INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

UMI®
800-521-0600

INTEGRATING MODELLING TECHNIQUES FOR FINANCIAL TIME SERIES

A Thesis

Presented to

The Faculty of Graduate Studies

of

The University of Guelph

by

VICTOR KHOON-LEE TAN

In partial fulfilment of requirements

for the degree of

Doctor of Philosophy

August, 1999

ABSTRACT

INTEGRATING MODELLING TECHNIQUES FOR FINANCIAL TIME SERIES

Victor Khoon-Lee Tan
University of Guelph, 1999

Supervisors:
Professor P. T. Kim and A. F. Desmond

The use of a related series of historical values as evidence or a guide line to predict and to draw conclusions about future events is a practice common to researchers of many disciplines. In order to achieve a *better* outcome, researchers are motivated to search for rules that can explain relationships between the past and future. However, any forecast is affected by the unpredictable nature of future events as well as by the limitations of past data. Moreover, a forecasting model used by one field may not necessarily be used by another, because there are problems which are unique to each field.

In standard statistical treatment of time series, time domain and frequency domain are the classical techniques used as a basis for characterising an observed system and forecasting its future behaviour. However, due to complicated data structures of time series, recent research developments–along with the availability of high-speed computers–have facilitated the development of modern nonparametric methods such as the *moving blocks bootstrap* and the *neural networks*.

The inadequacy of the efficient market hypothesis and the instability of data structures found in financial market create serious challenges to the implementation of modelling strategies for financial time series. However, a combination of both

classical techniques and modern nonparametric methods may offer a more effective solution to financial forecasting problems.

This thesis will implement a financial time series model which utilizes both classical techniques and nonparametric methods for estimation and inference. It will go on to propose a modified bivariate transfer function model. Both time domain and frequency domain techniques will be used in order to determine a relationship between bivariate time series. After generating the *less model dependent* samples using the moving blocks bootstrap, a neural network will be applied to achieve better point estimation. Integrating these techniques provides the best alternative solution to financial forecasting problems. The bivariate time series in question is interest rate spreads and the spot Canadian dollar. Simulation results showed undercoverage overall, but the integrated modelling technique appears robust to the choice of noise distribution as well as the sample size.

To: My Family

and

The Memory of My Grandparents and Parents

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to:

First and foremost, I would like to thank those professors on my supervisory committee for their time and dedication. I am grateful to my supervisor Dr. Peter Kim for his patience guidance, helpful suggestions and all the help since I came to Guelph. I would also like to thank Dr. Tony Desmond for his constructive criticism and suggestions. My appreciation is also extended to Dr. Thanasis Stengos and Dr. Qi Li for the honour of having both of them on my supervisory committee. In addition, I would like to thank the other professors who took time to answer any questions I may have had.

I would like to thank the Department of Mathematics and Statistics for all the physical and financial supports. My thanks are also extended to Linda Allen, John McCormick, departmental stuff and my fellow graduate students for their help and friendship.

During my stay at Guelph, I am fortunate to have Peter Yong, Siew-Hong Tan, Gary Rusland, Gary Cho, Dianqin Wang, and all others who are associated with me over the years as my friends. I would like to thank them for their friendship and for making my life more enjoyable.

A special thank to my lovely girlfriend, Sandy Pui-Yee Li, and her family for encouragement, support and love. Finally, I would like to thank my family for their endless support in everything!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The use of a related series of historical values (or past data collected) as evidence or a guide line to predict and to draw conclusions about future events, is a practice common to researchers of many disciplines. In order to achieve a 'better' outcome, researchers are motivated to search for rules that can explain relationships between the past and future. However, any forecast is affected by the unpredictable nature of future events as well as by the limitations of past data. Moreover, a forecasting model used by one field may not necessarily be used by another, because there are problems which are unique to each field. However, an awareness of the characteristics of the field specific models may allow the application of a translational model to obtain an adequate level of forecast accuracy.

In financial time series, the following two problems always exist. They are:

1. The inadequacy of the efficient market hypothesis; that is, market prices always reflect the available information set. However, in the actual market, deviations exist between *true* prices and *market* prices because not all traders have all the existing information, and may not always act rationally. Moreover, "the verification of the efficient market hypothesis (EMH) is impossible" Granger [27, p.12].

2. Instability of the data structure. The system may be influenced by major factors such as monetary policies, economic indicators, and interest rates. In addition, certain information cannot be modelled statistically, which leads to an unstable structure of past data sets.

These problems create serious challenges to the implementation of modelling strategies for financial time series.

In standard statistical treatment of time series, the first step is to 'look' at the data, and then select suitable mathematical models for the data. Before we can do this, we must first study and understand the techniques which have been developed for forecasting and drawing inferences from the series. Classical time series techniques are used as a basis for characterising an observed system and forecasting its future behaviour and, therefore, is the starting point for conventional statistical time series model building.

In the time series literature, classical time series analysis is based on the theory of stationary processes, and it has been demonstrated that most stationary processes can

be approximated by a model drawn from the class of *autoregressive moving average* (ARMA) models. In addition, some non-stationary processes may be represented by an ARMA process after differencing. This is known as the *autoregressive integrated moving average* (ARIMA) class of models, for which a model selection strategy has been developed by Box and Jenkins [8]. The significant contribution of Box and Jenkins to time series analysis is based on the invertible ARMA class of processes and the four phases of: identification, estimation, diagnostic checking, and forecasting. Particularly in the identification phase, plots of the sample autocorrelation function and the sample partial autocorrelation function can yield important information in identifying the orders p and q of an autoregression moving average (ARMA(p,q)) model. Because of the developments of Box and Jenkins [8], the problem of estimating the ARMA (p,q) model has attracted considerable attention in the time series literature.

The frequency domain analysis of time series has its own share in the time series literature. Frequency domain analysis is mathematically equivalent to time domain analysis, which is based on the autocovariance function, but provides an alternative way of viewing an observed system. In frequency domain analysis, "the spectral density function describes how the variation in a time series may be accounted for by cyclic components at different frequencies" Chatfield [10, p.7]. As a result, frequency domain analysis provides many important insights which would not be apparent in the time domain analysis.

"The spectral point of view is particularly advantageous in the analysis of multivariate stationary processes and in the analysis of very large data sets, for which

numerical calculations can be performed rapidly using the *fast Fourier transform*", Brockwell and Davis [9, p.112]. By combining the frequency domain technique and the transfer function model–as proposed by Box and Jenkins [8] in the time domain– we are able to exploit the relationship between any two financial time series.

Due to the complicated data structures of financial markets such as stock returns, exchange rates, and interest rates, there is a need to improve the performance of parameter estimates in time series. In addition, the behaviour of these financial time series could perhaps be described more accurately by non-linear mathematical models. Such major developments such as bootstrap techniques have been applied to problems from time series analysis. In the time series literature the use of bootstrap techniques can be found in Efron and Tibshirani [19], Freedman [23], Künsch [43] and Liu and Singh [46].

Recently, artificial neural networks (**ANNs**) have formed a basis for an entirely different non-linear approach to the analysis of time series. Together with the widespread availability of powerful computer hardware and efficient training algorithms, neural networks are now a subject of interest to professionals in almost every quantitative field and there is increasing focus on the potential benefits that neural networks can offer. Neural networks have been applied to a wide variety of problems, such as pattern recognition, classification, process control, and prediction. Recent reviews from a statistical perspective include: Cheng and Titterington [11], Ripley [51] and White [55] [56]. Weigend and Gershenfeld [54] provide a computer scientist's perspective of neural networks in time series. In addition, Cybenko [13], Funanhashi [24], Gallant and White [25] [26], Hecht-Nielsen [31], and Hornik, Stinchcombe and White [32]

[33] provide some important references on the existence properties of the feedforward networks.

1.2 Scope of The Thesis

The main objective of this thesis is to model financial time series. After a review of classical time series approaches to time domain and frequency domain approaches, the thesis will examine the moving blocks bootstrap and neural networks before going on to integrate these techniques to form an integrated modelling technique for the analysis of financial time series.

Chapter 2 will discuss classical and modern nonparametric methodologies to time series. In Section 2.1, time and frequency domain techniques are reviewed as these techniques are used (in a later section of the thesis) to determine the existence of a relationship between financial time series, and to develop a bivariate transfer function model. In Section 2.2, nonparametric methods–the moving blocks bootstrap and neural networks–are discussed in detail. These methods, together with classical techniques, provide the theoretical basis for the derivation of an integrated technique.

In Chapter 3, an integrated modelling technique will be developed to analyse financial time series. The proposed method offers an improved solution to financial forecasting problems.

Chapter 4 presents an application of the integrated modelling technique to financial time series, interest rate spreads and spot Canadian dollar data sets. The empirical results support the use of the integrated modelling technique to financial time series.

The integrated modelling technique will be applied to simulated time series data in Chapter 5, which will also summarise and present the results of the simulation. The simulation results show undercoverage overall, but the integrated modelling technique appears robust to the choice of noise distribution as well as the sample size. Conclusions and suggestions for future work will be presented in Chapter 6.

Chapter 2

Time Series Methodologies

Time series analysis is an important technique in a wide variety of quantitative fields. Over the years, a number of quantitative techniques and nonparametric methods have been developed for analysing and predicting time series values (essential references can be found in Chatfield [10]).

2.1 Classical Time Series Methods

To simplify the review of classical methods, we will not review time domain techniques here. Adequate references about the autoregressive integrated moving average (ARIMA) models, known as the Box and Jenkins approach, can be found in studies by Brockwell and Davis [9], Chatfield [10], and Priestley [48], which provide readable introductions to time series analysis and cover both time domain and

frequency domain techniques. In this section, a brief review of frequency domain techniques is presented.

2.1.1 Spectral Analysis

"Spectral analysis is essentially a modification of Fourier analysis so as to make it suitable for stochastic rather than deterministic functions of time" Chatfield [10, p.105]. Hence, spectral analysis of stationary time series involves decomposing the given time series into a linear combination of sinusoids.

For a given time series, Z_t, $t = 0, \pm 1, \pm 2, \ldots$, with $E(Z_t) = 0$, the frequencies

$$\nu_k = \frac{k}{n} \in \left[-\frac{1}{2}, \frac{1}{2}\right] , \qquad k = -\left[\frac{(n-1)}{2}\right], \ldots , \left[\frac{n}{2}\right] , \tag{2.1}$$

are called the *Fourier frequencies* of Z_t and $[\cdot]$ denotes the greatest integer function. Notice that k contains n integers. For example, at frequency k, a *waveform* is the sinusoidal function

$$\cos(2\pi\nu_k t) + \sin(2\pi\nu_k t) , \qquad t = 1, \ldots , n \tag{2.2}$$

which constitutes cycles over $t = 1, \ldots , n$, where $k = -[\frac{(n-1)}{2}], \ldots , [\frac{n}{2}]$. Since the length of time required to complete one cycle is called the *period*, the frequency ν_k measures the number of cycles per time point t. Note that, by knowing the Fourier

frequencies, we can capture the time path of Z_t as a weighted sum of the sinusoids in (2.2).

In order to estimate the contribution of each sinusoid (2.2), we estimate the autocovariances; $\gamma_Z(k) = Cov(Z_{t+k}, Z_t)$, by

$$c_Z(k) = \frac{1}{n}\sum_{t=1}^{n-|k|}(Z_{t+|k|} - \bar{Z})(Z_t - \bar{Z}) \tag{2.3}$$

for $k = -n+1, \ldots, n-1$ and $\bar{Z} = (\frac{1}{n}\sum_t Z_t)$. Note that $c_Z(0)$ is an estimate of the variance of Z_t. Following this we define the *periodogram*

$$I_n(\nu_k) = c_Z(0) + 2\sum_{k=1}^{n-1} c_Z(k)\cos(2\pi\nu_k k)\,. \tag{2.4}$$

The periodogram appears to be a natural way of estimating the *spectral density function* (or the *power spectrum*)

$$f_Z(\nu) = \gamma_Z(0) + 2\sum_{k=1}^{\infty}\gamma_Z(k)\cos(2\pi k\nu) \tag{2.5}$$

where

$$\gamma_Z(k) = \int_{-\frac{1}{2}}^{\frac{1}{2}} f_Z(\nu)\cos(2\pi k\nu)d\nu\,.$$

Moreover, the total area under the periodogram is an estimate of the variance of Z_t; that is,

$$\sum_{t=1}^{n}(Z_t - \bar{Z})^2 = \sum_{k=-[\frac{(n-1)}{2}]}^{[\frac{n}{2}]} I_n(\nu_k)\,. \tag{2.6}$$

Although the periodogram is asymptotically unbiased; that is

$$E\left[I_n(\nu_k)\right] \to f_Z(\nu)\ , \quad \text{as } n \to \infty \tag{2.7}$$

the variance of $I_n(\nu_k)$ is not zero as $n \to \infty$. Hence $I_n(\nu_k)$ is not a consistent estimator for $f_Z(\nu)$. Thus, in order to restore consistency, we will consider an alternative procedure of estimating a power spectrum–that is, the method of smoothing the periodogram. This procedure, which provides a consistent estimate of the spectral density function, is called a *lag window estimator*, defined by

$$\hat{f}_Z(\nu) \;=\; c_Z(0) \;+\; 2\sum_{k=1}^{m} w\left(\frac{k}{m}\right) c_Z(k)\cos(2\pi k\nu)\ , \qquad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right]\ , \tag{2.8}$$

where the *lag window*, $w(\cdot)$, is a weight function on [0,1] with $0 \le w(x) \le w(0) = 1$, and a sequence, m, is chosen such that $\frac{n}{m} \to \infty$ as $n \to \infty$. This method eliminates the inconsistency of the periodogram without destroying its asymptotic unbiasedness. Moreover, it allows us to examine the sources of variability in the total variation of a given time series. However, the choice of m is important because "the larger the value of m the smaller will be the variance of the resulting estimate but the larger will be the bias, and if m is too large then interesting features of $f_Z(\nu)$, such as peaks, may be smoothed out" Chatfield [10, p.117]. Since there is no effective method for making the choice of m, we can try several values in the region of $\frac{n}{40}$ according to Chatfield [10]. The most basic lag window estimator is the uniformly weighted average. In this thesis, the modified Daniell smoothing function, which is defined below, will be used

for data analysis

$$w(x_i) = \frac{1}{(span-1)}\left\{\frac{1}{2}\cdot x[i-\frac{(span-1)}{2}] + x[i-\frac{(span-1)}{2}+1] + \cdots + x[i+\frac{(span-1)}{2}-1] + \frac{1}{2}\cdot x[i+\frac{(span-1)}{2}]\right\}, \quad (2.9)$$

where $span$ $(= 2m+1)$ is an integer giving the length of the smoothing window for the given series. Note that $w(x_i)$ in (2.9) is the computational expression of $w(\frac{k}{m})$ in (2.20). Moreover, the smoothing window or average only considers the m neighboring values and x_i in smoothing x_i; the smoothing window actually includes span points with half-weights on the ends.

If Z_t is a generic *white noise* series which is defined to be an uncorrelated series, then we have $\gamma_Z(0) = \sigma_Z^2$, and $\gamma_Z(k) = 0\ \forall k = \pm 1, \pm 2, \ldots$. Moreover, we have a flat spectral density because $f_Z(\nu) = \sigma_Z^2$ for $\nu \in \left[-\frac{1}{2}, \frac{1}{2}\right]$. In addition, we have $F_Z(\nu) = \nu$ for all $\nu \in \left[0, \frac{1}{2}\right]$ hence define the *normalised spectral measure* by

$$F_Z(\nu) = \frac{\int_0^\nu f_Z(\nu)d\nu}{\int_0^{\frac{1}{2}} f_Z(\nu)d\nu}, \qquad \nu \in \left[0, \frac{1}{2}\right]. \quad (2.10)$$

By defining the sample version of equation (2.10) in the following way

$$U_n(\nu) = \frac{\sum_{\nu_k \leq \nu} I_n(\nu_k)}{\sum_{\nu_k} I_n(\nu_k)} \qquad \nu \in \left[0, \frac{1}{2}\right], \quad (2.11)$$

we can use $U_n(\nu)$ to test whether or not a time series is white noise. That is, we plot the $U_n(\nu)$ function and check its compatibility with the diagonal line (uniform

distribution function; $F(x) = x, 0 \leq x \leq 1$) using the Kolmogorov Smirnov test. Moreover, for $n > 62$ (see Brockwell and Davis [9]), a good approximation to a level α Kolmogorov Smirnov test is to reject the null hypothesis if the statistic falls outside of the bounds.

2.1.2 Cross Spectra Coherency

Thus far, we have been concerned with analysing a univariate time series. We will now examine the relationship between two time series. For two given non-white noise time series, Y_t and X_t, we use the *cross spectrum* to establish whether a structural relationship exists between them. The cross spectral density is defined as

$$f_{YX}(\nu) = \sum_{k=-\infty}^{\infty} \gamma_{YX}(k) e^{-2\pi i \nu k}, \qquad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right], \tag{2.12}$$

where

$$\gamma_{YX}(k) = Cov(Y_{t+k}, X_t), \qquad k = 0, \pm 1, \pm 2, \ldots \tag{2.13}$$

is called the *cross covariance* and its inverse representation is defined as

$$\gamma_{YX}(k) = \int_{-\frac{1}{2}}^{\frac{1}{2}} f_{YX}(\nu) e^{2\pi i \nu k}, \qquad k = 0, \pm 1, \pm 2, \ldots . \tag{2.14}$$

Note that $f_{YX}(\nu)$ is a complex valued function. The reason is that $\gamma_{YX}(k)$ (2.13) is not an even function; that is, $\gamma_{YX}(k) \neq \gamma_{YX}(-k)$.

The *cross-coherence function* between two series is defined by

$$K(\nu) = \frac{f_{YX}(\nu)}{[f_X(\nu) f_Y(\nu)]^{\frac{1}{2}}}, \quad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right] , \tag{2.15}$$

and the Cauchy - Schwarz inequality ensures that *squared coherence*, $K^2(\nu)$, is bounded as follows:

$$0 \leq K^2(\nu) \leq 1, \quad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right] . \tag{2.16}$$

Here $K^2(\nu)$ can be interpreted as the spectral equivalent of the ordinary linear regression R^2 statistic. Note that (2.16) measures the extent to which Y_t and X_t series are correlated at each frequency. Two situations of particular interest arise:

Case I: When Y_t series is an exact linear function of X_t series, we have

$$f_Y(\nu) = |A(\nu)|^2 f_X(\nu) \text{ and } f_{XY}(\nu) = A(\nu) f_X(\nu), \quad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right] , \tag{2.17}$$

where

$$A(\nu) = \sum_{h=-\infty}^{\infty} \alpha_t e^{-2\pi i \nu h}, \quad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right] \tag{2.18}$$

is called the *frequency response function.*

Case II: When no correlation between Y_t and X_t series exists, we have

$$f_{XY}(\nu) = 0 \quad \forall \quad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right] . \tag{2.19}$$

Therefore, by the definition of squared coherence, we have

$$|K(\nu)|^2 = 1, \quad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right], \qquad \text{for case I ,}$$

and

$$|K(\nu)|^2 = 0, \quad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right], \qquad \text{for case II .}$$

These are the two extremes that can happen to the Y_t and X_t series. However, in case II, no further analysis is required, as no relationship between Y_t and X_t series exists.

To test whether or not the Y_t and X_t series are correlated, we use

$$\hat{f}_Z(\nu) \;=\; c_Z(0) \;+\; 2\sum_{k=1}^{m} w\left(\frac{k}{m}\right) c_Z(k)\cos(2\pi k\nu) , \qquad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right] , \tag{2.20}$$

as well as

$$\hat{f}_{YX}(\nu) = \sum_{|k|\le m} c_{YX}(k) w\left(\frac{|k|}{m}\right) e^{-2\pi i\nu k}, \quad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right], \tag{2.21}$$

where $w(\cdot)$ is a weight function supported on the interval [0,1] with $0 \le w(x) \le$

$w(0) = 1$, and m is chosen to be a sequence such that $\frac{n}{m} \to \infty$ as $n \to \infty$ with

$$c_{YX}(k) = \begin{cases} \frac{1}{n}\sum_{t=1}^{n-k}(Y_{t+k} - \bar{Y})(X_t - \bar{X}) & \text{for} \quad 0 \le k \le n-1\,, \\ \\ \frac{1}{n}\sum_{t=-k+1}^{n}(Y_{t+k} - \bar{Y})(X_t - \bar{X}) & \text{for} \quad -n+1 \le k \le 0\,, \end{cases}$$

and

$$\hat{K}^2(\nu) = \frac{|\hat{f}_{YX}(\nu)|^2}{\hat{f}_X(\nu)\hat{f}_Y(\nu)}, \quad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right]\,. \tag{2.22}$$

Asymptotically, we can test the hypothesis $H_o : K^2(\nu) = 0, \quad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right]$, and the alternative hypothesis $H_a : K^2(\nu) > 0$ using the statistic

$$\mathcal{S} = \frac{2m|\hat{K}(\nu)|^2}{1 - |\hat{K}(\nu)|^2}\,, \quad \nu \in \left[-\frac{1}{2}, \frac{1}{2}\right]\,. \tag{2.23}$$

Since $K^2(\nu)$ is distributed as the square of a multiple correlation coefficient, so that $\mathcal{S} \sim F_{2,4m}$ under the hypothesis that $K^2(\nu) = 0$, we can therefore reject the hypothesis $K^2(\nu) = 0$ if $\mathcal{S} > F_{1-\alpha,\,2,\,4m}$, where $F_{1-\alpha,\,2,\,4m}$ is the $(1-\alpha)$ quantile of the F distribution with 2 and $4m$ degrees of freedom. As we mentioned earlier, the choice of m can be important; it can also affect the F test as demonstrated here.

2.1.3 Bivariate Transfer Function Model

After establishing the existence of a structural relationship between the X_t and Y_t series, we have the following model :

$$Y_t = \sum_{s=m_1}^{m_2} \alpha_s X_{t-s} + N_t \,, \tag{2.24}$$

where $m_1 \leq m_2$ and N_t is a random noise which is assumed to be uncorrelated with X_t. Here, m_1 and m_2 have to be estimated. Note that if $m_1 \geq 0$, we assume the past X_t will influence future Y_t, but not *vice versa*. Thus, in the former case the model given in (2.24) would be called *causal*.

Having considered the model, we follow the procedures in Brockwell and Davis [9, pp.455–457] to estimate α_s, $s = m_1, \ldots, m_2$ and possibly the N_t process in model (2.24). The procedures will be presented in Chapter 3.

2.2 Nonparametric Methods

In this section, nonparametric methods will be discussed. Bootstrap techniques and neural networks are computer based (nonparametric) techniques. Although the two techniques use different computer algorithms, they provide an alternative approach to modern data analysis and are particularly useful in the analysis of complicated data structures.

2.2.1 Moving Blocks Bootstrap Sample

The bootstrap is a computer based resampling technique (as initiated by Efron [15]) used to gain information on the distribution of an estimator. A more general resampling technique, called the *moving blocks bootstrap* (see Künsch [43], Liu and Singh [46], and Léger, Politis and Romano [44], and Efron and Tibshirani [20]), is used for model dependent data. The procedure for obtaining the moving blocks bootstrap time series samples will be discussed. In the case where $n \neq k \cdot l$ (where k is the number of blocks and l is the block size), it is proposed that a modified moving blocks bootstrap be used to generate the bootstrap time series.

Simple bootstrap resampling involves fitting a model and then sampling from the residuals. An alternative method is the moving blocks bootstrap, for bootstrapping time series: For any given time series sample, we can generate a bootstrap time series. In order to do so we must (1) choose a block length; (2) form the time points to all possible contiguous blocks of chosen length; (3) draw the samples with replacement from these blocks; and (4) join all the samples together to form a bootstrap time series. These steps can be applied to generate a bivariate bootstrap time series–that is, we pair-up the original bivariate time series and then apply the four steps to the series. For example, if we have samples of size n, and the block length is chosen to be l, then we can have k blocks to form the bootstrap time series. The number of blocks, k, is chosen so that $n \approx k \cdot l$. Figure 2.1 shows the bootstrap samples for a bivariate

time series with block length $l = 3$. "The idea for the moving blocks bootstrap is to choose a block size l large enough that observations more than l time units apart will be nearly independent" Efron and Tibshirani [20, p.102]. However, Künsch [43] and Liu and Singh [46] show that $B_i = (X_i, X_{i+1}, \ldots, X_{i+l-1})$, $i = 1, 2, \ldots, k$, are *i.i.d.* for fixed l under some conditions. Note that the moving blocks bootstrap coincides with the classical *i.i.d* bootstrap of Efron [15] when $l = 1$. The advantage of using the moving blocks bootstrap is that we will not destroy the correlation that we are trying to capture since we are resampling from the blocks of observations. Next, we propose a modified moving blocks bootstrap when $k \cdot l \neq n$.

2.2.2 Modified Moving Blocks Bootstrap

We note that if $k \cdot l \neq n$; that is, when n is not exactly divisible by l, then Efron and Tibshirani [20] suggest that we have to multiply the bootstrap standard errors by $\sqrt{\frac{k \cdot l}{n}}$ to adjust for the difference in lengths of the series. We propose an additional step to the moving blocks bootstrap procedure; that is, the addition of a random block with the block size $l' = n - (k \cdot l)$ to the bootstrap time series, in order to resolve this adjustment problem.

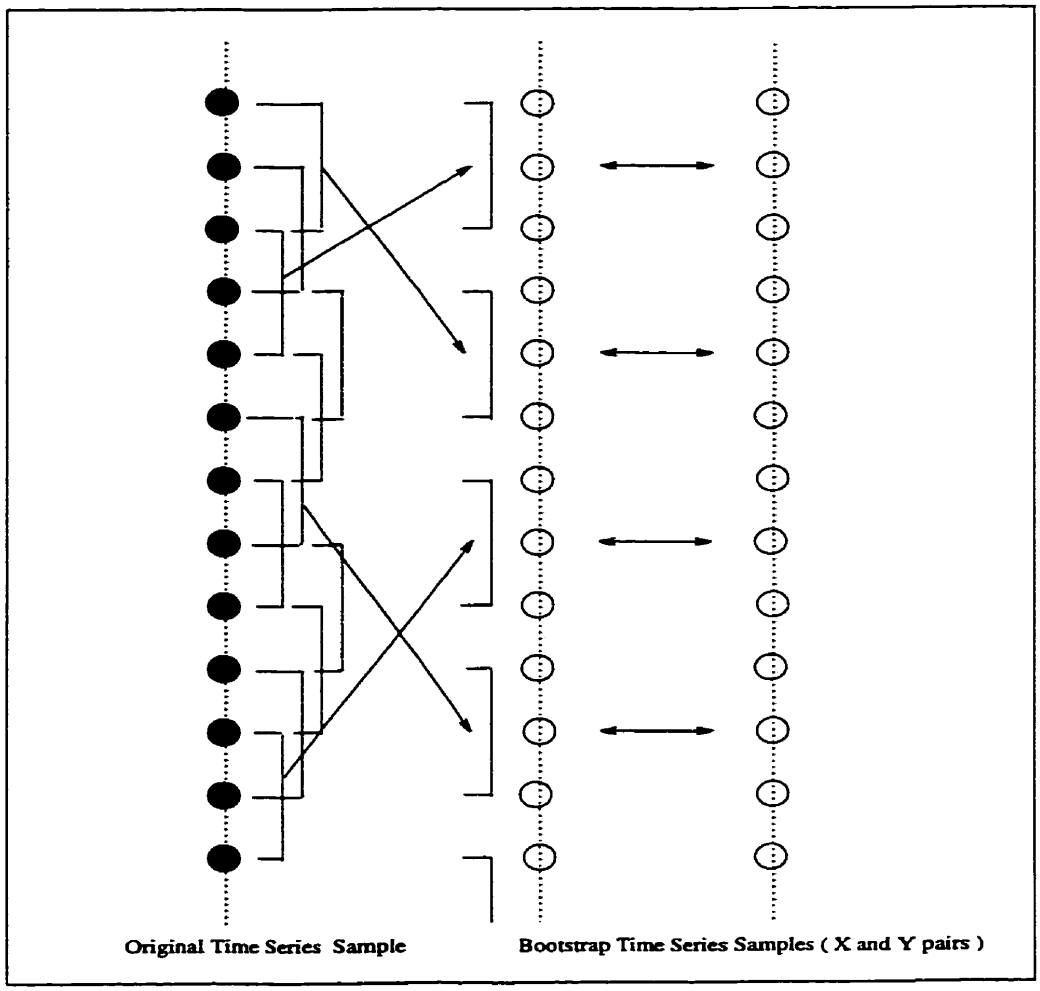


Figure 2.1: A schematic diagram of the moving blocks bootstrap for time series (Modified from Efron and Tibshirani [21]).

2.2.3 Neural Networks

Artificial neural networks (ANNs) are mathematical models for human brain activities. In general, neural networks are often applied to applications such as classification, pattern learning, and prediction.

Neural networks are considered a nonparametric method for drawing statistical inferences. The important aspect of neural networks is their capability to construct nonlinear relationship between the given input and output data. When used together with powerful computer hardware and training algorithms, ANNs allow statistical inference to be made without any structural formula. This implies that nonlinear time series are more suitable for analysis by neural networks. Because time series are statistically more volatile in financial markets than in other applications, the financial industry has become a prime application area for neural networks.

Although there are a numerous of networks of which many are recurrent neural networks (e.g. Kuan and Liu [42]), the multilayer perceptrons is the simplest neural network representation, and it can well approximate a large class of functions. Hence, in this thesis, the feed-forward networks are used as the only nonparametric model. Authors such as Cybenko [13], Funanhashi [24], Gallant and White [25] [26], Hecht-Nielsen [31], and Hornik, Stinchcombe and White [32] [33] provide some important references on the existence properties of the feedforward networks. Recent reviews from a statistical perspective include Cheng and Titterington [11], Ripley [51], and White [55] [56]. Weigend and Gershenfeld [54] present a computer scientist's perspective of Neural networks in time series.

2.2.4 Feed-forward Networks : Multilayer Perceptrons

Feed-forward networks are an important class of neural networks. Typically, the network consists of a set of source nodes that constitute the *input layer* of one or more *hidden layers* of computation nodes, and an *output layer* of computation nodes. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. These feed-forward networks are commonly referred to as *mulitlayer perceptrons* (MLP) when there is at least one hidden layer. From the statistical point of view, the multilayer perceptron networks provide *nonlinear regression functions* that are estimated by optimising some measure of fit to the training data. Figure 2.2 shows a typical multilayer perceptron with only one hidden layer, flowing from the bottom up.

The multilayer perceptron network is a supervised neural network since, during training, each input data vector is paired with a corresponding desired target for the network output. The actual outputs in the output layer are now compared with their desired targets, the difference being the output errors. The errors are then fed back to the network to improve the performance. This process is known as the *backpropagation algorithm.* This algorithm is based on the *mean squared error* (MSE) as an *error-correction learning rule.* Basically, the back-propagation process consists of two passes: the *forward pass* and *backward pass.* In the forward pass, the input data vectors are applied to the source nodes and propagate, layer by layer, through

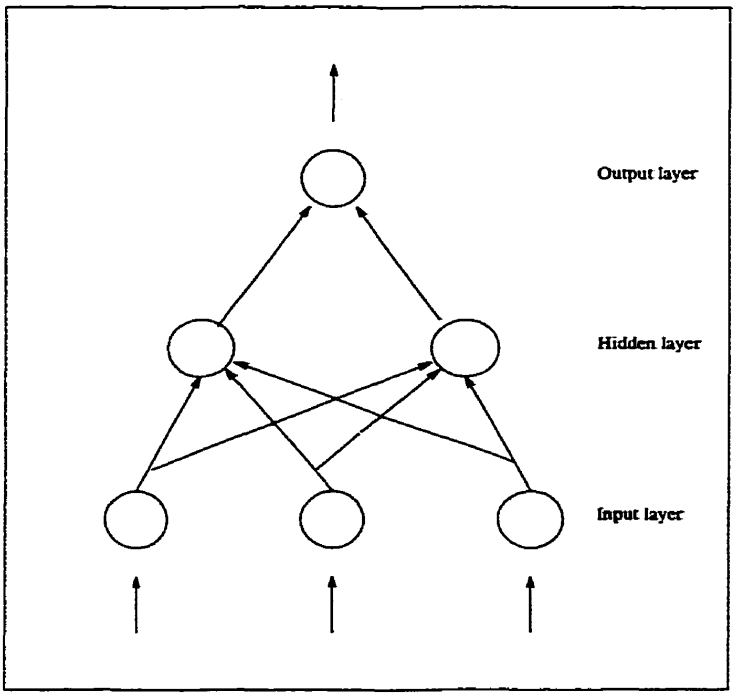


Figure 2.2: Three-Layer Feedforward Network.

the network. Once one forward pass is completed, a set of outputs is obtained as the actual output of the network. Note that the *synaptic weights* (which are the inter-node connection weights of the network) are fixed during the forward pass. During the backward pass, the synaptic weights of the network are all adjusted according to the error-correction rule. An *error signal* is then obtained by subtracting the actual output of the network from the desired target. This updated error signal is then propagated backward through the network; that is, in the opposite direction of synaptic connections. The synaptic weights are adjusted to improve the performance until a zero, or minimum, error signal is obtained. The learning process performed by this algorithm is called *backpropagation learning.* A complete presentation *cycle*, or *iteration*, comprises the forward pass and backward pass cycles. One complete presentation of the entire training set is called an *epoch.* The learning process is

maintained on an epoch by epoch basis, until the synaptic weights and the threshold levels of the network stabilise and the E_{MSE} over the entire training set converges to some minimum value. The E_{MSE} is defined by

$$E_{mse} = \frac{\sum_s \sum_o (Target - Output)^2}{N_o N_s}, \tag{2.25}$$

where N_s is the number of training vectors in one complete presentation of the entire training set during the learning process (an epoch), and N_o is the number of output neurons.

It is advisable to randomise the order of presentation of training sets from one epoch to the next in order to make the search in weight space stochastic over the learning cycles, and avoid the possibility of limit cycles in the evolution of the synaptic weight vectors. Moreover, the randomisation of the training sets allows for a testing technique that reflects the practical utilisation of the network in actual forecasting problems.

The *normalised mean squared error* (NMSE) is used to evaluate the model of the network, and is defined by

$$NMSE = \frac{\sum_{s=1}^{K} (Y_s - \hat{Y}_s)^2}{\sum_{s=1}^{K} Y_s^2}, \tag{2.26}$$

where K is the number of data in the validation set, Y_s is the desired target, and $\hat{Y}_s$ is the actual output. Note that the quantity (1− NMSE) measures the percentage variance in the data which is explained by the model. Moreover, if we plot the *learning*

curve (the NMSE versus the epoch) for the function estimated from the training set, we can easily see that the *learning curve* is monotonically decreasing.

The essence of backpropagation learning is to encode an input-output relation. A multilayer perceptron will be considered well trained if it learns enough about the past to be able to generalize about the future. However, if we allow the learning process to proceed indefinitely, we will overfit the data and begin modelling only noise–that is, the NMSE reduces to zero. The cause of this problem is understandable if we consider the linear regression case: if we constantly add parameters to its full rank, the R-square becomes 1. Hence, we need a stopping criteria in order to end the learning process when the generalization performance is adequate.

Kim, Martin and Staley [39] identify the technique most commonly used to end the learning process: First, the available data set (or the moving blocks bootstrap time series samples) is randomly partitioned into a training set and a validation set (there are no fixed rules for selecting the amount of data for either the training set or the validation set). We propose to select 75% of the data to be used for the training set, and 25% of the data to be used for the validation set. The motivation here is to *validate* the model on a data set which is different from the one used for parameter estimation. In this way, we can compute the NMSE (2.26) based on a validation set using the resulting network used on the training set. Consequently, we plot the learning curve of the validation set, and look for the epoch at the point on the curve where the direction starts to change. Note that the learning curve of the validation set will not necessarily monotonically decrease. This number of epochs will then be used to determine the stopping point for the learning process. It should be noted that

this situation does not always occur, but if it does happen, the network should be re-trained on the validation set. Finally, the configuration of the resulting multilayer perceptron network on the given training set and validation set is used for prediction or forecasting on the given *test set.*

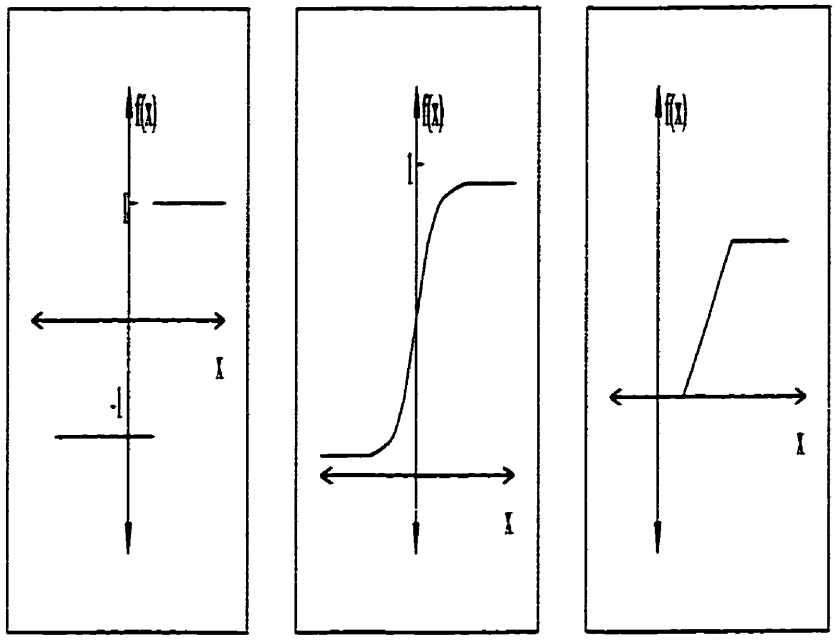
In neural network architecture, Figure 2.3 shows the commonly used *threshold functions.* The threshold function is used in the forward pass computation. Using the architecture of Figure 2.2 as an example, we process input $\boldsymbol{X}$'s to calculate new hidden layer i^{th} neuron using

$$b_i = f(\sum_{h=1}^{k} x_h \nu_{hi} + \Phi_i)\,, \tag{2.27}$$

where $f(\cdot)$ is the threshold function, ν_{hi} is the weight between the input layer and hidden layer, and Φ_i is the hidden layer threshold values. Note that the backpropagation algorithm is presented in Appendix A.

Remarks : A multilayer perceptron has the following distinctive characteristics :

1. "The model of each neuron in the network includes a *nonlinearity* at the output end. The important point of emphasis here is that the nonlinearity is *smooth*; that is, differentiable everywhere" Haykin [30, p.138]. In the prediction context, a nonlinear function that satisfies this requirement is a symmetric *sigmoidal*



(a) Step function (b) Sigmoid function (c) Ramp function

Figure 2.3: Threshold Functions for Neural Networks.

function defined by the *logistic function* :

$$Y = \frac{1}{1+\exp(-x)}, \tag{2.28}$$

$$f' = (1-Y)Y, \tag{2.29}$$

where f' is the derivative of the logistic function. Note that the logistic function is a commonly used function which transforms the input value in the range $x \in [-\infty, +\infty]$ into the output range $O \in [0, 1]$. "The presence of nonlinearities is important because, otherwise, the input - output relation of the network could be reduced to that of a single-layer perceptron" Haykin [30, p.139].

2. For the choice of the maximum number of layers in the network, Lippmann [45] recommends that three layer networks, with just one hidden layer, are sufficient to solve arbitrarily complex input-output mappings.

3. The hidden neurons which are not part of the input or output of the network enable the network to learn complicated data by extracting more information or features from the input vectors.

4. The network is connected between the layers by the synapses of the network. Hence, a change in the connectivity of the network requires a change in synaptic weights of the network.

Chapter 3

Integrated Modelling

In this chapter, a strategy called *integrated modelling* will be implemented in order to analyse time series based on financial data. Since the modelling tool is a combination of the methodologies described in the previous chapters, the term *integrated modelling* is used. The objective for implementing this technique is to achieve a more reliable solution to the modelling and forecasting problems which occur in the financial world. The technique utilizes both classical techniques and nonparametric methods for estimation and inferences. This chapter will begin with an examination of the theoretical foundations for integrated modelling.

3.1 Theoretical Foundations for Integrated Modelling.

In Chapter 2, the theoretical background for the derivation of classical techniques and nonparametric methods is discussed. These techniques and methods form the basis of implementing techniques in modelling time series. In this section, with the

support of these techniques and methods, we present the theoretical foundations in detail for the derivation of the integrated modelling technique.

An initial approach is to follow the method of Kim and Martin [40], using spectral analysis to uncover certain statistical details. For a given bivariate time series, the statistic $U_n(\nu)$ (see Chapter 2) is used to determine whether the two time series are white noise. That is, the Kolmogorov Smirnov test is used to reject the null hypothesis with level α if the function $U_n(\nu)$ falls outside of the bounds for each time series. Once we reject the null hypothesis and conclude that the two given time series are not white noise, the cross spectrum is applied to determine the existence of the relationship between the two series.

After establishing the existence of a structural relationship between the two series, the proposed model (which is modified from the bivariate transfer function model) is applied to further analyse the time series. Before implementing the proposed model, it is helpful to provide the rationale for modifying the bivariate transfer function model. The bivariate transfer function model is restated below:

$$Y_t = \sum_{s=m_1}^{m_2} \alpha_s X_{t-s} + N_t \,. \tag{3.1}$$

Recall that (3.1) is valid only if it is causal, that is, the past X_t will influence future Y_t, but not *vice versa*. In particular, the model requires that $m_1 \leq m_2$ and $m_1 \geq 0$. Next, we follow the procedures listed in Brockwell and Davis [9, pp.455–457] in order to estimate m_1, m_2, and α_s, $s = m_1, \ldots, m_2$. The following steps are applied to obtain $\hat{m}_1$ and $\hat{m}_2$:

Step 1. We calculate the cross correlation function (**CCF**); $\hat{\rho}_{W_t Y_t^*}(h)$ of the two filtered series; namely W_t and Y_t^*. The filtered series W_t and Y_t^* are obtained by applying the prewhitening techniques to X_t and Y_t series, respectively. This is achieved by transforming the X_t series to a white noise series by

$$\hat{\phi}(B)\hat{\theta}^{-1}(B)X_t = W_t\,, \tag{3.2}$$

and applying the same transformation; that is, $\hat{\phi}(B)\hat{\theta}^{-1}(B)$, to the Y_t series to obtain

$$Y_t^* = \hat{\phi}(B)\hat{\theta}^{-1}(B)Y_t\,. \tag{3.3}$$

Step 2. We performed an asymptotic test of no cross correlation at the 5% level using the bounds $\pm 1.96n^{-\frac{1}{2}}$. The comparison of $\hat{\rho}_{W_t Y_t^*}(h)$ with the bounds $\pm 1.96n^{-\frac{1}{2}}$ gives an indication of the lags h at which $\hat{\rho}_{W_t Y_t^*}(h)$ is significantly different from zero.

Step 3. From the previous step, we estimate $\hat{m}_1$ and $\hat{m}_2$ to be the minimum and maximum of lags h at which $\hat{\rho}_{W_t Y_t^*}(h) \neq 0$, respectively.

To continue the process of estimating α_s, $s = m_1, \ldots, m_2$ for the bivariate transfer function model, we apply the filter $\hat{\phi}(B)\hat{\theta}^{-1}(B)$ to the model (3.1), which yields

$$Y_t^* = \sum_{s=m_1}^{m_2} \alpha_s W_{t-s} + N_t^* \ . \tag{3.4}$$

This implies that the transfer function between X_t and Y_t series is the same as that between W_t and Y_t^*. By multiplying both sides of (3.4) by W_{t-h}, and taking expectation, we obtain

$$E(W_{t-k}\, Y_t^*) = \{\alpha_0 E\,(W_{t-k}W_t) + \alpha_1 E\,(W_{t-k}W_{t-1}) + \cdots\} + E(W_{t-k}N_t^*) \ . \tag{3.5}$$

Since

$$\begin{aligned} E(W_{t-k}Y_t^*) &= \gamma_{W_t Y_t^*}(k) \\ E(W_{t-k}W_{t-j}) &= 0 \quad \text{for} \quad j \neq k \end{aligned}$$

and $E(W_{t-k}N_t^*) = 0, \ \forall\, k$ (because X_t and N_t are uncorrelated). Therefore,

$$\gamma_{W_t Y_t^*}(k) = \alpha_k \sigma_{W_t}^2 \tag{3.6}$$

and

$$\begin{aligned}\alpha_k &= \frac{\gamma_{W_t Y_t^*}}{\sigma_{W_t}^2} \\ &= \frac{\sigma_{Y_t^*}}{\sigma_{W_t}} \frac{\gamma_{W_t Y_t^*}(k)}{\sigma_{Y_t^*}\sigma_{W_t}} \\ &= \frac{\sigma_{Y_t^*}}{\sigma_{W_t}} \rho_{W_t Y_t^*}(k), \quad k = 0, 1, 2, \ldots\end{aligned}$$

The estimates of α for lag h at which $\hat{\rho}_{W_t Y_t^*}(h)$ is found to be significantly different from zero are

$$\hat{\alpha}_h = \hat{\rho}_{W_t Y_t^*} \frac{\hat{\sigma}_{Y_t^*}}{\hat{\sigma}_{W_t}}, \quad h = \hat{m}_1, \ldots, \hat{m}_2. \tag{3.7}$$

Now that we understand the frame work of the bivariate transfer function model, we present our proposed model for bivariate time series.

The first step in integrated modelling is a modification of (3.1) as follows:

$$Y_t = \sum_{s=0}^{p} w_s(X_{t-s} + \beta_{t-s}) + N_t, \tag{3.8}$$

where N_t is random noise assumed to be uncorrelated with X_t, and β_t is defined by

$$\beta_t = \begin{cases} \kappa & \text{if } X_{t-1} > X_{t-2} \\ 0 & \text{if } X_{t-1} = X_{t-2} \\ \lambda & \text{if } X_{t-1} < X_{t-2} \end{cases}$$

where β's, κ, and λ are unknown parameters. Note that we fix the term p to be 4 in (3.8) for daily closing data and, for simplicity we let $\tilde{X}_t = X_t + \beta_t$. In addition, the coefficient w_s will be estimated by using the *maximum likelihood estimate* (MLE).

Furthermore, we introduce two constant terms, namely κ and λ, in (3.8). The reason for this is that we believe that the increase (decrease) in previous time points X_{t-1} will result in an increase (decrease) in present time point X_t. The estimates of κ and λ are defined as follow:

$$\hat{\kappa} = \text{average of} \left\{ \sum_{t=2}^{n} (X_t - X_{t-1} | X_t > X_{t-1}) \right\} ,$$

and

$$\hat{\lambda} = \text{average of} \left\{ \sum_{t=2}^{n} (X_t - X_{t-1} | X_t < X_{t-1}) \right\} .$$

So far, we have only used the classical techniques with modifications. The next step in integrated modelling is to incorporate the nonparametric methods. We generate bootstrap time series samples using the moving blocks bootstrap method for drawing samples. It is important to note that we do not apply the moving blocks bootstrap method to the residuals. The reason for this is that the bootstrapping of residuals method is "model dependent", that is, the model, for example an AR(1) model, is fit to the original time series.

Next, we apply the multilayer perceptron network to the original bivariate time series and the bootstrap generated bivariate time series samples for estimation and inference. These applications complete the derivation of the theoretical foundations for integrated modelling where the procedures of the nonparametric methods introduced in Chapter 2 are followed closely.

3.2 Modelling Procedures

After deriving the theoretical foundations for integrating a model, the modelling procedure is used to analyse bivariate financial time series.

The correct approach for analysing bivariate financial time series is to use spectral analysis in order to uncover certain statistical details. In particular, we need to determine the existence of the relationship between the two given time series. If it is determined that a relationship exists, we continue to apply the integrated modelling technique to further the analysis of the bivariate time series. Having defined the model structure, the moving blocks bootstrap is applied to the given time series. However, we need to know the block size l in order to generate the bootstrap time series samples. Since there is no effective way of choosing the block size at this moment, we suggest an algorithm for choosing an appropriate block size.

3.2.1 Algorithm for Choosing the Block Size

Step 1. We fit an ARMA(p,q) model to the stationary time series. We use the first coefficient and its associated standard error as true values. For example, we have the AR(1) model for the given series z_t, $t = 1, 2, \ldots, n$;

$$z_t = \hat{\beta} z_{t-1} + \epsilon \, .$$

We now choose $\hat{\beta}$ and $\hat{\sigma}_{\beta}$ as our true values.

Step 2. Here we fix the block size to be $l = 1, 5, 10, \ldots, 100$, and select $B = 200$ (Where $l < \frac{n}{10}$). For each l, we generate the independent bootstrap time series $z_t^{*b}, b = 1, 2, \ldots, B$, each consisting of n data values drawn with replacement from z_t.

Step 3. For each l, we fit the ARMA(p,q) model to each bootstrap time series $z_t^{*b}, b = 1, 2, \ldots, B$. In the example given, we fit an AR(1) model to each generated series and obtain the bootstrap coefficients of $\hat{\beta}^*(b), b = 1, 2, \ldots, B$.

Step 4. For each l, we estimate the standard error $\sigma_F(\hat{\beta})$ by the sample standard deviation of the B replications

$$\hat{\sigma}_B = \left\{ \sum_{b=1}^{B} \left[\hat{\beta}^*(b) - \hat{\beta}^*(\cdot) \right]^2 / [B-1] \right\}^{\frac{1}{2}},$$

where

$$\hat{\beta}^*(\cdot) = \sum_{b=1}^{B} \hat{\beta}^*(b)/B \,.$$

Step 5. Here we plot the standard errors obtained from Step (4) versus the block size l. Finally, we choose the block size l such that the standard error of l is closest to the true estimate obtained in Step (1).

3.2.2 Generating Bootstrap Bivariate Time Series.

Once the block size is chosen, the bootstrap time series samples are generated by using the following steps:

Step 1. Use the algorithm described above to select an appropriate block size l.

Step 2. Select B independent bootstrap samples $(X_t^{*1}, Y_t^{*1}), (X_t^{*2}, Y_t^{*2}), \cdots, (X_t^{*B}, Y_t^{*B})$, each consisting of n data values drawn with replacement from (X_t, Y_t). Here we use the moving blocks bootstrap procedures to generate the bootstrap sample pairs. Note that we need to have $B \geq 1000$ for constructing the intervals (See Tibshirani [53]).

3.2.3 Neural Network Estimations

After obtaining the moving blocks bootstrap time series for each set of bootstrap sample pairs $(X_t^{*b}, Y_t^{*b}), b = 1, 2, \ldots, B$, we construct the multilayer perceptron network with 4 input layers, n hidden units for the hidden layer (where n is determined by trial and error in the region of $\frac{2}{3}$ of input vectors), and 1 output layer. The three-layer perceptron network with the backpropagation algorithm is now used to train on the input-output pair; that is, $((\tilde{X}_{t-4}, \tilde{X}_{t-3}, \tilde{X}_{t-2}, \tilde{X}_{t-1}),\ Y_t), t = 1, 2, \ldots, n$, where $\tilde{X}_{-3}, \tilde{X}_{-2}, \tilde{X}_{-1}$ and $\tilde{X}_0$ are set to zero. Since the underlying (2.28) activation function of the network is logistic, it would be better to choose the data that fits a range

that does not "saturate" the network neurons. Thus, we have the range of the inputs $\tilde{X}_t, t = 1, 2, \ldots, n$ to be $\tilde{X}_t \in [-1, +1]$, and the range of the output $Y_t, t = 1, 2, \ldots, n$ to be $Y_t \in [0, 1]$ (i.e., Sigmoid range).

Finally, we can construct the bootstrap forecast intervals for the financial time series. For example, if we want to have a forecast interval for one step ahead future value, Y_{n+1}, then the procedure is as follows:

Step 1. Use the neural network to obtain the forecasted values; $\hat{Y}^{*b}_{n+1}, b = 1, 2, \ldots, B$.

Step 2. We now construct the forecast intervals for Y_{n+1}. Letting

$$D^* = \hat{Y}_{n+1} - \hat{Y}^{*b}_{n+1}, \quad b = 1, 2, \ldots, B,$$

where $\hat{Y}_{n+1}$ is the forecasted future value obtained from the original time series, we approximate the bootstrap distribution of D^* by its empirical distribution.

Step 3. Let u^*_α be the α quantile of the bootstrap distribution of D^*. Then a bootstrap forecast interval is given by

$$\left[\hat{Y}_{n+1} - u^*_{1-\alpha}, \ \hat{Y}_{n+1} - u^*_\alpha\right].$$

Note that this is the bootstrap percentiles method applied to prediction. Other methods of bootstrap confidence intervals can be found in Efron and Tibshirani [20].

Chapter 4

Application to Financial Time Series

Based on the theoretical foundations discussed in Chapter 2, we derive the integrated model presented in Chapter 3. This chapter will apply the integrated model to a given set of data, namely the interest rate spread and the spot Canadian dollar series, in order to examine how the integrated model will perform.

4.1 The Interest Rate and Spot Canadian Dollar Data

The interest rate spread data (IS_t) and the spot Canadian dollar data (C_t) are taken from Kim and Martin [40]. A brief description of the data set is given below.

A total of 1476 daily trading data, commencing January 1991 through to August 1996, are used in this study. The interest rate spread IS_t (in percent) is defined as the difference between 90 day Canadian and 90 day US treasury bill rates, and the

spot Canadian dollar C_t is defined as the closing price of the exchange rate measured in US cents. The time plots of C_t and IS_t are presented in Figure 4.1.

Next, we define the changes in the interest rate spread IS_t by

$$X_t = IS_t - IS_{t-1}\,, \tag{4.1}$$

and define the changes in log C_t (or Return R_t) by

$$Y_t = \log C_t - \log C_{t-1}\,. \tag{4.2}$$

Note that the time unit for Y_t is one day. The time plots of Y_t and X_t series are presented in Figure 4.2A and 4.2C. It is apparent that the two series appear to be random quantities over time.

In Figure 4.2C however, the X_t series exhibits extreme movements, while the Y_t series (Figure 4.2A) is bounded within a smaller range over the time period. Moreover, since we are trying to identify the "stable" model for the long term link between the X_t and Y_t series, we will remove these extreme points and replace them as follows:

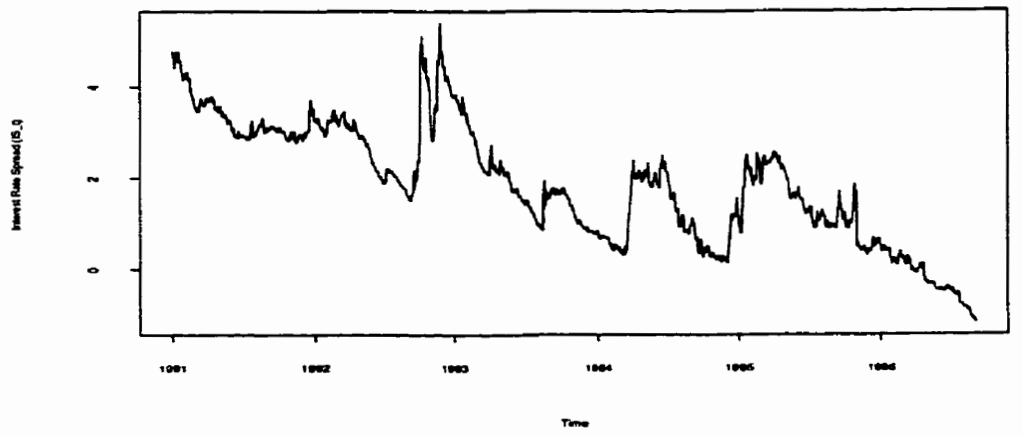
$$X_t = \begin{cases} +a & \text{if } X_t > +a \\ -a & \text{if } X_t < -a \end{cases}$$
$$Y_t = \begin{cases} +b & \text{if } Y_t > +b \\ -b & \text{if } Y_t < -b\,, \end{cases}$$

where $a = 4 \cdot \sigma_{X_t}$ and $b = 4 \cdot \sigma_{Y_t}$.

In particular, we will bound the movement in X_t between ± 38 $(= \pm 4 \cdot \hat{\sigma}_{X_t})$ basis

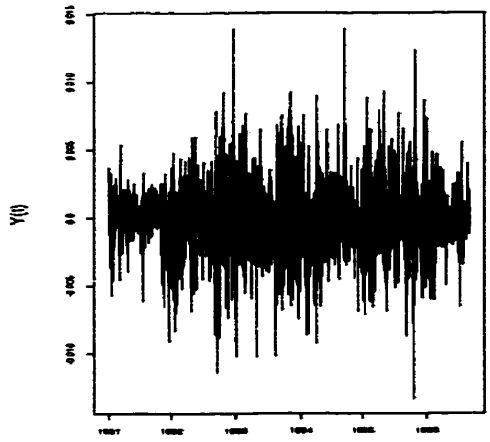
Figure 4.1: Time Series Plots for Spot Canadian Dollar and Interest Rate Spread.

A) Time Series Plot for Spot Canadian Dollar.

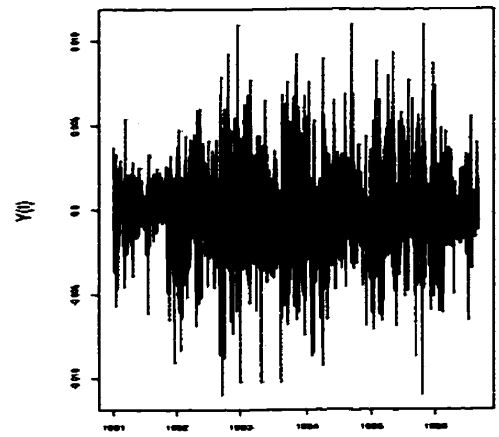


B) Time Series Plot for Interest Rate Spread.

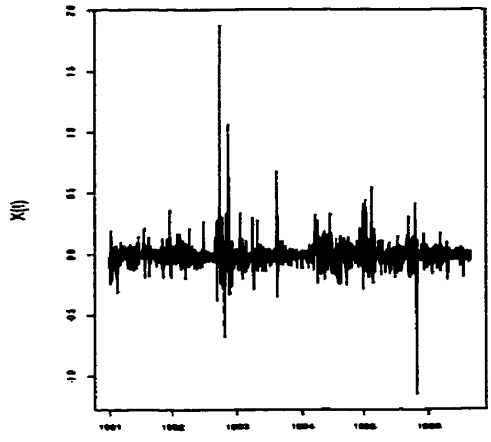
Figure 4.2: Changes in Spot Canadian Dollar (Y_t) and Interest Rate Spread (X_t).



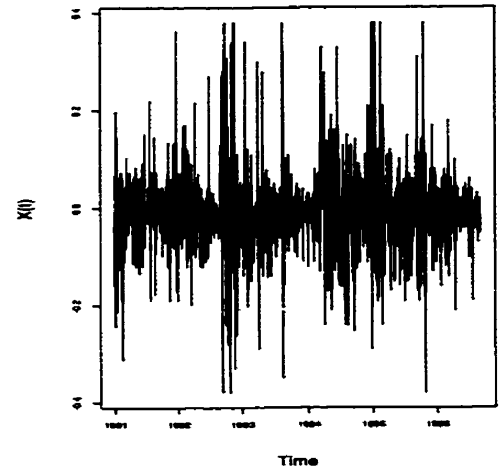
A) The Changes in Spot Canadian Dollar.



B) The Changes in Spot Canadian Dollar wit Adjustment.



C) The Changes in Interest Rate Spread.



D) The Changes in Interest Rate Spread witl Adjustment.

points, and bound the movement in Y_t between ± 1.1 $(= \pm 4 \cdot \hat{\sigma}_{Y_t})$ basis points. This means that we adjust the extreme points and replace them as follow:

$$X_t = \begin{cases} 0.38 & \text{if } X_t > 0.38 \\ -0.38 & \text{if } X_t < -0.38 \end{cases}$$

$$Y_t = \begin{cases} 0.011 & \text{if } Y_t > 0.011 \\ -0.011 & \text{if } Y_t < -0.011 \end{cases}.$$

In Figure 4.2B and 4.2D, we replot the time plots of Y_t and X_t with these changes. There are a total of 12 extreme points in X_t and a total of 5 extreme points in Y_t adjusted using the $4 \cdot \hat{\sigma}$ rule.

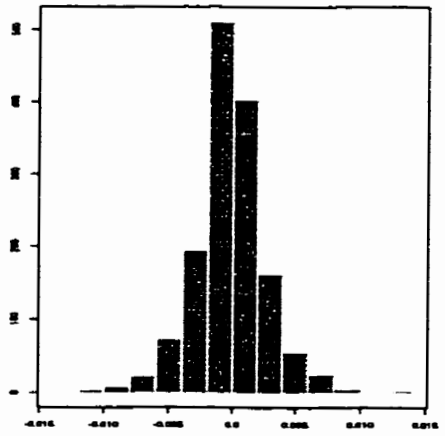
The histogram of Y_t and X_t series are presented in Figure 4.3. Notice that all the histograms are symmetrical at zero.

Now that we have the data in hand, an initial approach is to use spectral analysis to determine the existence of the relationship between the Y_t and X_t series. From Chapter 2, we have a consistent estimator of the spectral density function, which we apply to the Y_t and X_t series using the modified Daniell smoothing function in Splus with a span of 71 points ($m = 35$). In Figure 4.4, we plot the spectrums of the two series.

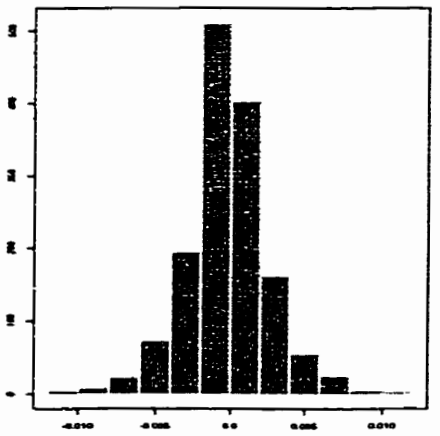
From the figures, we can see that there is some power in the high frequencies in Y_t series and low frequencies in X_t series. Although we do not know where the power is coming from, we can relate some of the power to the short noisy appearance in the time plots of Figure 4.2B and 4.2D.

Next, Figure 4.5 shows the cumulative periodogram and the 95% confidence limits

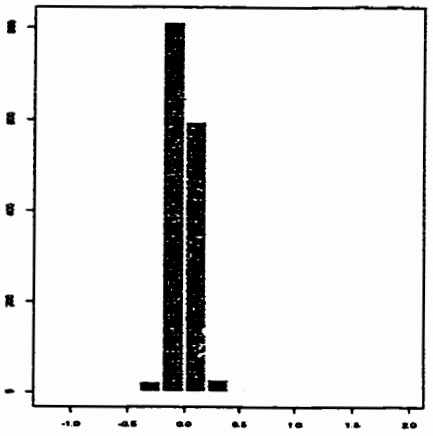
Figure 4.3: Histograms of the Changes in Spot Canadian Dollar and Interest Rate Spread Data.



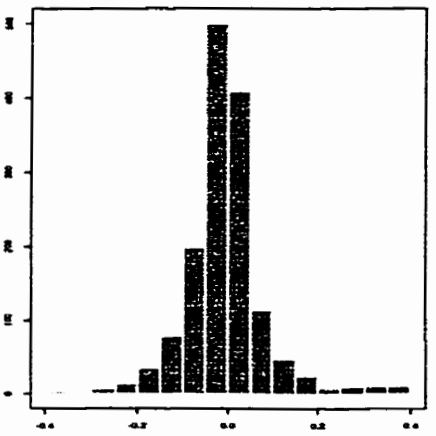
A) Histogram of the Changes in Spot Canadian Dollar.



B) Histogram of the Changes in Spot Canadian Dollar with Adjustment.

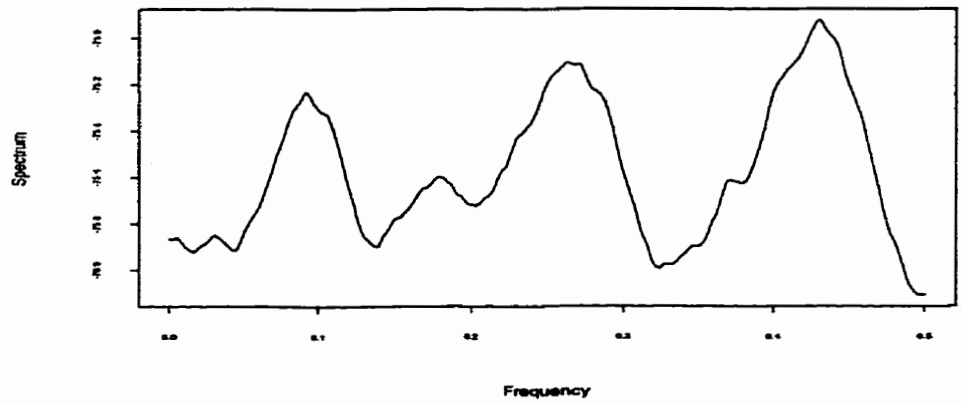


C) Histogram of the Changes in Interest Rate Spread.

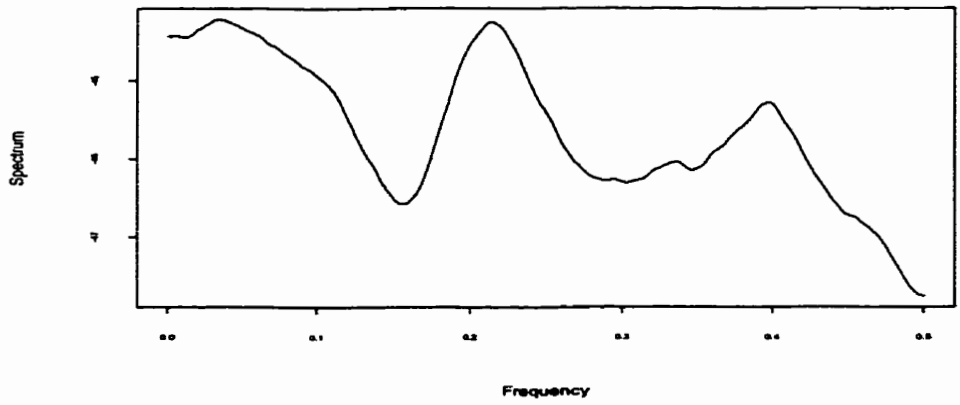


D) Histogram of the Changes Interest Rate Spread with Adjustment.

Figure 4.4: Spectral Density Plots of Y_t and X_t Series.



A) Spectral Density of Y_t Series.



B) Spectral Density of X_t Series.

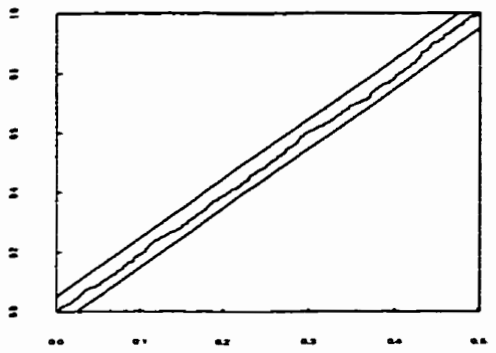
(Kolmogorov Smirnov boundaries) for both the X_t and Y_t series. It is apparent that the Y_t series falls within the confidence limits, whereas the X_t series fall outside of the confidence limits. The Kolmogorov Smirnov test (see Chapter 2) suggests that Y_t is white noise, although it is interesting to note that the spectral density of the Y_t series in Figure 4.4A is not "flat" enough; that is, this "visual" criterion disagrees with the Kolmogorov Smirnov test. Thus, we reject that the two time series are white noise, and we can carry on our analysis.

Since the bivariate time series is not white noise, we carry out the hypothesis testing defined in Chapter 2 in order to test whether these series are correlated at each frequency. To continue our example, using the modified Daniell smoothing function with $m = 35$, and $\alpha = 0.1$, we reject H_o if

$$\mathcal{S} = \frac{70\hat{K}^2(\nu)}{1 - \hat{K}^2(\nu)} > F_{0.90,\,2,\,140} = 2.3408\ . \tag{4.3}$$

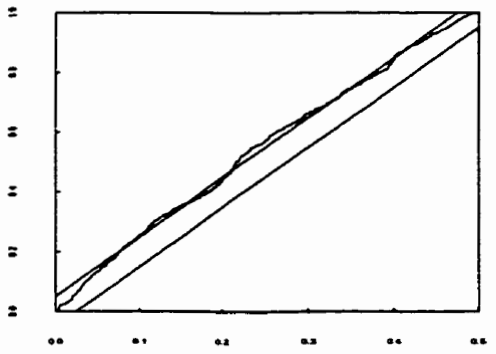
In Figure 4.6, a plot of the statistic $\mathcal{S}$ is presented with the accompanying F statistic ($F_{0.90,\,2,\,140}$). It is apparent that the hypothesis $K^2(\nu) = 0$ is rejected for all $\nu \in \left[0, \frac{1}{2}\right]$ and that the Y_t and the X_t series have correlation over the entire frequency range. Therefore, there is strong statistical evidence that the two series are correlated and we can proceed with our analysis.

Figure 4.5: Cumulative Spectrum Plots of Y_t and X_t Series.



Frequency

A) Cumulative Spectrum of Y_t Series.



Frequency

B) Cumulative Spectrum of X_t Series.

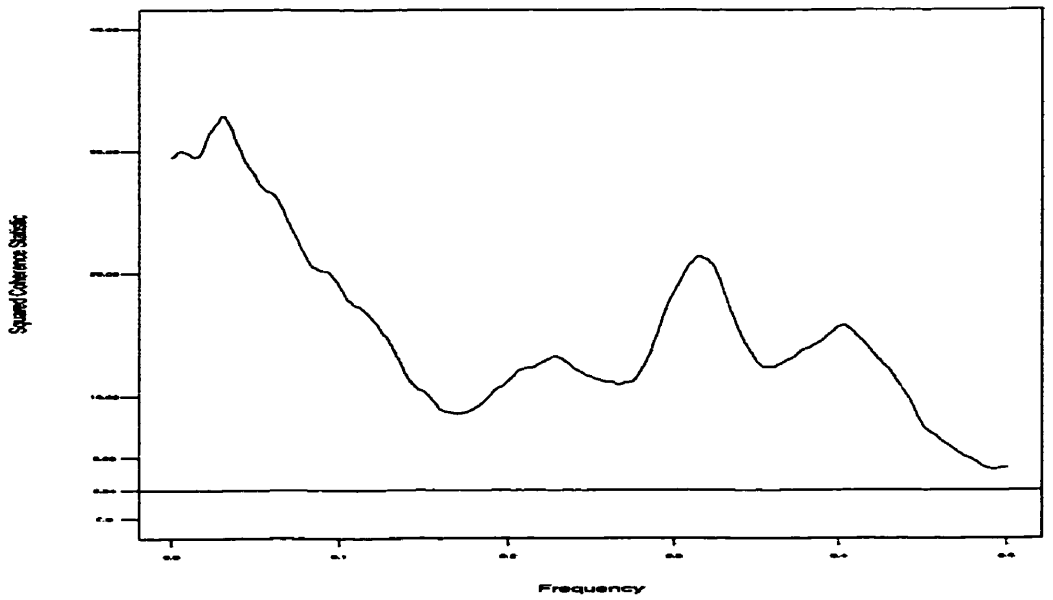
To proceed to our data analysis, that is, to generate bootstrap time series samples, we first apply the algorithm for choosing appropriate block size. Note that here we apply the algorithm to the Y_t series (spot Canadian dollar data). We plot the standard errors versus the block size in Figure 4.7. Note that the horizontal line is the true value of $\hat{\sigma}_\beta$ obtained from the Y_t series. It is evident that the plot suggests that $l = 60$ is preferred here. After fixing the block size $l = 60$, we follow the steps described in Chapter 2 in order to generate $B = 1500$ bootstrap time series samples. Next, we can use the integrated model to obtain a point estimate and/or inference for spot Canadian dollar series.

4.2 Empirical Results

In this section, the bootstrap forecast interval for the spot Canadian dollar is presented. The intervals are constructed with $B = 1500$, moving block size $l = 60$, and the multilayer perceptron network with 4 input layers, 3 hidden units for the hidden layer, and one output layer. Note that the three-layer multilayer perceptron network with the backpropagation algorithm is used to train on the input-output pair; that is, $((\tilde{X}_{t-4}, \tilde{X}_{t-3}, \tilde{X}_{t-2}, \tilde{X}_{t-1}),\ Y_t), t = 1, 2, \ldots, n$, where $\tilde{X}_{-3}, \tilde{X}_{-2}, \tilde{X}_{-1}$ and $\tilde{X}_0$ are set to zero.

Before we provide the point estimate and the forecast interval for spot Canadian dollar series, we need to carry out the following transformation:

Figure 4.6: Plot of Squared Coherence Statistic.



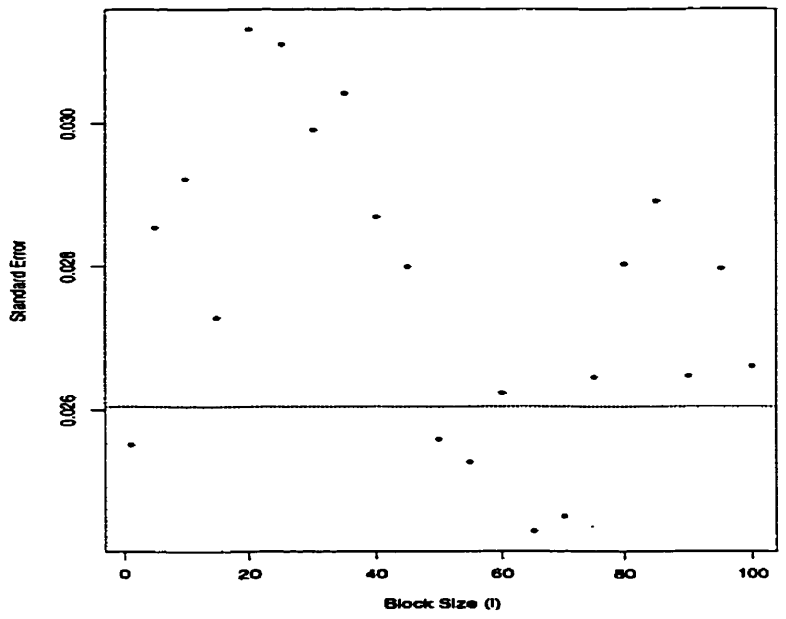


Figure 4.7: The plot of standard error estimates for all chosen block sizes. The true standard error is 0.02605

Recall that Y_t in (4.2) is defined as

$$Y_t = \log C_t - \log C_{t-1},$$

hence the forecast value for the spot Canadian dollar $\hat{C}_{n+1}$ is :

$$\begin{aligned} \hat{C}_{n+1} &= e^{\hat{Y}_{n+1} + \log(C_n)} \\ &= e^{\hat{Y}_{n+1}} \cdot C_n \, . \end{aligned} \tag{4.4}$$

As an example, the forecasted one step ahead future value for spot Canadian dollar, at $n = 1476$, is given by

$$\hat{C}_{n+1} = 73.0678 \, ,$$

and the 95% forecast interval is given by

$$(72.0251 < C_{n+1} < 74.0755) \, .$$

Note that, the actual value of C_{n+1} is 73.0033. These results justify the use of the integrated modelling technique to financial time series.

Chapter 5

Simulation

A simulation study is carried out to evaluate the performance of the model described in Chapter 3. Note that the integrated modelling technique utilizes both the classical techniques and nonparametric methods for this data analysis. For this study, we generate $n+1$ data with sample sizes $n = 500$, and $n = 1000$, for the X_t and Y_t series and simulate X_t as Interest rate spreads data and Y_t as spot Canadian dollars data. For each sample size, the simulated time series are generated with Gaussian and non-Gaussian noises. In addition, the X_t series are generated as follows: For simulation studies I - V, we have $MA = 0$ order and the orders for autoregressive (AR) are assigned to be 2, 3, 5,7, and 10; For simulation study VI, we have $AR = 2$ and $MA = 2$; and for simulation study VII, we have $AR = 5$ and $MA = 5$.

We will only use the first n samples of (X_t, Y_t) for model building. The following steps are applied to each simulated series: (1) The extreme points are adjusted according to the "adjustment rule" described in Chapter 4; (2) spectral analysis is

used to determine the existence of the relationship between the two series; (3) if the two series are not white noise and are correlated over the frequency range, we proceed to the next step, otherwise, we have to start on a new simulated series; (4) the algorithm for choosing the block size is applied; (5) the bootstrap time series samples are generated with $B = 1000$; (6) a three-layer multilayer perceptron network with the backpropagation algorithm is used to train on the input-output pair; (7) the future value for $\hat{Y}_{n+1}$, and the forecast interval for Y_{n+1} are forecasted; and (8) after obtaining the 95% forecast intervals for the Y_{n+1}, we then compare them to the actual Y_{n+1}. By repeating this procedure 200 times, with only the complete procedure counted, we can then calculate the percentage of the intervals that cover the actual or targeted Y_{n+1} for sample sizes $n = 500$ and $n = 1000$, respectively. Note that we use a bootstrap resample size of $B = 1000$ to construct the bootstrap forecast intervals. Next, we present the simulated time series and the simulation results.

5.1 Simulated Series I and Simulation Results

Simulated Series I(a): With Gaussian Noise

We simulate time series X_t as $ARIMA(2,1,0)$, as shown below:

$$(1 - \sum_{j=1}^{2} \phi_j B^j)(1 - B)X_t = U_t \,, \quad U_t \sim N(0,1) \tag{5.1}$$

where the coefficients $\phi_j,\ j = 1,2$ are randomly chosen to be 0.65 and -0.2. We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1-0.2^2}\cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t\,, \end{aligned} \tag{5.2}$$

where the function $min(\cdot)$ returns a number that is the minimum of the inputs, $trunc(\cdot)$ is the function that creates integers from floating point numbers by going to the next integer closer to zero, and H_t is defined by

$$(1-\phi B)(1-B)H_t = V_t\,, \quad V_t \sim N(0,1) \tag{5.3}$$

and a randomly chosen coefficient, $\phi = 0.32$.

Simulated Series I(b): With Non-Gaussian Noise

We simulate time series X_t as $ARIMA(2,1,0)$, as shown below:

$$\begin{aligned} &(1-\sum_{j=1}^{2}\phi_j B^j)(1-B)X_t = U_t\,, \\ &U_t \sim (80\%\text{ of } N(0,1) + 20\%\text{ of } Cauchy(0,1)) \end{aligned} \tag{5.4}$$

where the coefficients $\phi_j,\ j = 1,2$ are randomly chosen to be 0.58 and -0.12. We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1-0.2^2}\cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t\,, \end{aligned} \tag{5.5}$$

where H_t is defined by

$$(1-\phi B)(1-B)H_t = V_t\ , \quad V_t \sim N(0,1) \tag{5.6}$$

and a randomly chosen coefficient, $\phi = 0.45$.

Simulation Result I

The simulation results are summarised in Table 5.1. Note that the coverage probability for simulation results are lower than the nominal level of 95%. Moreover, the results show indifferently for both sample sizes as well as the noise distributions. However, for both samples sizes, the average error is higher in non-Gaussian noise series than the Gaussian noise series. The *average error*

Table 5.1: Results for Forecast Intervals on Simulated Series I

Noise	$N = 500$ $AR = 2$ and $MA = 0$			$N = 1000$ $AR = 2$ and $MA = 0$		
	Nominal Level	Coverage Proportion	Average Error	Nominal Level	Coverage Proportion	Average Error
Gaussian	95 %	92.0 %	0.801	95 %	92.5 %	0.763
Non-Gaussian	95 %	92.0 %	0.922	95 %	92.0 %	0.920

is defined by

$$\text{Average Error} = \frac{\sum_{t=1}^{N} |Y_t - \hat{Y}_t|}{N}\ . \tag{5.7}$$

5.2 Simulated Series II and Simulation Results

Simulated Series II(a): With Gaussian Noise

We simulate time series X_t as $ARIMA(3,1,0)$, as shown below:

$$(1-\sum_{j=1}^{3}\phi_j B^j)(1-B)X_t = U_t\,, \quad U_t \sim N(0,1)\,. \tag{5.8}$$

Note that the series is generated using the Splus function. We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1-0.2^2}\cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t\,, \end{aligned} \tag{5.9}$$

where H_t is defined by

$$(1-\phi B)(1-B)H_t = V_t\,, \quad V_t \sim N(0,1)\,. \tag{5.10}$$

Note that the series H_t is also generated using Splus function.

Simulated Series II(b): With Non-Gaussian Noise

We simulate time series X_t as $ARIMA(3,1,0)$, as shown below:

$$(1-\sum_{j=1}^{3}\phi_j B^j)(1-B)X_t = U_t\ ,$$
$$U_t \sim (80\% \text{ of } N(0,1) + 20\% \text{ of } Cauchy(0,1))\ . \qquad (5.11)$$

We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1-0.2^2}\cdot H_t \\ Y_t &= |trunc\ \{min(Z_t)\}| + 1 + Z_t\ , \end{aligned} \qquad (5.12)$$

where H_t is defined by

$$(1-\phi B)(1-B)H_t = V_t\ , \quad V_t \sim N(0,1)\ . \qquad (5.13)$$

Simulation Result II

The simulation results are summarised in Table 5.2. Note that the autoregressive order is 3 in this simulation study. Although the coverage probability for simulation result II show slight improvement in coverage probability as compared with the simulation result I, the coverage probability is still under the nominal level of 95%. The average error for non-Gaussian noise series remained high at about 1 basis point (which is about $4\cdot\sigma_{Y_t}$ in actual spot Canadian dollar data in Chapter 4).

Table 5.2: Results for Forecast Intervals on Simulated Series II

Noise	$N = 500$ $AR = 3$ and $MA = 0$			$N = 1000$ $AR = 3$ and $MA = 0$		
	Nominal Level	Coverage Proportion	Average Error	Nominal Level	Coverage Proportion	Average Error
Gaussian	95 %	93.0 %	0.873	95 %	93.0 %	0.861
Non-Gaussian	95 %	92.0 %	0.902	95 %	92.5 %	0.911

5.3 Simulated Series III and Simulation Results

Simulated Series III(a): With Gaussian Noise

We simulate time series X_t as $ARIMA(5,1,0)$, as shown below:

$$(1 - \sum_{j=1}^{5} \phi_j B^j)(1 - B)X_t = U_t \,, \quad U_t \sim N(0,1) \,. \tag{5.14}$$

We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1 - 0.2^2} \cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t \,, \end{aligned} \tag{5.15}$$

where H_t is defined by

$$(1 - \phi B)(1 - B)H_t = V_t \,, \quad V_t \sim N(0,1) \,. \tag{5.16}$$

Simulated Series III(b): With Non-Gaussian Noise

We simulate time series X_t as $ARIMA(5,1,0)$, as shown below:

$$(1-\sum_{j=1}^{5}\phi_j B^j)(1-B)X_t = U_t \ ,$$
$$U_t \sim (80\% \text{ of } N(0,1) + 20\% \text{ of } Cauchy(0,1)) \ . \qquad (5.17)$$

We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1-0.2^2}\cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t \ , \end{aligned} \qquad (5.18)$$

where H_t is defined by

$$(1-\phi B)(1-B)H_t = V_t \ , \quad V_t \sim N(0,1) \ . \qquad (5.19)$$

Simulation Result III

The simulation results are summarised in Table 5.3. In this study, the autoregressive order is increase from 3 to 5. The coverage probability for simulation results show decreases as compared with simulation results I and II. However, the average error for non-Gaussian noises are appears closer to Gaussian noise series for both sample sizes. The coverage probability for these simulation results are still lower than the nominal level of 95%.

Table 5.3: Results for Forecast Intervals on Simulated Series III

Noise	$N = 500$ $AR = 5$ and $MA = 0$			$N = 1000$ $AR = 5$ and $MA = 0$		
	Nominal Level	Coverage Proportion	Average Error	Nominal Level	Coverage Proportion	Average Error
Gaussian	95 %	92.0 %	0.882	95 %	92.0 %	0.863
Non-Gaussian	95 %	91.5 %	0.900	95 %	92.0 %	0.874

5.4 Simulated Series IV and Simulation Results

Simulated Series IV(a): With Gaussian Noise

We simulate time series X_t as $ARIMA(7,1,0)$, as shown below:

$$(1 - \sum_{j=1}^{7} \phi_j B^j)(1 - B)X_t = U_t \ , \quad U_t \sim N(0,1) \ . \tag{5.20}$$

We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1 - 0.2^2} \cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t \ , \end{aligned} \tag{5.21}$$

where H_t is defined by

$$(1 - \phi B)(1 - B)H_t = V_t \ , \quad V_t \sim N(0,1) \ . \tag{5.22}$$

Simulated Series IV(b): With Non-Gaussian Noise

We simulate time series X_t as $ARIMA(7,1,0)$, as shown below:

$$(1-\sum_{j=1}^{7}\phi_j B^j)(1-B)X_t = U_t\ ,$$
$$U_t \sim (80\% \text{ of } N(0,1) + 20\% \text{ of } Cauchy(0,1))\ . \qquad (5.23)$$

We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1-0.2^2}\cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t\ , \end{aligned} \qquad (5.24)$$

where H_t is defined by

$$(1-\phi B)(1-B)H_t = V_t\ , \quad V_t \sim N(0,1)\ . \qquad (5.25)$$

Simulation Result IV

The simulation results are summarised in Table 5.4. Note that the autoregressive order is now 7. The coverage probability for the simulation results show no improvement; they also show decreases as compared with the previous simulation results. As usual, the coverage probability for the simulation results are lower than the nominal level. But the average errors are almost the same for both noise distributions within the same sample size.

Table 5.4: Results for Forecast Intervals on Simulated Series IV

Noise	$N = 500$ $AR = 7$ and $MA = 0$			$N = 1000$ $AR = 7$ and $MA = 0$		
	Nominal Level	Coverage Proportion	Average Error	Nominal Level	Coverage Proportion	Average Error
Gaussian	95 %	91.5 %	0.781	95 %	92.0 %	0.832
Non-Gaussian	95 %	90.5 %	0.779	95 %	91.0 %	0.869

5.5 Simulated Series V and Simulation Results

Simulated Series V(a): With Gaussian Noise

We simulate time series X_t as $ARIMA(10, 1, 0)$, as shown below:

$$(1 - \sum_{j=1}^{10} \phi_j B^j)(1 - B)X_t = U_t \ , \quad U_t \sim N(0, 1) \ . \tag{5.26}$$

We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1 - 0.2^2} \cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t \ , \end{aligned} \tag{5.27}$$

where H_t is defined by

$$(1 - \phi B)(1 - B)H_t = V_t \ , \quad V_t \sim N(0, 1) \ . \tag{5.28}$$

Simulated Series V(b): With Non-Gaussian Noise

We simulate time series X_t as $ARIMA(10,1,0)$, as shown below:

$$(1-\sum_{j=1}^{10}\phi_j B^j)(1-B)X_t = U_t\ ,$$
$$U_t \sim (80\% \text{ of } N(0,1) + 20\% \text{ of } Cauchy(0,1))\ . \qquad (5.29)$$

We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1-0.2^2}\cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t\ , \end{aligned} \qquad (5.30)$$

where H_t is defined by

$$(1-\phi B)(1-B)H_t = V_t\ , \quad V_t \sim N(0,1)\ . \qquad (5.31)$$

Simulation Result V

The simulation results are summarised in Table 5.5. Notice that the coverage probability for simulation results are decreasing as the autoregressive order increases to 10. In fact, the average errors are higher than usual. In this simulation study, we still have the coverage probability lower than nominal level.

Table 5.5: Results for Forecast Intervals on Simulated Series V

Noise	$N = 500$ $AR = 10$ and $MA = 0$			$N = 1000$ $AR = 10$ and $MA = 0$		
	Nominal Level	Coverage Proportion	Average Error	Nominal Level	Coverage Proportion	Average Error
Gaussian	95 %	90.0 %	1.080	95 %	90.5 %	0.897
Non-Gaussian	95 %	89.0 %	1.110	95 %	90.0 %	0.903

5.6 Simulated Series VI and Simulation Results

Simulated Series VI(a): With Gaussian Noise

We simulate time series X_t as $ARIMA(2,1,2)$, as shown below:

$$(1 - \sum_{j=1}^{2} \phi_j B^j)(1 - B)X_t = (1 + \sum_{j=1}^{2} \theta_j B^j)U_t \,, \quad U_t \sim N(0,1) \,. \tag{5.32}$$

We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1 - 0.2^2} \cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t \,, \end{aligned} \tag{5.33}$$

where H_t is defined by

$$(1 - \phi B)(1 - B)H_t = V_t \,, \quad V_t \sim N(0,1) \,. \tag{5.34}$$

Simulated Series VI(b): With Non-Gaussian Noise

We simulate time series X_t as $ARIMA(2,1,2)$, as shown below:

$$(1-\sum_{j=1}^{2}\phi_j B^j)(1-B)X_t = (1+\sum_{j=1}^{2}\theta_j B^j)U_t \ ,$$
$$U_t \sim (80\% \text{ of } N(0,1) + 20\% \text{ of } Cauchy(0,1)) \ . \qquad (5.35)$$

We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1-0.2^2}\cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t \ , \end{aligned} \qquad (5.36)$$

where H_t is defined by

$$(1-\phi B)(1-B)H_t = V_t \ , \quad V_t \sim N(0,1) \ . \qquad (5.37)$$

Simulation Result VI

The simulation results are summarised in Table 5.6. Note that we have the $MA = 2$ and $AR = 2$ in this simulation study. The coverage probability for simulation results are very similar to simulation results I and II. However, for respective sample size, the average errors are almost indifferent for both Gaussian and non-Gaussian noises. Again, the coverage probability results are still lower than the nominal level.

Table 5.6: Results for Forecast Intervals on Simulated Series VI

Noise	$N = 500$ $AR = 2$ and $MA = 2$			$N = 1000$ $AR = 2$ and $MA = 2$		
	Nominal Level	Coverage Proportion	Average Error	Nominal Level	Coverage Proportion	Average Error
Gaussian	95 %	92.0 %	0.887	95 %	92.0 %	0.916
Non-Gaussian	95 %	92.0 %	0.879	95 %	92.0 %	0.920

5.7 Simulated Series VII and Simulation Results

Simulated Series VII(a): With Gaussian Noise

We simulate time series X_t as $ARIMA(5,1,5)$, as shown below:

$$(1-\sum_{j=1}^{5}\phi_j B^j)(1-B)X_t = (1+\sum_{j=1}^{5}\theta_j B^j)U_t\ , \quad U_t \sim N(0,1)\ . \tag{5.38}$$

We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1-0.2^2}\cdot H_t \\ Y_t &= |trunc\,\{min(Z_t)\}| + 1 + Z_t\ , \end{aligned} \tag{5.39}$$

where H_t is defined by

$$(1-\phi B)(1-B)H_t = V_t\ , \quad V_t \sim N(0,1)\ . \tag{5.40}$$

Simulated Series VII(b): With Non-Gaussian Noise

We simulate time series X_t as $ARIMA(5,1,5)$, as shown below:

$$(1-\sum_{j=1}^{5}\phi_j B^j)(1-B)X_t = (1+\sum_{j=1}^{5}\theta_j B^j)U_t \ ,$$
$$U_t \sim (80\% \text{ of } N(0,1) + 20\% \text{ of } Cauchy(0,1)) \ . \qquad (5.41)$$

We simulate time series Y_t as follows:

$$\begin{aligned} Z_t &= -0.2X_t + \sqrt{1-0.2^2}\cdot H_t \\ Y_t &= |trunc\{min(Z_t)\}| + 1 + Z_t \ , \end{aligned} \qquad (5.42)$$

where H_t is defined by

$$(1-\phi B)(1-B)H_t = V_t \ , \quad V_t \sim N(0,1) \ . \qquad (5.43)$$

Simulation Result VII

The simulation results are summarised in Table 5.7. As we increase both the order for MA and AR to 5, the coverage probability for simulation results are considered the lower as compared to previous simulation results. Also, the average errors show no improvement.

Table 5.7: Results for Forecast Intervals on Simulated Series VII

Noise	$N = 500$ $AR = 5$ and $MA = 5$			$N = 1000$ $AR = 5$ and $MA = 5$		
	Nominal Level	Coverage Proportion	Average Error	Nominal Level	Coverage Proportion	Average Error
Gaussian	95 %	90.0 %	1.053	95 %	89.5 %	0.976
Non-Gaussian	95 %	89.5 %	1.092	95 %	89.5 %	1.022

5.8 Discussion

In Table 5.8, it is apparent that the coverage probability for all simulation results are lower than the nominal level of 95%. There are five possible reasons for this low coverage probability: (1) The estimates taken to be the true values used for constructing the bootstrap forecast intervals are inaccurate. (2) The confidence intervals based on bootstrap percentiles method may not have performed well due to the consequence of nonparametric inferences. It may have underestimated the tails of the distribution of prediction. (3) The number of repetitions for the simulation procedure in this study was too low (in this study the simulation procedure was only repeated 200 times). (4) The adjustment rule for extreme points might affect the results for coverage probability. However, because we are trying to model the long term relationship for bivariate financial series, it seems reasonable to apply the adjustment rule to those extreme points for better model analysis. (5) As the simulation results show, the coverage probability decreases as we increase the order

for MA and/or AR. This may be due to computation errors occurring as the model becomes more complicated.

Table 5.8: Results for 95% Forecast Intervals on Simulated Series

Simulated Series	$N = 500$		$N = 1000$	
	Gaussian	Non-Gaussian	Gaussian	Non-Gaussian
I ($AR = 2$, $MA = 0$)	92.0 %	92.0 %	92.5 %	92.0 %
II ($AR = 3$, $MA = 0$)	93.0 %	92.0 %	93.0 %	92.5 %
III ($AR = 5$, $MA = 0$)	92.0 %	91.5 %	92.0 %	92.0 %
IV ($AR = 7$, $MA = 0$)	91.5 %	90.5 %	92.0 %	91.0 %
V ($AR = 10$, $MA = 0$)	90.0 %	90.0 %	90.5 %	90.0 %
VI ($AR = 2$, $MA = 2$)	92.0 %	92.0 %	92.0 %	92.0 %
VII ($AR = 5$, $MA = 5$)	90.0 %	89.5 %	89.5 %	89.5 %

The results obtained when $n = 500$ do not differ greatly as compared with those obtained when $n = 1000$. In addition, the simulation results show that the integrated modelling technique is insensitive to the choice of distribution for the noise. Since we must clean up the extreme points for each of the simulated series, this may explain why this technique is insensitive to the Gaussian or non-Gaussian noise. However, there is a slight decrease in coverage probability when we increase the order for AR

and MA. This also suggests that we may want to keep the model simple in order to save computation time and obtain better results. Again, the integrated modelling technique is not sensitive to the complicated series.

Although the results seem reasonable when using the integrated modelling technique, the operation costs are high in terms of CPU.

To conclude, the integrated modelling technique is worthwhile when the complexity of the data requires a more versatile approach than allowed by classical methods. As to the reason for under coverage in probability, this remains an issue to be resolved in future studies.

Chapter 6

Conclusions

This thesis has demonstrated the usefulness of both classical methods and non-parametric methods in analysing financial time series. The approach developed in Chapter 2 give the basic ideas for implementing techniques in modelling financial time series.

As discussed in Chapter 2, commonly used methodologies (such as ARIMA models) provide us with estimation and forecast interval. However, in financial markets, time series are statistically more volatile as compared to other applications. Thus, simple models alone are unable to solve difficult problems such as exchange rate prediction. The bivariate transfer function model is introduced in order to improve the forecast estimation as compared with the univariate case. However, the transfer function model requires the value of X_{n+1}, which is not available and must itself be forecast in order to forecast Y_{n+1}. The accuracy of the forecasting interval for Y_{n+1} can be improved by not requiring the estimation of X_{n+1}, thus reducing the probability of error in estimation of Y_{n+1}, as shown by the integrated model.

The frequency domain methods provide us (with the aid of high speed computer hardware), a more efficient way to uncover certain statistical details. Particularly for the bivariate time series, we need to determine the existence of a relationship between two time series in order to further the data analysis. Together with the classical techniques, modern nonparametric methods such as the moving blocks bootstrap and neural network provide us with a theoretical basis for the derivation of an integrated modelling technique.

The integrated modelling technique described in Chapter 3 provides us with the tools to further analyse the bivariate time series. The technique utilizes both classical techniques and nonparametric methods for better estimation and inference. In Chapter 4, an illustration of integrated modelling techniques is applied to the financial time series. The empirical result support the use of the proposed modelling technique.

In Chapter 5, although the simulation study was only carried out for n=500 and n=1000, the simulated time series were generated with Gaussian and non-Gaussian noises. The simulation results showed undercoverage overall, but the integrated modelling technique appears robust to the choice of noise distribution as well as the sample size.

6.1 Future Research

This research could be extended by adding more variables to the model given in chapter 3. For the spot Canadian dollar example, we can add other variables such as the demand and/or supply of Canadian dollars, demand and/or supply of U.S. dollars, and the stock indexes to the model. The demand and supply of currency will help us predict whether or not future values will soon rise or fall. As for the stock indices, increase or decrease, this would help us to determine whether the market is performing well. Other possible variables to add are those pertaining to available information from the derivative markets.

By adding more variables to the model, two approaches to analysing the time series are possible. The first approach would be to utilize multiple linear regression techniques. Note that economists describe a regression model as an econometric model. Although multiple linear regression models are widely used due to their computational simplicity, there are problems inherent to this approach which need to be pointed out. One problem is that explanatory variables can be highly correlated, leading to singularity problems. Hence, it is advisable to look at the correlation matrix of the explanatory variables before carrying out a multiple linear regression analysis. Another problem involves the structure of the error terms. It is always assumed that the errors are $i.i.d.N(0, \sigma^2)$, but this assumption may not be correct. If the residuals are correlated, one can try fitting a multiple linear regression model

with autocorrelated errors (See Kendall *et al.* [37]).

The second approach is to follow the approach developed in Chapter 3, but to pair up the spot Canadian dollar with other variables. However, how do we combine the information from different pairings of Y_t (spot Canadian dollar series) and other variables, without producing too many forecast intervals?

Since the confidence intervals based on the bootstrap percentiles method have less satisfactory probability coverage, this research could be improved by replacing the confidence intervals with more advanced bootstrap intervals methods such as *bias-corrected and accelerated* (**BCa**) method, and the *approximate bootstrap confidence intervals* (**ABC**) method. These advance bootstrap intervals methods could partially correct the undercoverage problems in our simulation studies.

In dynamic representation, the recurrent networks may be more in line with nonlinear time series. Hence, by replacing the feed-forward network with recurrent network, we may achieve a better point estimation and satisfactory probability coverage.

Bibliography

[1] Abraham, B. and Ledolter, J. [1983]. *Statistical methods for forecasting.* John Wiley & Sons, New York.

[2] Anderson, T.W. [1971]. *The statistical analysis of time series.* John Wiley & Sons, New York.

[3] Beran, R. [1982]. Estimating sampling distributions : The bootstrap and competitors. *The Annals of Statistics* **10**, 212-225.

[4] Beran, R. [1988a]. Balanced simultaneous confidence sets. *Journal of the American Statistical Association* **83**, 679-686.

[5] Beran, R. and Ducharme, G.R. [1991]. *Asymptotic theory for bootstrap methods in statistics.* Centre de Recherches Mathématiques : Université de Montréal.

[6] Bickel, P.J. and Freedman, D.A. [1981]. Some asymptotic theory for the bootstrap. *The Annals of Statistics* **9**, 1196-1217.

[7] Bloomfield, P. [1976]. *Fourier analysis of time series : An introduction.* John Wiley & Sons, New York.

[8] Box, G.E.P. and Jenkins, G.M. [1976]. *Time series analysis : Forecasting and control.* Holden-Day, San Francisco.

[9] Brockwell, P.J. and Davis R.A. [1987]. *Time series : Theory and methods.* Springer-Verlag, New York.

[10] Chatfield, C. [1996]. *The Analysis of Time Series : An introduction.* 5th Ed., Chapman and Hall, London.

[11] Cheng, B. and Titterington, D.M. [1994]. Neural networks : A review from a statistical perspective. *Statistical Science* **9**, 2-54.

[12] Christensen, R. [1991]. *Linear models for multivariate, time series, and spatial data.* Springer-Verlag, New York.

[13] Cybenko, G. [1989]. Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals and Systems* **2**, 303-314.

[14] Diaconis, P. and Efron, B. [1983]. Computer intensive methods in statistics. *Scientific American* **248(5)**, 116-130.

[15] Efron, B. [1979a]. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics* **7**, 1-26.

[16] Efron, B. [1979b]. Computers and the theory of statistics : Thinking the unthinkable. *SIAM Review* **21**, 460-480.

[17] Efron, B. [1981]. Nonparametric estimates of standard error : The jackknife, the bootstrap and the other methods. *Biometrics* **68**, 589-599.

[18] Efron, B. [1982]. *The jackknife, the bootstrap and other resampling plans.* Society for Industrial and Applied Mathematics. Philadelphia, Pennsylvania.

[19] Efron, B. and Tibshirani, R.J. [1986]. Bootstrap methods for standard errors, confidence intervals and other measures of statistical accuracy. *Statistical Science* **1**, 54-77.

[20] Efron, B. and Tibshirani R.J. [1993]. *An introduction to the bootstrap.* Chapman and Hall, New York.

[21] Franke, J. and Härdle, W. [1992]. On bootstrapping kernal spectral estimation. *The Annals of Statistics* **20**, 121-145.

[22] Freedman, D.A. [1981]. Bootstrapping regression models. *The Annals of Statistics* **9**, 1218-1228.

[23] Freedman, D.A. [1984]. On bootstrapping two-stage least-squares estimates in stationary linear models. *The Annals of Statistics* **12**, 827-842.

[24] Funanhashi, K.I. [1989]. On the approximate realization of continuous mappings by neural networks. *Neural Networks* **2**, 183-192.

[25] Gallant, A.R. and White, H. [1988]. There exists a neural network that does not make avoidable mistakes. *Proceedings of the second annual IEEE Conference on Neural Networks*, San Diego, CA, Vol 1 (IEEE Press, New York), 657-664.

[26] Gallant, A.R. and White, H. [1992]. On learning the derivatives of an unknown mapping with multilayer feedforward networks. *Neural Networks* **5**, 129-138.

[27] Granger, C.W.J. [1992]. Forecasting stock market prices : Lessons for forecasters. *International Journal of Forecasting* **8**, 3-13.

[28] Hall, P. [1986]. On the number of bootstrap simulations required to construct a confidence interval. *The Annals of Statistics* **14**, 1463-1562.

[29] Hall, P. [1988]. Theoretical comparison of bootstrap confidence intervals (with discussion). *The Annals of Statistics* **16**, 927-985.

[30] Haykin, S. [1994]. *Neural networks : A comprehensive foundation.* Macmillan, New York.

[31] Hecht-Nielsen, R. [1989]. Theory of the backpropagation neural networks. *Proceedings of the International Joint Conference on Neural Networks*, Washington D. C., Vol 1 (IEEE Press, New York), 593-605.

[32] Hornik, K., Stinchcombe, M. and White, H. [1989]. Multilayer feedforward networks are universal approximators. *Neural Networks* **2**, 359-366.

[33] Hornik, K., Stinchcombe, M. and White, H. [1990]. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks* **3**, 551-560.

[34] Hornik, K., Stinchcombe, M., White, H. and Auer, P. [1994]. *Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives.* UCSD discussion paper.

[35] Hwang, J.T. and Ding, A.A. [1997]. Prediction intervals for artificial neural networks. *Journal of the American Statistical Association* **92**, 748-757.

[36] Jenkins, G.M. and Watts, D.G. [1969]. *Spectral analysis and its applications.* Holden-Day, California.

[37] Kendall, M.G., Stuart, A. and Ord, J.K. [1983]. *The advanced theory of statistics.* vol. 3 (4th edn.), London, Griffin.

[38] Kim, P.T., Martin, P., and Kelly, D. [1992]. Unanticipated changes in interest rate spreads and their effect on the Canadian spot exchange rate : An update. *Fixed Income Research* : Wood Gundy Inc.

[39] Kim P.T., Martin P. and Staley M. [1996]. A neural network approach to forecasting the Canadian spot exchange rate. *Fixed Income Research* : Wood Gundy Inc.

[40] Kim P.T. and Martin P. [1996]. On the relationship between the interest rate spread and the spot Canadian dollar. University of Guelph Lecture Notes.

[41] Kuan, C.M. and White, H. [1994]. Artificial neural networks: An econometric perspective. *Econometric Reviews* **13**, 1-91.

[42] Kuan, C.M. and Liu, T. [1995]. Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of Applied Econometrics* **10**, 347-364.

[43] Künsch, H.R. [1989]. The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics* **17**, 1217-1241.

[44] Léger, C., Politis, D.N. and Romano, J.P. [1992]. Bootstrap technology and applications. *Technometrics* **34**, 378-398.

[45] Lippmann, R.P. [1987]. An introduction to computing with neural nets. *IEEE ASSP Magazine*, April Issue, 4-23.

[46] Liu, R.Y. and Singh, K. [1992]. *Moving blocks jackknife and bootstrap capture weak dependence*, In Exploring the limits of bootstrap, eds. LePage, R. and Billard, L., John Wiley, New York.

[47] Ljung, G.M. and Box, G.E.P. [1978]. On a measure of lack of fit in time series models. *Biometrika* **65**, 297-303.

[48] Priestly, M.B. [1981]: *Spectral analysis and time Series.* Academic Press, New York.

[49] Quenouille, M.H. [1956]. Notes on bias in estimation. *Biometrika* **43**, 353-360.

[50] Rao, V.B. and Rao, H.V. [1993]. *C++ neural networks and fuzzy logic.* MIS:Press, New York.

[51] Ripley, B.D. [1993]. Statistical aspects of neural networks. In *Networks and Chaos - Statistical and Probabilistic Aspects* (eds O.E. Barndorff-Nielsen *et al.*). Chapman & Hall, London, 40-123.

[52] Stacey, D. [1995] Lecture notes : COMP 27-6420 (Spring Semester). Department of Computing and Information Science, U. of Guelph.

[53] Tibshirani, R.J. [1985]. *How many bootstraps ?* (Tech. rep. no. 362). Stanford University, Dept. of Statistics. Stanford, CA.

[54] Weigend, A.S. and Gershenfeld, N.A. (eds) [1994]. *Time series prediction*, Reading, MA : Addison-Wesley.

[55] White, H. [1989]. Learning in artificial neural networks : A statistical perspective. *Neural Computation* **1**, 425-464.

[56] White, H. [1992]. *Artificial neural networks: approximation and learning theory.* Cambridge, MA: Blackwell.

[57] Wu, C.F.J. [1986]. Jackknife, bootstrap and other resampling methods in regression analysis. *The Annals of Statistics* **14**, 1261-1295.

Appendix A : The Back-Propagation Algorithm

Before we provide the back-propagation algorithm, we present the multilayer perceptron network model used for financial data analysis. For the given input $\boldsymbol{X}(s) = (X_1(s), X_2(s), \ldots, X_k(s))^t$, $s = 1, 2, \ldots, n$, the output of the network is defined as

$$G_\Theta(\boldsymbol{X}) = \Gamma_o + \sum_{s=1}^{p} w_s \, f(\boldsymbol{v}_s^t \boldsymbol{X} + \Phi_s)$$

where the activation function $f(\cdot)$ is the logistic function defined in (2.28), and $\Theta = (w_1, w_2, \ldots, w_p, \boldsymbol{v}_1^t, \boldsymbol{v}_2^t, \ldots, \boldsymbol{v}_p^t, \Gamma_o, \Phi_1, \Phi_2, \ldots, \Phi_p)$ where $\boldsymbol{v}_s^t = (v_{s1}, v_{s2}, \ldots, v_{sk})$, are the weights and threshold values, and the superscript t is denoted as transpose.

The following backpropagation algorithm is modified from the lecture notes by Stacey [52].

1. Notations :

L_I	:	the input layer
L_H	:	the hidden layer
L_O	:	the output layer
v_{ih}	:	the synaptic weight between the L_I and L_H
w_{ho}	:	the synaptic weight between the L_H and L_O
Φ_h	:	L_HProcessing Elements (PE) threshold values
Γ_o	:	L_OPE threshold values

Note that we assign random values in the range $[-1, +1]$ to v_{ih}, w_{ho}, Φ_h, and Γ_o.

2. **Forward Pass Computation :** For each input-output pair ($\boldsymbol{X}_s$, O_s):

   a. Process $\boldsymbol{X}_s$'s values to calculate new L_H PE activations using

$$b_i = f(\sum_{h=1}^{k} x_h v_{hi} + \Phi_i)$$

   where $f(\cdot)$ is the sigmoid function defined in equation 2.28.

   b. Filter the L_H activations through the weights; W, to L_O using

$$O_j = \sum_{i=1}^{p} b_i w_{ij} + \Gamma_j$$

   c. Compute the output differences by $d_j = (O_j^s - O_j)$

3. **Backward Pass Computation :**

   I. Compute the output error at L_O PE values using

$$e_j = O_j(1 - O_j)d_j$$

   II. Calculate the error of each L_H PE relative to each d_j with

$$t_i = b_i(1 - b_i)\sum_{i=1}^{p} w_{ij}e_j$$

   III. Adjust the L_H to L_O connections; that is, the amount of change made to the connection from the i^{th} L_H to the j^{th} L_O.

$$\Delta w_{ij} = \alpha b_i e_j$$

   where α is a positive constant controlling the learning rate.

   IV. Adjust the L_O threshold value $\Delta\Gamma_j = \alpha e_j$

   V. Adjust the L_I to L_H connections $\Delta v_{hi} = \beta x_h t_i$ where β is a positive constant controlling the learning rate.

   VI. Adjust the L_H threshold value $\Delta\Phi_i = \beta t_i$

4. **Iteration :** Repeat Step 2 and Step 3 until e_j's are all either zero or at the minimum value.

5. **Recall :** The recall mechanism consists of the two feed-forward operations.

   1. Create L_h PE values

$$b_i = f(\sum_{h=1}^{k} x_h v_{hi} + \Phi_i)$$

   2. After all L_H PE activations have been calculated, they are used to compute the output.

$$O_j = \sum_{i=1}^{p} b_i w_{ij} + \Gamma_j$$