accounting report V_{t+1} is released at time $t+1$ then, $p_{t+1}(x|\theta)$ is considered *being trustful* at time $t+1$. The discrepancy δ is estimated by the number of standard deviations that the estimated asset value (EAV) - mean of $p_{t+1}(x|\theta)$ will reach V_{t+1} . This number is called the Accounting Distance (AD)³.

$$AD_{t+1} = \frac{EAV - V_{t+1}}{EAV \times \sigma} \quad (4.11)$$

where σ is the estimated asset volatility. Instead of giving full trust to the current information provided by accounting report, we make comparison between estimated and real data for the period. The proposed AD helps to quantify the data noise in reports so that the inconsistency ($YS_t^* - YS_t^e$) of the Merton function or risk residual can be minimized.

Once we have better credit spreads estimation at the time point of accounting report, the neural network is trained to interpolate such function. So, not only the given time point, AD can be estimated during other periods. For a broader view, we can consider it as adding one more feature in the function of credit spread estimation.

³The concept of Accounting Distance is inspired by the similar concept proposed by KMV - Distance to Default. That is, the number of standard deviation that the value of the firm's assets must drop in order to reach the default point is considered.

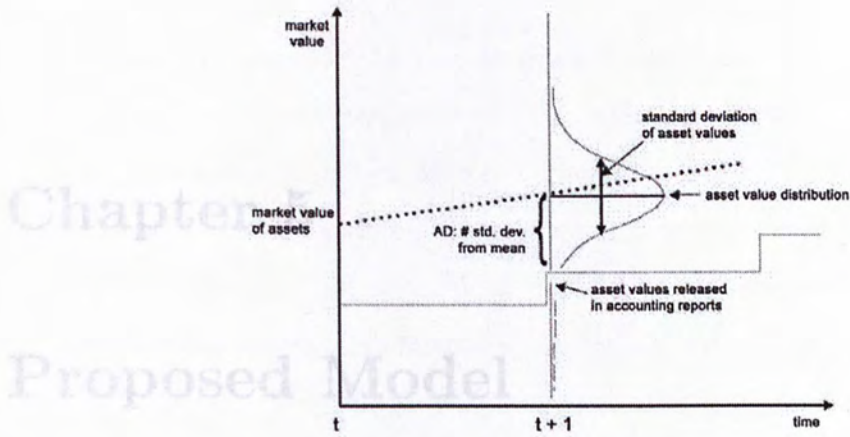


Figure 4.6: Accounting Distance

This feature is directly accounting for the inconsistency of the model. Referring to cross-sectional regression analysis used in Yu (2003), however, it is essential to use other factors provided in the public (e.g. credit rating of the bond, maturity & duration, economical factors, bond age) as the basic predictive features to estimate the residual of credit spreads in practice. Hence, the network learning will be more accurate. In the next chapter, a proposed model is presented.

5.1 Core of the model

The core of bi-directional modeling has two main components, the top-down and the bottom-up. They are responsible to model *theoretical part* and *practical part* of the true nature, respectively. In the application to default risk modeling based on prediction of yield spreads and hence the PD), we define:

1. True nature: That is the underlying mechanism of asset value distribution

Chapter 5

Proposed Model

In previous chapters, we have discussed how framework of bi-directional modeling interprets the Merton structural default model and the statistical NN model as a "system" in order to facilitate the design of a hybrid system. Following that modular design, we discuss in more detailed how the proposed model be realized in this chapter. First, we present the core of the model in detailed. Then, the analysis and selection of key features / variables for the model (especially for the risk residual) is shown and explained. The overall architecture of the model is illustrated at last.

5.1 Core of the model

The core of bi-directional modeling has two main components, the top-down and the bottom-up. They are responsible to model *theoretic part* and *chaotic part* of the *true nature* respectively. In the application to default risk modeling (specifically, prediction of yield spreads and hence the PD), we define:

1. **True nature:** That is the underlying mechanism of default, or the observed

market yield spreads.

2. **Theoretic part:** We choose the most profound and theoretic Merton model. The reason is very obvious at all. As this part should only response to the basic rationale in nature, Merton's framework can indeed do so. It is shown in literature that thousands of variants has already been developed further and still active nowadays, just because the model is conceptual and theoretical enough. That perfectly captures the essence of default. We denote the function as $f(\bullet)$.

$$f_M(\mathbf{x}) \text{ where } \mathbf{x} = (V_t, F, \sigma_{V_t}, r, T) \quad (5.1)$$

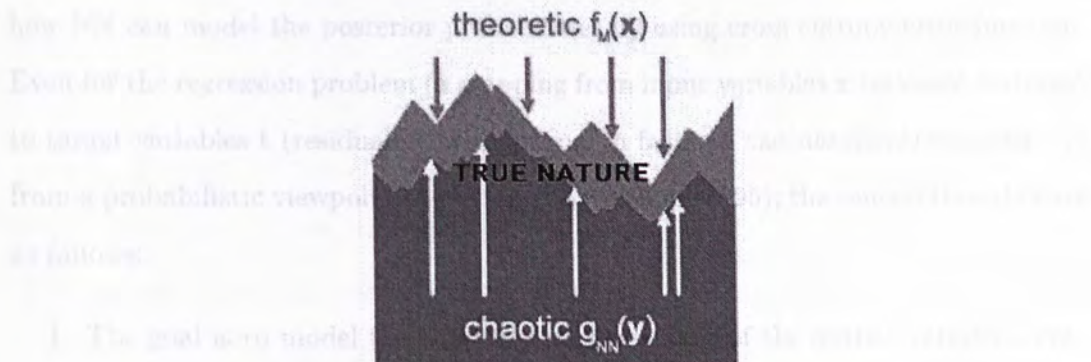


Figure 5.1: Core of "True Nature"

3. **Chaotic part:** Contrast to the rationale, this part response to the chaotic behavior in nature. Obviously, a structure-free model would be the best choice as none can really model chaos by structural approach. We select *two-layer feed-forward* networks as the statistical learning approach so that relationship

can automatically be learned and improved with experience. By using non-parametric estimation for the seemingly chaotic behavior, network learning is fairly suitable. The function of network is denoted by $g(\bullet)$.

$$g_{NN}(\mathbf{w}; \mathbf{y}) = h\left(\sum_{j=0}^m w_{kj}^{(2)} g\left(\sum_{i=0}^d w_{ji}^{(1)} y_i\right)\right) \text{ where } \mathbf{y} = (AD, MAT, LEV, VOL, AOS) \quad (5.2)$$

Based on our perception, the function g is trying to model the part that function f can never achieve. From the viewpoint of Gultekin et al. (1982), Jacquier and Jarrow (1996) and Connor and Lajbcygier (1997), the function g is predicting the *residuals* between the "conventional" or "classic" function f and the observed behavior (say, yield spreads) in true nature. Referring to the previous chapter, we have already seen how NN can model the posterior probabilities by using cross-entropy error function. Even for the regression problem (a mapping from input variables \mathbf{x} (selected features) to target variables \mathbf{t} (residuals)) at this time, in fact, we can *absolutely* consider NN from a probabilistic viewpoint. Mentioned by Bishop (1995), the central thoughts are as follows:

1. The goal is to model the *conditional distribution* of the output variables, conditioned on the input variables for regression problems. While for classification problems the goal is to model the *posterior probabilities* of class membership conditioned on the input variables.
2. The central goal in network learning is *not* to memorize the data, but to model the *underlying generator* of the data.
3. The most general and complete description of the generator is in terms of the probability density $p(x, t)$ in the joint input-target space.

Same as the situation in asset price distribution modeling, the error function and the corresponding optimization principle are the keys. This time, the error function is motivated by the *principle of maximum likelihood*. Because of the joint probability $p(x, t)$ can be transformed as

$$p(x, t) = p(t|x)p(x). \quad (5.3)$$

The likelihood for a set of training data $\{x^n, t^n\}$ can be written as follows

$$L = \prod_n p(x^n, t^n) = \prod_n p(t^n|x^n)p(x^n). \quad (5.4)$$

For convenience, negative logarithm of the likelihood is minimized and the error function E is therefore minimized as

$$E = -\ln L = -\sum_n \ln p(t^n|x^n) - \sum_n \ln p(x^n). \quad (5.5)$$

By assuming that there are c target variables t_k with $k = 1 \dots c$ and, the distribution of the target data is Gaussian, we would find that the conditional density of target variables is

$$p(t_k|x) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left(-\frac{\{g_k(w; x) - t_k\}^2}{2\sigma^2}\right) \quad (5.6)$$

where $g_k(w; x)$ is the output of a neural network and w is the weight parameters governing the network mapping. Hence, the previous error function E can further transformed as

$$E = \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^c \{g_k(w; x^n) - t_k^n\}^2 \quad (5.7)$$

which is actually our very familiar *squared error function*. After all, in order to use NN to model the conditional probability density for regression problem, squared error function derived from the maximum likelihood principle on the Gaussian assumption of target data can achieve.

However, if NN is really used for this prediction, more characteristics about this residual should be investigated in order that the input / predictor variables can be better selected. Feature extraction is the following process.

Referring to the modular design and analysis of bi-directional modeling in last chapter, we know that g can also be treated as a "system" which is mainly for the troubleshooting of the second weakness of f - the risk underestimation. We actually treat the residual that f has simply ignored *as yield spread to predict*. The sensitivity test is given in the next section.

5.2 Feature selection

The direction is now clear that we would like to use neural network to model function g of the risk residual and consider its features of credit spread. One worth noticing point is how our proposed model treats the input variables and target values during training the network. As we assume that all corporate bonds issued by the counterparties are very *credit-sensitive*, their credit spreads should be the best choice of being the network's target values. Hence, the predicted spreads should be coherent with the actual credit risk of those corporate. For the input variables, therefore, we would consider both bond-oriented variables for the actual bond yield spread assessment and corporate-oriented variables for the corporate credit risk assessment.

Summarized by the table 5.1, each element in the input vector \mathbf{y} of function g

(excluding AD) are all well-known to be predictors for spreads in normal industrial practice.

Table 5.1: Predictors for yield spread residual

Predictors	Corresponding Risk
MAT, DUR - maturity, duration (bond-oriented)	Not directly correspond to risk but useful in determining the yield curve shape
LEV - leverage (corporate-oriented)	Explain the structural risk coming from firms by distance between firm value and default boundary - similar to distance-to-default
VOL - volatility (corporate-oriented)	Also explain the structural risk from firms
AOS - amount outstanding (bond-oriented)	Explain the liquidity risk of the bond itself
AD (corporate-oriented)	Mainly account for the risk induced by incomplete (accounting) information which is always neglected by structural models. It can also refer as the disclosure/transparency of one firm

In order to show how well these variables predicting credit spread and its residual, and to get the sense of nonlinearity of the relationship in nature, let's now consider our proposed model predictors: accounting distance (AD).

Figure 5.2 plots the relationship between the yield spreads in basis points and the value of proposed AD for the using the dataset from the Canadian market¹. By using a smoothed line superimposed within the data sample, we can successfully fit a *trendline* for the data. Obviously, variable AD shows its significant predictiveness. The red trendline plotted in the figure suggests that the yield spread increases as the AD increases in general. It does make perfect sense because *more noisy the accounting reports should result in more risky firms and higher yield spreads*. Most importantly,

¹This data set will be used throughout this paper. For detailed experimental preparation and results refer to chapter 6.

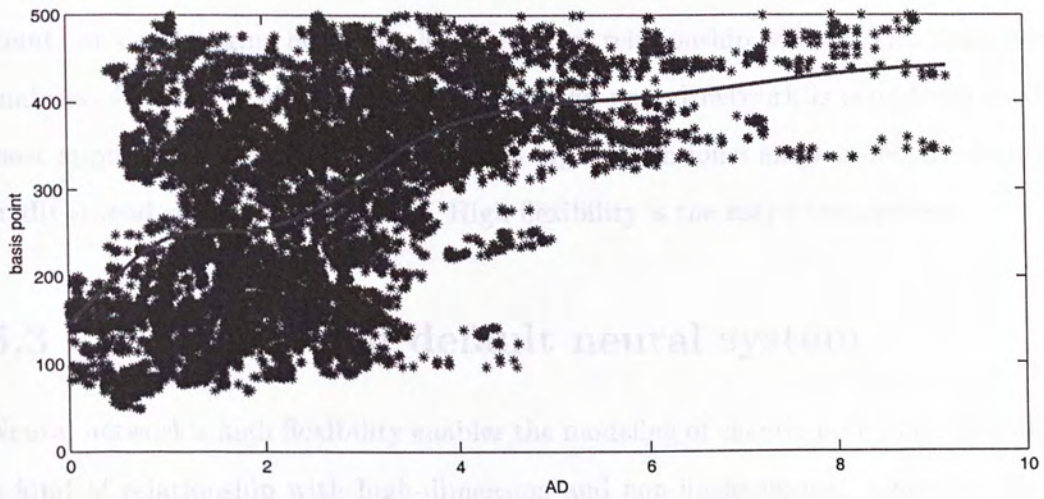


Figure 5.2: Relationships of AD on yield spreads in the sample set of Canadian firms and bonds: the level of yield spreads versus AD in panel A shows the nonlinear nature (a smoothed trendline superimposed on data) of the relationship and how predictive the AD is (in a univariate sense).

the trend prediction is not in a simple linear way but logarithmic (nonlinear).

One crucial issue we must notice is the non-linear character of the relationships showing above, that pushes us away from simple linear regression methods. In fact, similar relationships exist for each of the input variables used in the chaotic part of the bi-directional modeling as shown in figure 5.3 and 5.4. If, not only considering the univariate relationship (one independent variable with respect to one dependent variable) but all the relationships among variables (multivariate) are taken into account, we can imagine how complex the actual relationship will be. This sensitivity analysis of variables gives strong evidence why neural network is considered as the most appropriate approach to model such high-dimensional and non-linear nature of credit spread residual relationship. High flexibility is the major requirement.

5.3 Bi-directional default neural system

Neural network's high flexibility enables the modeling of chaotic part which is always a kind of relationship with high-dimension and non-linear nature. Once the whole modeling process combining with the theoretic part is carried out in a rational manner, it must outperform any existing uni-directional modeling methods and, the detailed experimental result will be shown in the next chapter. In this section, a walk-through of the default neural system is presented.

In Merton (1974), the valuation of risky debt at any time point t is $D_t = \min\{V_t, F\}$ where the asset price dynamic V_t is used. Recall that the pricing equation is given by

$$D_t = V_t \Phi(-d(V_t, t, \sigma_{V_t})) + F e^{-r(T-t)} \Phi(d(V_t, t, \sigma_{V_t}) - \sigma_{V_t} \sqrt{T-t}) \quad (5.8)$$

Consequently, the model implies both the credit spread and the probability of

default of the corporate which is very important in credit risk measurement. We however do not observe $V_0, V_1, V_2 \dots V_T$ and that is the first empirical problem of Merton model encountered. We therefore use a time series of the equity values $E_0, E_1, E_2 \dots E_T$. Based on the equity pricing equations introduced in last chapter (4.5 and 4.6), they define the transformations between E_t and V_t and, σ_{E_t} and σ_{V_t} respectively. They are denoted by

$$E_t = g'(V_t, F, \hat{\sigma}_{V_t}, r, T) \quad (5.9)$$

and

$$\sigma_{E_t} = g''(V_t, E_t, F, \hat{\sigma}_{V_t}, r, T) \quad (5.10)$$

where $\hat{\sigma}_{V_t}$ is the estimated asset volatility as we do not observe it either.

For a better learning and generalization, we of course are more interested in the general distribution of the asset price dynamic rather than using the transformation equations to calculate the values at each time point. As a result, neural network is used. Recall that the network can either treat the output as a density function directly or as determining free parameters of a predefined density. In the proposed neural system, the latter method would be adopted. Since the point estimate for the asset value can be obtained by

$$\hat{V}_t = g'^{-1}(E_t, F, \hat{\sigma}_{V_t}, r, T) \quad (5.11)$$

and it is a continuously differentiable function of $\hat{\sigma}_{V_t}$, the distribution for the asset value can be assumed and approximated by a Gaussian distribution (Rao (1973), Lo (1986), Duan et al. (2003)). As we have assumed that the distribution of the target data - asset value is Gaussian, it would then be perfect to use the squared error function for training the network.

However, it can be observed that the estimated spread \hat{YS} is different greatly from the real market behaviour and that is the second problem of Merton model. We introduce additive residual in our proposed model, that is

$$YS^* = f_M(\hat{V}_t, F, \sigma_{\hat{V}_t}, r, T) + rr \quad (5.12)$$

where rr is risk residual that can never be explained by Merton model. Here we can see *two* core parts of the bi-directional modelling: Merton model f_M is a theoretic representation of the credit spread, while the residual is a chaotic representation of the model. Interestingly, the chaotic part is actually responsible to the empirical world and could be interpreted as a kind of unexplained variation from the model when confronted to empirical data, say, *spreads, omitted variable* like AD, or *bad model specification*. After all, the proposed neural system can be depicted as follows:

$$YS^* = YS^{ns} = f_M(\mathbf{x}^{nn}) + g_{NN}(\mathbf{y}) \quad (5.13)$$

where \mathbf{x}^{nn} are the neural network estimated variables, $g_{NN}(\bullet)$ is the neural network function and \mathbf{y} are the factors explaining the residual rr .

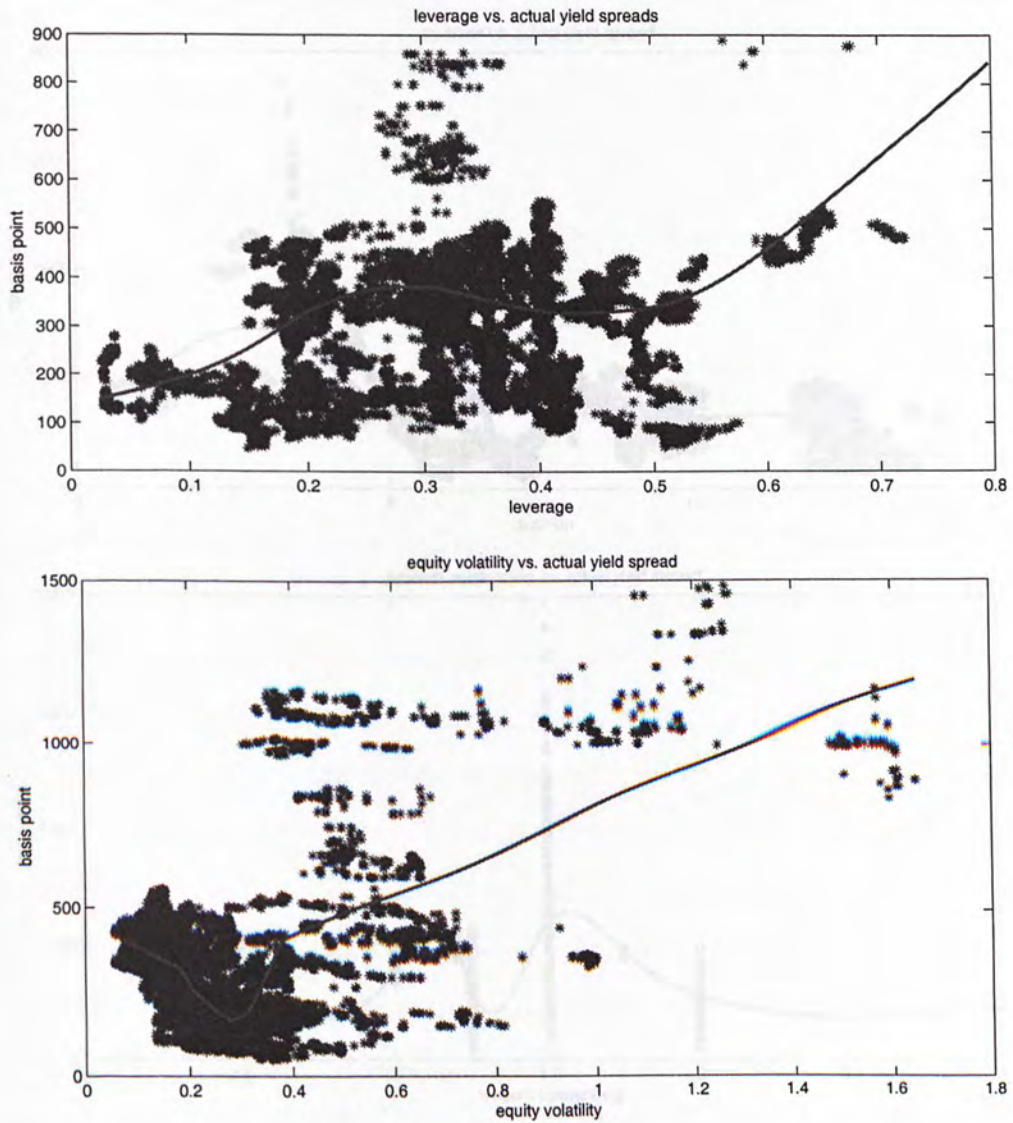


Figure 5.3: Relationship between other variables and yield spreads (1): Yield spread inclines gradually and goes steeper at the end while leverage and equity volatility increase. The level of yield spreads versus leverage and volatility shows the highly nonlinear nature (*polynomial* trendlines fitted in data with the order of 3 and 2 respectively) of the relationship.

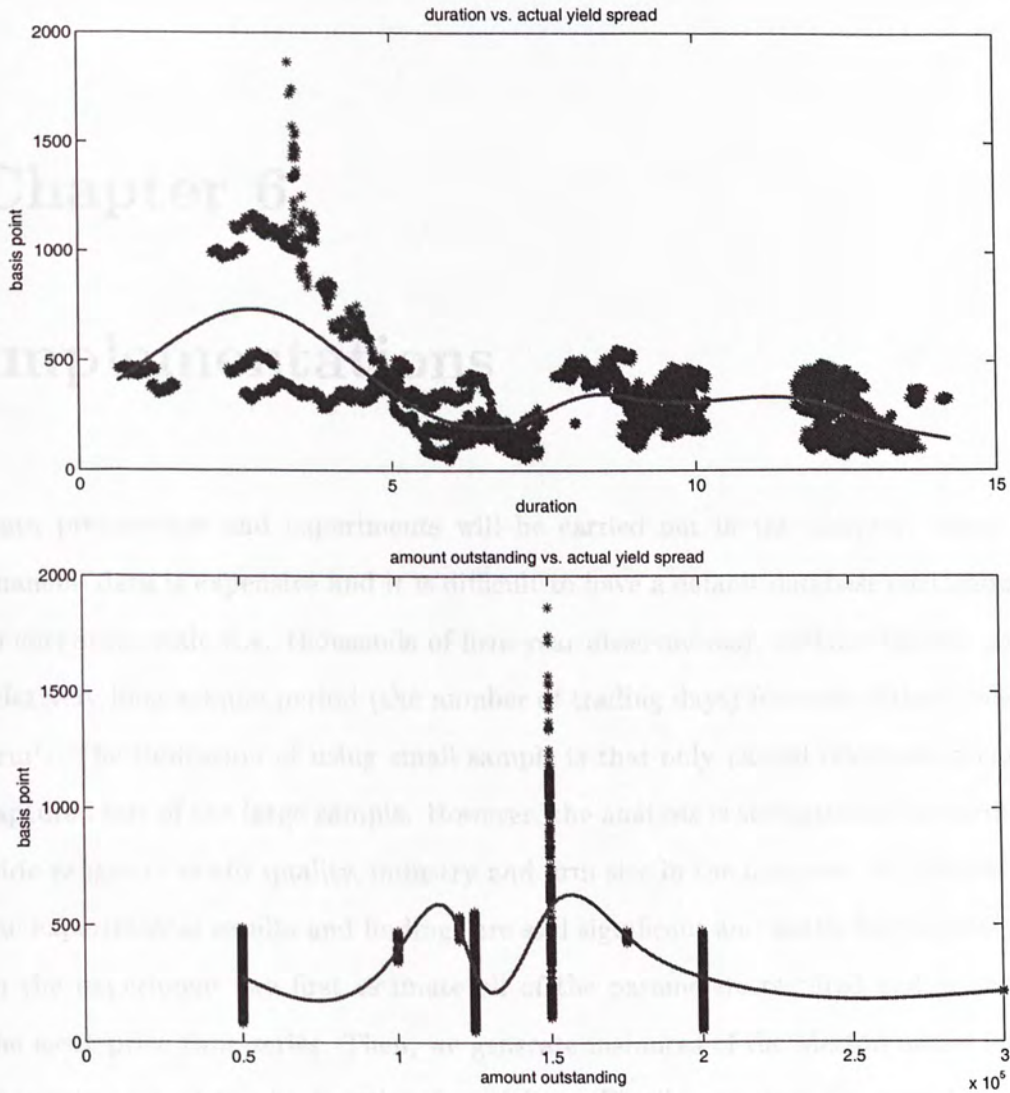


Figure 5.4: Relationship between other variables and yield spreads (2): For duration, yield spread reaches its highest point in the middle and then declines and stays flat after. Relationship between amount outstanding and yield spread is almost like normal distributed. Nonlinear nature is observed (*polynomial* trendlines fitted in data with the order of 5 and 4 respectively).

Chapter 6

Implementations

Data preparation and experiments will be carried out in this chapter. Since the financial data is expensive and it is difficult to have a default database containing up to corporate-scale (i.e. thousands of firm-year observations), we have instead used a relatively long sample period (the number of trading days) for each of the Canadian firm¹. The limitation of using small sample is that only partial relationship can be captured out of the large sample. However, the analysis is strengthened by including wide ranges of credit quality, industry and firm size in the data set. We believe that the experimental results and findings are still significant and worth further studying. In the experiment, we first estimate all of the parameters required and preprocess the asset price time-series. Then, we generate instances of the Merton model for the theoretic part of the bi-directional modeling. Finally, we train the neural network to predict risk residual for chaotic part by using yield spreads from bond market as target values. Corporate yield spreads are hence estimated and empirical results are

¹In our database, we use 7 firms and 13 bonds from Canadian market because default data is expensive. Also, it is not usual for Asian financial data to be managed like in Reuters and DataStream officially. For better use of the sparse data for prediction, we use about 15 years of trading days for each firm. Therefore, we have a total of $7 \times 15 \times 252$ sample points for experiment.

presented at the end of this chapter.

6.1 Data preparation

We select a sample of firms from Canadian market with simple capital structures to implement bi-directional modeling, so that learning can focus on the pricing errors related to the deficiencies of the uni-directional model instead of the complication of liabilities. We use bond specifics, equity prices, the number of shares outstanding and interest rate from DataStream between 1993 and 2005. Data on company histories and financials such as total assets, annualized long- and short- term debts on accounting reports are collected from Reuters.

By assuming default is a state that affects all obligations of that firm equally, credit-sensitive bonds issued are selected as the target values of the network. We consider the bonds having standard cashflows. That is, fixed rate coupons and principal at maturity. We exclude convertible bonds and bonds with call options or put options. In order to keep the capital structure simple, we choose firms with only one or two publicly traded bonds in the market. Of course, the firms we choose must have publicly traded stock so as to estimate asset price and its volatility for the Merton model. Minimum ten years of equity price data must be contained for optimal network training result. The final sample includes 7 firms and 13 bonds². Wide range of industry is shown in table 6.1.

Table 6.2 and 6.3 present summary statistics on the bonds and issuers in the sample. Table 6.2 shows that the average bond issue in our sample is associated with a coupon rate of 7.03%, a duration of 8.7 years and an amount outstanding of

²We exclude financial firms from our sample so that the leverage ratios in the dataset are more comparable. For detailed discussion, see Lyden and Saraniti (2000) and Eom, Helwege and Huang (2004).

Table 6.1: Canadian firms and corresponding industries in the sample

Firm	Industry
Hudson's Bay	Wholesale and Retail Stores
Sears Canada	Wholesale and Retail Stores
Enbridge	Electronic and High Technology
Gaz Metro LP.	Oil and Gas
Loblaw	Grocery
Sobeys	Construction
Saskatchewan Wheat Pool	Farming and Agriculture

135 thousand. The firms have relatively high yield spreads averaged at 293 bp and large coverage of credit quality spectrum, ranged from 48 bp to 2106 bp. Also, they are very large with the average asset value up to 6.7 billion with fairly low leverage. Table 6.3 indicates that the observations in the sample come from different interest rate environments. Average interest rate (using one-year treasury rate) ranges from the highest point 7.26% in 1995 down to the lowest 2.58% in 2004.

Table 6.2: Summary statistics on the bonds and issuers in the sample

	Mean	Std. Dev.	Min	Max
Coupon(%)	7.030	1.202	4.9	10.45
Duration(year)	8.665	3.514	0.629	14.22
Amount outstanding (000s)	135	49	50	300
Yield spread (bp)	293.494	216.374	47.78	2106.02
Asset value (\$ millions)	6699	5966	520	24327
Leverage	0.312	0.124	0.026	0.798
Asset volatility (over 60 days)	0.172	0.133	0.037	1.019
Accounting distance	2.475	1.461	0	9.179

6.2 Experiment

In this section, we will discuss the overall procedure of implementing the bi-directional modeling in an application to the Merton model. First of all, the estimation of

Table 6.3: Average risk-free rate in the sample period

Observation Year	Average 1-year Treasury Rate (%)
1993	5.17
1994	6.50
1995	7.26
1996	4.74
1997	4.10
1998	5.04
1999	5.14
2000	5.90
2001	3.95
2002	3.00
2003	3.03
2004	2.58
2005	2.72

parameters for the model is introduced.

In the Merton structural model, a set of parameters including firm value, levels of debt and assets, asset volatility and the risk-free rate must be estimated. In addition, parameters related to bond features are required for the implementation also. Table 6.4 summarizes how to estimate three main types of parameters, namely bond features, firm characteristics and interest rate.

1. **Firm-related parameters:** To estimate the firm value as an asset price distribution, we first calculated the implied asset value and volatility by the KMV measure (equations 4.5 & 4.6). Then, a network will be trained based on the proposed neural system. Parameters of firm's equity price and volatility, sum of long- and short-term debts (book value of total liabilities) as face values and risk-free rate are essential (equation 4.3). The leverage is measured as total liabilities over the sum of total liabilities and market value of equity (i.e. total assets reported in the firm's financial statement). For the calculation of the

Table 6.4: Estimation of parameters

Parameter	Description	Estimation	Data Source
Bond features:	Duration DUR	Given	DataStream
	Maturity MAT	Given	DataStream
	Amount outstanding AOS	Given	DataStream
	Yield spread YS	Yield-to-maturity (YTM) of bond minus YTM of treasury bill	DataStream
	Face value F	Total liabilities: long- plus short- term debts	Reuters
Firm characteristics:	Market value of equity MV	Equity price times number of shares outstanding	DataStream
	Total assets V_{FS}	Total liabilities plus market value of equity	Reuters
	Leverage LEV	Total liabilities over total assets	Reuters
	Equity volatility σ_E	Historical volatility	DataStream
	Firm value V	Parametric prediction by neural network with the KMV measure of (implied) asset value	DataStream and Reuters
	Asset volatility σ_V	Historical equity volatility adjusted for leverage	DataStream and Reuters
	Accounting Distance AD	Number of standard deviations that the firm value will reach total assets	DataStream and Reuters
Interest rate:	Risk-free rate r	One year Canadian government treasury bill	DataStream

corporate-oriented variable AD, total assets and the mean of estimated asset distribution are used (equation 4.11).

2. **Interest rate parameters:** Interest rate parameters are estimated by using the yield to maturity of a representative one year Canadian government T-bill as risk-free rate.
3. **Bond related parameters:** Most of the parameters are simply given in the database. The yield spread observed in the market is estimated by the difference between the yields from the bond and the one year T-bill. One point needed to be noticed is that total liabilities are used as the face value (i.e. the default boundary of the Merton structural model) instead of the bond's face value itself.

After we have estimated all the parameters and organized them into one data set, data preprocessing is a very important step right before any of the learning procedure by neural network. That is helpful especially when high-dimensional problems are encountered like the financial ones. Normally preprocessing includes feature extraction, data normalization and feature selection. In the step one of the experiment, an asset price time-series is preprocessed for prediction.

Step One: Preprocessing asset price time-series: We separate a time series $\{z_t\}_{t=1}$ of asset price of 7 Canadian corporate into three parts: training, validation and test series. One fourth of the series for the validation set, one fourth for the test set and one half for the training set. Since the data is a *function of time* and our goal is to predict the value of z a short time in the future, the set is extracted from the series by shifting a sliding window successively as shown in figure 6.1 below, where input pattern is denoted as

$Z(t) = [z_t, z_{t+1}, \dots, z_{t+d}]$ and its dimension is hence set at $d + 1$, and output pattern is denoted as $U(t) = [\mu_{t+d+1}, \sigma_{V_{t+d+1}}]$ at each time point t .

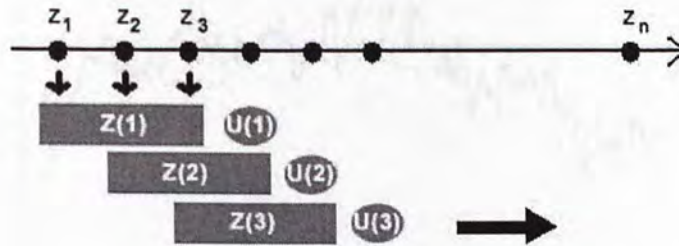


Figure 6.1: A sliding window extracts a sample set of input-output pairs from an asset price time-series.

Here we set the dimension of input space to be 60 (trading days) to ensure the prediction based on sufficient historical data. A set of raw input data is then extracted from the time-series. To facilitate the network training and make efficiency use of computation time, normalization of input data are performed so that only the *subset* of input features from raw data set needs to be used and most of the information can still be retained. We use two algorithms for data normalization: Simple Normalization (normalize input/output feature separately to zero mean and unit variance) and Principal Components Analysis (PCA). After the full set of pattern is normalized, inputs and targets are then scaled and fall in the range $[-1, 1]$ approximately so that the training algorithm can work best. By performing PCA, we have retained those principal components which account for 99.9% of the variation in the input data set while the size of the space has significantly reduced from 60 to 6. Now, the resulting input pattern is ready to training further.

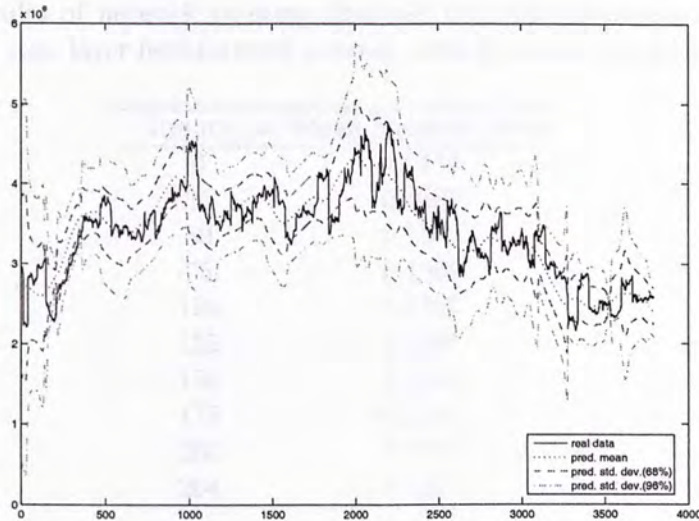


Figure 6.2: Predicted temporal asset distribution (mean and std. dev.) for Hudson's Bay in 1990-2004 by neural network

One of the prediction results ('Hudson's Bay') of the asset price dynamics from those seven firms are plotted in figure 6.2. For this example, a total of 3854 trading day records are used for the analysis. A two-layer feed-forward network with 6 input nodes (normalized input space), 20 nodes in the hidden layer, and 2 nodes as the output variables (parameters: mean and standard deviation of a Gaussian distribution). That is called a 6-20-2 network architecture. The target asset value is assumed to be Gaussian and *mean squared error (MSE)* function is therefore used for training. Table 6.5 summarizes the results below. The MSE decreases up to 200 iterations and then levels off at 204 with lowest error value (0.1277).

Step Two: Generating instances of Merton equation (theoretic part): After all

Table 6.5: Results of network training obtained by using preprocessed asset price time-series and two- layer feed-forward network with 20 nodes in hidden layer (6-20-2 architecture).

Iterations	Mean Squared Error
0	6.3433
25	0.1667
50	0.1399
75	0.1368
100	0.1357
125	0.1341
150	0.1318
175	0.1298
200	0.1277
204	0.1277

of the parameters are estimated and most importantly, the asset price time-series generator: a Gaussian distribution with mean μ and standard deviation σ_{V_t} being predicted by the network in step one, we start to generate instances of our theoretic part - the Merton structural model.

In our sample set of the seven Canadian firms, we have generated 26,126 instances of yield spreads from the Merton equation (equations 4.2 & 4.3) starting in January, 1990 through March, 2005 which are used to calculate the risk residual.

By observing the real spreads YS^* in the market, we can therefore calculate the risk residual rr by $YS^* - f_M(\bullet)$. Here we have generated 13,071 instances of risk residual from 13 bonds in our sample for further training (in-sample) and testing (out-of-sample) in step three.

Although we observe that the yield spreads from the Merton equation are far less than the observed spreads of the publicly-traded bonds, we trust the quality

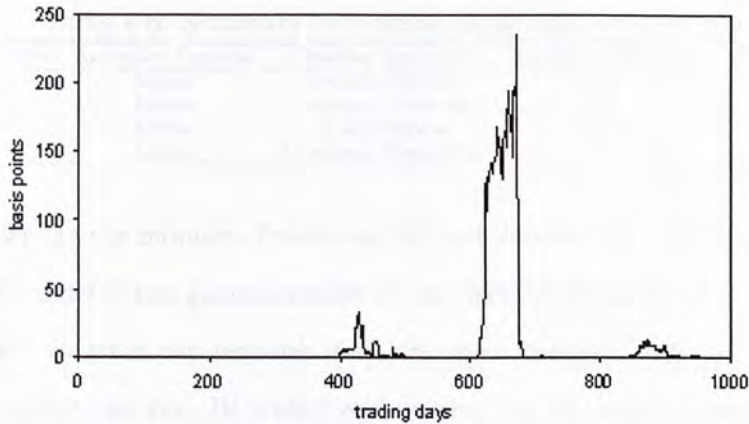


Figure 6.3: An instance of yield spreads generated from the Merton equation for Sears Canada

of the spreads generated by the theoretic model as KMV does³. That is, the spreads are in fact reflecting the credit risk in certain extent. The risk residual must be referring to other sort of risk (spread) and, able to be modeled and generated. Therefore, one network is used at the final step of the proposed system so as to approximate that function.

Step Three: Training network for risk residual prediction (chaotic part): In the very beginning of the network training, we first testify which the best training algorithm is given the same sample set, the number of nodes in hidden layer and the activation function in the output layer. Based on the number of iterations and test error (root mean squared error - RMSE), we choose Levenberg-Marquardt algorithm for the rest of the analysis.

In order to avoid the over-fitting problem of the trained network, we carefully

³Moody's KMV model uses Merton's model as its very major and crucial component. With the help of huge database support and fine tuning parameters, KMV model can predict credit risk with promising accuracy.

Table 6.6: Summary of training algorithm performance

Number of nodes	Activation function	Training algorithm	Number of iterations	Test error (RMSE)
20	Linear	Gradient Descent	1000	0.38597
20	Linear	Conjugate Gradient	252	0.04950
20	Linear	Quasi-Newton	131	0.04359
20	Linear	Levenberg-Marquardt	72	0.04062

consider (1) the number of nodes in the architecture (i.e. the complexity of the model) and (2) the generalization of the network training. For the first issue, we start to train the network from the most complex one (i.e. 40 nodes) to the simplest one (i.e. 10 nodes) and observe the test error changes in between. So that, the network is *just large enough* to provide an adequate fit but not "memorizing" the data. For the second issue, we consider two methods: *regularization* and *early stopping*. Briefly speaking, regularization is adding one regularizer called *weight decay*

$$\frac{1}{n} \sum_i w_i^2 \quad (6.1)$$

in the existing squared error function so that the trained network will be forced to have smaller weights and biases and hence smoother. Early stopping is using the error in the validation set to monitor the minimum number of iterations and weights and biases. Table 6.7 summarizes the training result below.

Table 6.7: Prediction errors of the chaotic part of the bi-directional modeling on training (in-sample) and testing (out-of-sample) sets.

Number of nodes	Generalization methods	Number of iterations	RMSE (in-sample)	RMSE (out-of-sample)
40	Regularization	20	0.1455	0.9959
30	Regularization	23	0.1840	0.9308
20	Early stopping	8	0.5335	0.7961
10	Early stopping	13	0.1332	1.4999

As we can observe from the result, the prediction errors in the test sets vary with the complexity of the network architecture. It is shown that either too complex (40 nodes) or too simple (10 nodes) model did not provide suitable

fitness of data and result in large RMSE (over 0.93). RMSE in the training sets also give hints that once it is driven to a very small value (under 0.18), over-fitting problem still occurs. The time-series of the risk residual on the test set is plotted in figure 6.4. With generalizing to the new situation, the predicted basis points are able to capture the lowest point at 430 bp and have generally predicted the average level in 120-day testing set. The chaotic $g_{NN}(\bullet)$ is capable to capture the risk ignored by the theoretic $f_M(\bullet)$.

After all, based on the equation 5.13, the corporate yield spreads are predicted by bi-directional modeling.

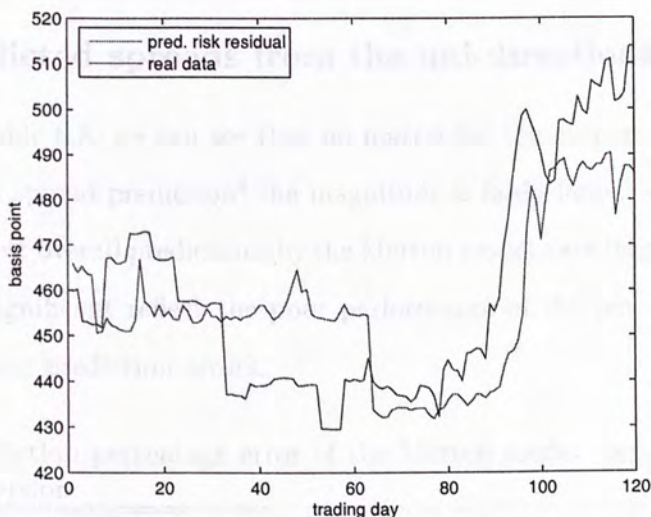


Figure 6.4: The prediction results of risk residual on testing set of bi-directional modeling

6.3 Empirical results

Before showing the experimental results of the proposed bi-directional neural system, we must first take a close look at the actual ability of the uni-directional models: the Merton model and the ordinary least squares (OLS) regression to fit the market behavior, so that comparison can be made after. Figure 6.5 and 6.6 plot the predicted yield spreads from the uni-directional models and the actual yield spreads observed in the bond market. It clearly shows that extreme under- and over-estimation happen in many cases. When we present this figure in terms of percentage errors, it will be something much more obvious.

6.3.1 Predicted spreads from the uni-directional models

As plotted in table 6.8, we can see that no matter for the average percentage errors in yield or yield spread prediction⁴ the magnitude is fairly large. Also, the standard deviations suggest overall predictions by the Merton model have large dispersion error. These figures significant reflect the poor performance of the pure structural model and its systematic prediction errors.

Table 6.8: Prediction percentage error of the Merton model: very large percentage errors and dispersion

	Percentage Error in Yield	Absolute Percentage Error in Yield	Percentage Error in Yield Spread	Absolute Percentage Error in Yield Spread
Mean	-28.9%	55.6%	-78.6%	115.2%
Std. Dev.	153.7%	146.2%	211.0%	193.3%

⁴The percentage errors in yields and spreads, and their absolute values, are calculated as the predicted value minus the observed value and then divided by the observed one. The errors are generated from implementing the Merton model using 13 bonds with simple capital structures during 1993-2005. We consider the error in spreads to be the key measure of the model performance. They relate directly to the risk residual being predicted by our proposed networks.

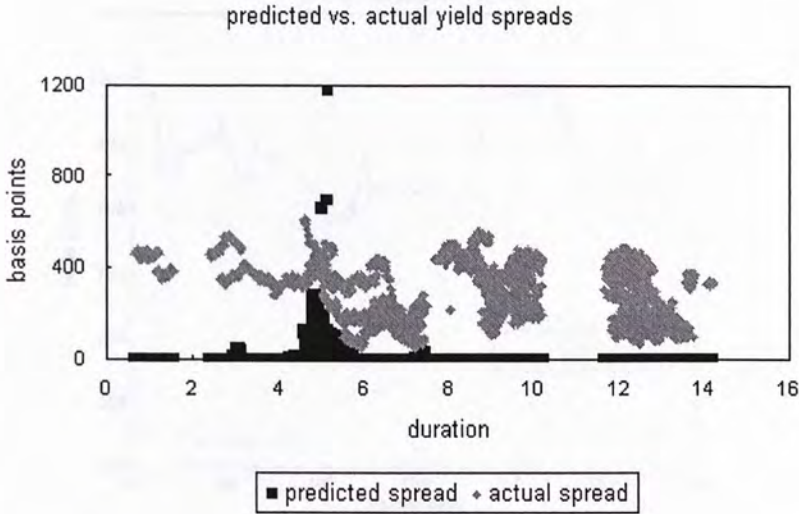


Figure 6.5: Performance of the Merton structural model: extreme under- and over-estimation

Based on the observations of the relationships between yield spreads and selected features (figure 5.2, 5.3 & 5.4), we suggested that the concern of nonlinearity should be the key of modeling. We conclude in the last chapter that nonlinear modeling and learning such as neural networks and hybrid systems must be preferred in yield spread prediction. To testify the fitness of using nonlinear model, we therefore apply another uni-directional model: simplest OLS (equation 3.6) regression to estimate the yield spreads and observe its accuracy in the out-of-sample set.

As shown in figure 6.6, the prediction result of linear regression is very disappointing and shown extreme over-estimation. The average out-of-sample RMSE is as high as 1.634. This figure significant reflect the poor performance of the pure statistical model and its systematic prediction errors.

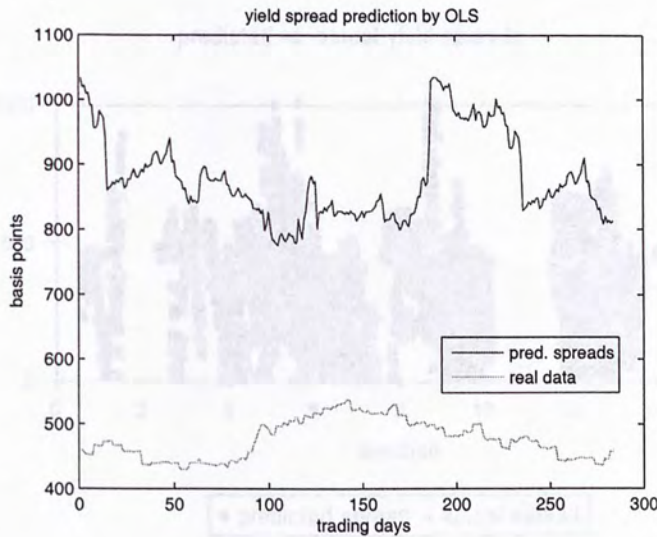


Figure 6.6: Performance of the OLS regression (out-of-sample): extreme over-estimation

6.3.2 Predicted spreads from the proposed bi-directional model

Finally, the bi-directional modeling is implemented and the prediction results are shown in the following figure and table. The new approach has significantly improved the average percentage errors of the overall prediction especially, the error in yield spread is just -0.5% . Without keeping tracking to every single data point in the training set, the network prediction has decreased the generalization error. Even though examples of over- and under- estimation still exist, a majority of prediction shows promising accuracy already.

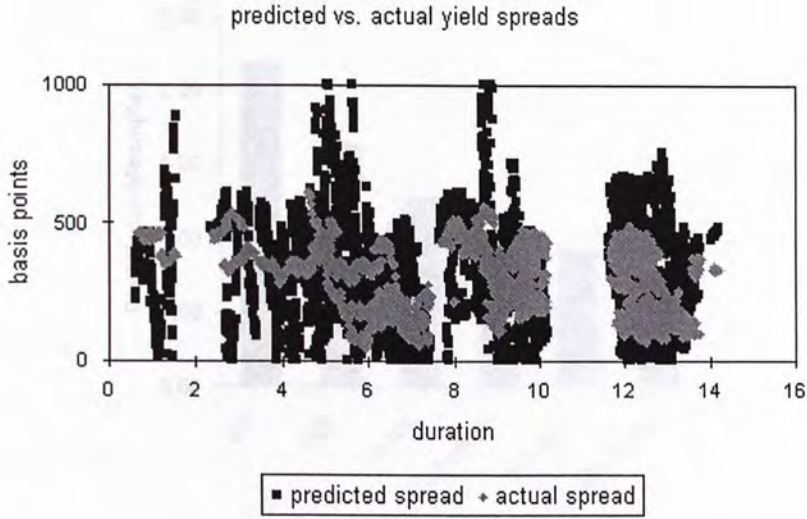


Figure 6.7: Performance of the bi-directional modeling: ability to generate high yield spreads and match the observed market.

Table 6.9: Prediction percentage error of the bi-directional modeling: average percentage errors significantly improved

	Percentage Error in Yield	Absolute Percentage Error in Yield	Percentage Error in Yield Spread	Absolute Percentage Error in Yield Spread
Mean	-2.8%	33.5%	-0.5%	87.4%
Std. Dev.	95.6%	89.6%	185.2%	163.3%

6.3.3 Performance comparison

As plotted in figure 6.8, the performance comparison among both uni- and bi-directional models is summarized in terms of the RMSE in the out-of-sample set⁵. The proposed bi-directional model (with the lowest test error 0.767) is shown to outperform other uni-directional models such as the Merton model (with the test error 1.272), OLS

⁵The behavior of the root mean square error (RMSE) in the out-of-sample set significantly shows the generalization ability for the tested models. With a lower generalization (test) error, the model finds more support to characterize regularities and therefore generalizes better.

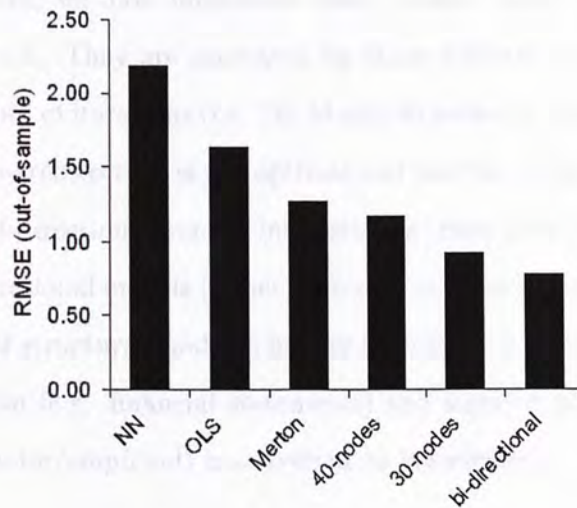


Figure 6.8: Accuracy for the six tested models in the out-of-sample set: the bars depict the performance of each of the default models tested (less RMSE represents better prediction). Note the performance gap between NN, OLS (statistical) and Merton model (structural). Also note the performance gap between Merton model and our proposed bi-directional model. These gaps represent the gain in model accuracy from incorporating additional financial information for prediction and learning process.

regression (1.634) and neural network learning (2.192). We can also observe that the performance of Merton structural model is better than pure statistical regression models such as OLS and NN. The gap of RMSE difference can be seen as representing the gain in model accuracy from incorporating additional financial information for credit risk prediction. The flexibility and power of NN seem to be "over-acting" when it is used without any financial theoretic ground. Bi-directional modeling (with modular design and additional variables) is therefore shown to provide platform for merging both statistical and structural model so that the performance is further improved. *True nature of dynamics is theoretic (structural) plus chaotic (statistical).*

In the experiment, we have implemented three models based on our proposed bi-directional framework. They are generated by three different network architectures with different number of iterations (i.e. 20, 30 and 40 nodes in the hidden layers). We found that 20-node-architecture is the optimal and has the lowest test error. Hence, it is our ultimate bi-directional model. Interestingly, these three models also perform better than uni-directional models (either pure structural or statistical). We conclude that the accuracy of structural model is further improved by incorporating additional financial information (e.g. financial statements) and learning process by which real life experience (chaotic/empirical) is converted to knowledge.

Chapter 7

Conclusions

Motivated by the shortcomings of the uni-directional modeling (i.e. pure structural and statistical models), we propose a hybrid financial risk model using a combination of structural model and neural networks. We demonstrate the proposed framework by applying on the Merton structural default model. The classic Merton model is theoretically strong. However, it faces several empirical problems during implementation. In this paper, we particularly focus on unobservable underlying asset and severe risk underestimation problems. Though interested researchers in academia suggest many more theoretically stronger model and turn down most of the pure empirical approaches for those problems. The proposed hybrid system called bi-directional modeling tries to merge the best world between structural and statistical methodologies. Making use of the advantages in statistical learning theory, we can see its possibility of solving empirical problems in the Merton model. It is also shown that better accuracy in prediction can be made once the model is embedded a learning process by which real life experience is converted to knowledge and incorporating additional financial information onto uni-directional models.

Grounded on the above basic concept, the definition and selection of explanatory features for model inconsistency (i.e. the risk residual) would be the major problem encountered. In this paper, we define Accounting Distance (AD), number of standard deviations that the estimated asset value will reach the real data released in the accounting report, so that the accuracy of corporate yield spreads can be increased. Moreover, sensitivity analysis of selected features has been carried out to testify the nonlinear nature of the data and hence the neural networks and hybrid models are proposed to be used in the framework. For experimental demonstration, an application to seven Canadian firms and corresponding thirteen bonds in the real market is presented. The empirical results show that the proposed bi-directional model outperforms existing financial models including the Merton model, OLS regression and neural network learning. Data limitation in the experiment is still contained. It would be worth further studying for more complete data analysis.

- [2] Anderson, B., Sudarshan, S.: A comparative study of corporate bond yields: An early start at risk-neutral pricing. *Finance*, Vol. 24 (2003) 197-208
- [4] Agive, A.F.: Bankruptcy prediction by neural networks: A review and new results. *IEEE Transactions on Systems, Man, and Cybernetics* (2003) 625-637
- [5] Rabinov, G.: Neural networks in financial engineering. *World Scientific Press*, (1998).
- [6] Black, P., Scholes, M.: Pricing options and liabilities. *IN Financial Engineering*, 2
- [7] Brennan, T., Kelly, M.: An explicit solution to a family of parabolic partial differential equations. *Annals of Applied Probability* (1985) 773-780

Bibliography

- [1] Altman, E.: Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance*, Vol. 13, (1968) 589–609
- [2] Altman, E., Marco, G., Varetto, F.: Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks. *Journal of Banking and Finance*, Vol. 18, (1994) 505–529
- [3] Anderson, R., Sundaresan, S.: A comparative study of structural models of corporate bond yields: An exploratory investigation. *Journal of Banking and Finance*, Vol. 24, (2000) 255–269
- [4] Atiya, A.F.: Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Transactions on Neural Networks*, Vol. 12, No. 4 (2001) 929–935
- [5] Bishop, C.: *Neural networks in pattern recognition*. Oxford: Oxford University Press, (1995)
- [6] Black, F., Scholes, M.: The pricing of options and corporate liabilities. *Journal of Political Economy*, Vol. 81, (1973) 637–659
- [7] Buchen, P., Kelly, M.: Maximum entropy distribution of an asset inferred

- from option prices. *Journal of Finance and Quantitative Analysis*, Vol. 31, No. 1 (1996) 143–159
- [8] Cheung, Y. -M., Leung, W. -M., Xu, L.: Adaptive rival penalized competitive learning and combined linear predictor model for financial forecast and investment. *International Journal of Neural Systems*, Vol. 8, No. 5&6 (1997) 517–534
- [9] Choi, H. -J., Lee, H., Han, G. -S., Lee, J.: Efficient option pricing via a globally regularized neural network. *ISNN 2004, LNCS 3174* (2004) 988–993
- [10] Collin-Dufresne, P., Goldstein, R.: Do credit spreads reflect stationary leverage ratios?. *Journal of Finance*, Vol. 56, (2001) 1929–1957
- [11] Crouhy, M., Galai, D., Mark, R.: A comparative analysis of current credit risk models. *Journal of Banking and Finance*, Vol. 24, (2000) 59–117
- [12] Duan, J. C.: Maximum likelihood estimation using price data of the derivative contract. *Mathematical Finance*, Vol. 4, (1994) 155–167
- [13] Duan, J. C.: Correction: maximum likelihood estimation using price data of the derivative contract. *Mathematical Finance*, Vol. 10, (2000) 461–462
- [14] Duan, J. C., Gauthier, G., Simonato, J. G., Zaanoun, S.: Estimating Merton's model by maximum likelihood with survivorship consideration. Working paper, (2003)
- [15] Duffie, D., Lando, D.: Term structure of credit spreads with incomplete accounting information. *Econometrica*, Vol. 69, (2001) 633–664
- [16] Edelman, D.: Local cross-entropy. *Risk*, (2004)

- [17] Eom, Y. H., Helwege, J., Huang, J. -Z.: Structural models of corporate bond pricing: an empirical analysis. *Review of Financial Studies*, Vol. 17, No. 2 (2004) 499–544
- [18] Ericsson, J., Reneby, J.: The valuation of corporate liabilities: Theory and tests. Working paper, McGill University and Stockholm School of Economics, (2001)
- [19] Galindo-Flores, J.: A framework for comparative analysis of statistical and machine learning methods: an application to the Black-Scholes option pricing equation. *Computational Finance*, (1999)
- [20] Geske, R.: The valuation of corporate liabilities as compound options. *Journal of Financial and Quantitative Analysis*, Vol. 12, (1977) 541–552
- [21] Jang, J., Sun, C., Mizutani, E.: *Neuro-fuzzy and soft computing*. Prentice Hall, (1997)
- [22] Jones, E.P., Mason, S., Rosenfeld, E.: Contingent claims analysis of corporate capital structures: An empirical investigation. *Journal of Finance*, Vol. 39, (1984) 611–625
- [23] Kim, J., Ramaswamy, K., Sunderasan, S.: Does default risk in coupons affect the valuation of corporate bonds? A contingent claims model. *Financial Management*, (1993) 117–131
- [24] King, D., Khang, K.: On the cross-sectional and time-series relation between firm characteristics and corporate bond yield spreads. Working paper, University of Wisconsin-Milwaukee, (2002)

- [25] Lajbcygier, P.R., Jerome, T.C.: Improved option pricing using artificial neural networks and bootstrap methods. *International Journal of Neural Systems*, Vol. 8, No. 4 (1997) 457–471
- [26] Lang, M., Lundholm, R.: Cross-sectional determinants of analyst ratings of corporate disclosures. *Journal of Accounting Research*, Vol. 31, (1993) 246–271
- [27] Ljung, L.: *System identification: Theory for the user*. Second Edition, Prentice Hall, (1999)
- [28] Longstaff, F., Schwartz, E.: A simple approach to valuing risky fixed and floating rate debt. *Journal of Finance*, Vol. 50, (1995) 789–819
- [29] Lyden, S., Saraniti, D.: An empirical examination of the classical theory of corporate security valuation. Barclays Global Investors, (2000)
- [30] McQuown, J. A.: A comment on market vs. accounting based measures of default risk. KMV Corporation, (1993)
- [31] Merton, R.: On the pricing of corporate debt: The risk structure of interest rates. *Journal of Finance*, Vol. 29, (1974) 449–470
- [32] Mitchell, T.M.: *Machine Learning*. MIT: McGraw-Hill, (1997)
- [33] Moody's Investors Service: Moody's public firm risk model: A hybrid approach to modeling short term default risk. (2000)
- [34] Moody's KMV Corporation: *Modeling financial risk - modeling methodology*. (2003)

- [35] Piramuthu, S.: Feature selection for financial credit-risk evaluation decision. *Inform Journal of Computing*, Vol. 11, (1999) 258–266
- [36] Smolensky, P., Mozer, M.C., Rumelhart, D.E.: *Mathematical Perspectives on Neural Networks*. Lawrence Erlbaum Associates, NJ, (1996)
- [37] Vassalou, M., Xing, Y.: Default risk in equity returns. *Journal of Finance*, (2003)
- [38] Wei, D. G., Guo, D.: Pricing risky debt: An empirical comparison of the Longstaff and Schwartz, and Merton Models. *Journal of Fixed Income*, Vol. 7, (1997) 8–28
- [39] Wong, H. Y., Li, K. L.: On bias of testing Merton's model. Working paper, Chinese University of Hong Kong, (2004)
- [40] Xie, J. -G., Wang, J., Qiu, Z. -D.: Effectiveness of neural networks for prediction of corporate financial distress in China. ISSN 2004, LNCS 3174 (2004) 994–999
- [41] Xu, L.: Mining dependence structures from statistical learning perspective. *IDEAL 2002*, LNCS 2412 (2002) 285–306
- [42] Yu, F.: Accounting transparency and the term structure of credit spreads. *Journal of Financial Economics*, Vol. 75, (2005) 53–84

CUHK Libraries



004280633