DECISION TECHNOLOGIES FOR TRADING PREDICTABILITY IN FINANCIAL MARKETS

by

Neville Towers

A dissertation submitted to the University of London

for the degree of Doctor of Philosophy

Decision Technology Centre

LONDON BUSINESS SCHOOL

UNIVERSITY OF LONDON

Acknowledgements

First and foremost I like to thank my supervisor, Paul Refenes, for his constructive criticism and guidance whilst allowing me the space to explore my own research interests.

I am particularly grateful to Neil Burgess for his creative ideas and suggestions on many aspects of this work and for generously giving up his "spare" time to proof read my thesis. Special thanks also go to Peter Bolland, Yves Bentz, Jerry Connor and the other members of the Computational Finance Group for their helpful discussions and for generally making my time at London Business School enjoyable.

I would also like to thank Jimmy Shadbolt and Drago Indjic for encouraging me to pursue a Ph.D. in the first place and for stimulating some of my initial research ideas.

Finally, I am immensely grateful for the love and support of my family and friends, especially my wife Sarah, who has been a constant source of strength over the last four years.

Financially, this work was supported by the ESRC, Nestlé UK and a Ph.D. programme scholarship from London Business School.

Abstract

The traditional view amongst financial economists is that well-functioning financial markets are unpredictable and hence provide no means for systematic excess profit. In the last decade, however, this view has been consistently challenged by empirical studies showing low but significant levels of predictable behaviour in asset returns, and explicit evidence that statistical forecasting models can have a measure of significant predictive power. If these apparent "systematic regularities" exist then it invites the development of methodologies to exploit predictability by generating economically significant trading strategies.

This thesis develops a methodology to optimise trading strategies for arbitrary forecasting models which possess some degree of predictive ability. It exploits recent advances in decision theory to account for both expected returns and trading costs within the decision making process and thus reflect the reality of typical trading environments. In this context, we develop a framework to jointly optimise the trading performance for a pair of decision and forecasting models.

The methodology consists of two stages. First, given an arbitrary forecasting model, we develop methods to approximate the optimal trading strategy dependent on the trading conditions. In particular, we develop trading strategies using parameterised decision rules and an enhanced reinforcement learning algorithm. Secondly, given a trading policy, we examine the multi-objective optimisation of the forecasting model. We describe a *meta*-parameter approach in which the forecasting model is optimised with respect to a number of different statistical characteristics which affect trading performance. We investigate three specific characteristics, namely forecast horizon, prediction smoothness and predictive correlation. The two stages are then combined to perform a joint optimisation over both forecasting and decision models. We empirically evaluate optimisation procedures using controlled simulations and in the application of statistical arbitrage trading. Our results demonstrate that joint optimisation can significantly improve performance in the presence of trading costs.

Table of Contents

1 Introduction

1.1 Scope

The dominant view amongst financial economists is that well–functioning financial markets are unpredictable and "efficient" implying that changes in asset prices are random and that prices fully reflect all available information. This perspective is encapsulated in two unproven and hotly debated theories known as the Efficient Markets Hypothesis and the Random Walk Hypothesis of asset returns. One implication of these related theories is to advocate that optimal investment strategies have a single, relatively long-term investment horizon and that excess profits cannot be systematically achieved through dynamic trading strategies.

In the last ten years, however, in apparent contradiction to the conventional view, explicit empirical studies have produced significant evidence to conclude that financial markets are to some degree predictable, and that, through maximising predictive ability, statistical forecasting models of asset returns can have predictive power. Furthermore, these forecasting models have been applied to simplistic trading strategies to produce economically significant trading, further reducing the reliability of the "perfect" market efficiency hypothesis. These empirical discoveries not only raise the issue of whether predictable models can be systematically exploited in practical trading environments, through market timing or tactical asset allocation, but also how trading should be controlled in order to optimise an investor's risk-return trade-off.

In this thesis we build upon the evidence for predictability in financial markets to develop a methodology to optimise dynamic trading strategies for statistical forecasting models. In this context, we define a trading strategy as the active management, through the buying and selling, of a subset of assets, using some model-based predictive signals or forecasts, from a wider universe of potential assets. This broad definition covers a rapidly growing group of quantitative investment management styles.

A methodological framework is developed for optimising a trading strategy for a generic form of forecasting model. The construction of an explicit forecasting model is abstracted away by assuming any model can be distinguished by a set of *characteristics* (e.g. forecast horizon, predictive accuracy and prediction smoothness) which will influence the performance of a trading strategy. These characteristics are simulated under various trading environments in order to explore a number of issues including:

- Given a predictive model, how can a trading strategy be implemented and optimised under practical trading conditions?
- How can the construction of a forecasting model with multiple characteristics be controlled to maximise trading performance?
- How can a trading system comprised of a forecasting model and a trading model be jointly optimised to maximise performance?

The methodology uses discrete time decision modelling techniques to approximate the optimal trading strategy, under different simulated trading conditions, for predicted asset returns from a single step ahead forecasting model. Where no tractable analytical or traditional numerical solutions exist, or where restrictive assumptions of traditional modelling techniques lead to sub-optimal decision making, then techniques are drawn from the field of machine learning. This leads to the development of parameterised decision rules and reinforcement learning methods to approximate the optimal trading policy.

This thesis presents a synthesis of research ideas from decision science, machine learning and specific aspects of investment finance. We demonstrate the potential of the proposed methodology on a specific class of trading strategies, based on a statistical form of relative value, which is referred to as *Statistical Arbitrage*. The underlying principles of statistical arbitrage are commonly used by many quantitative hedge funds. Methodologies are applied to practical, empirical examples of these types of trading systems using synthetic and real financial data. In general, this work is likely to be of interest to decision scientists who are interested in problems involving sequential decision making under uncertainty. The methodologies may be of particular interest of academics and practitioners in investment management especially in the field of quantitative hedge fund management and "statistical arbitrage" trading.

1.2 Motivation

In the last decade the rapid developments in computational modelling techniques, and the "almost-free" availability of financial market data has lead to the emergence of a multi-disciplined research area known as "Computational Finance". In this area, substantial research has focused on the controversy of predictability in financial assets and has motivated the development of forecasting models which attempt to expose the underlying regularities of financial markets.

The high complexity and almost stochastic nature of financial markets has spurred the development of sophisticated forecasting techniques that attempt to capture different deterministic components of market dynamics. This has led to many methodological developments in modelling non-linear and time varying relationships in non-stationary, stochastic environments. However, by concentrating on improving the accuracy of forecasting, financial modelling research has in some part neglected to consider that ultimately forecasting models are essentially just another input into the process of making investment decisions, as shown in figure 1.1.

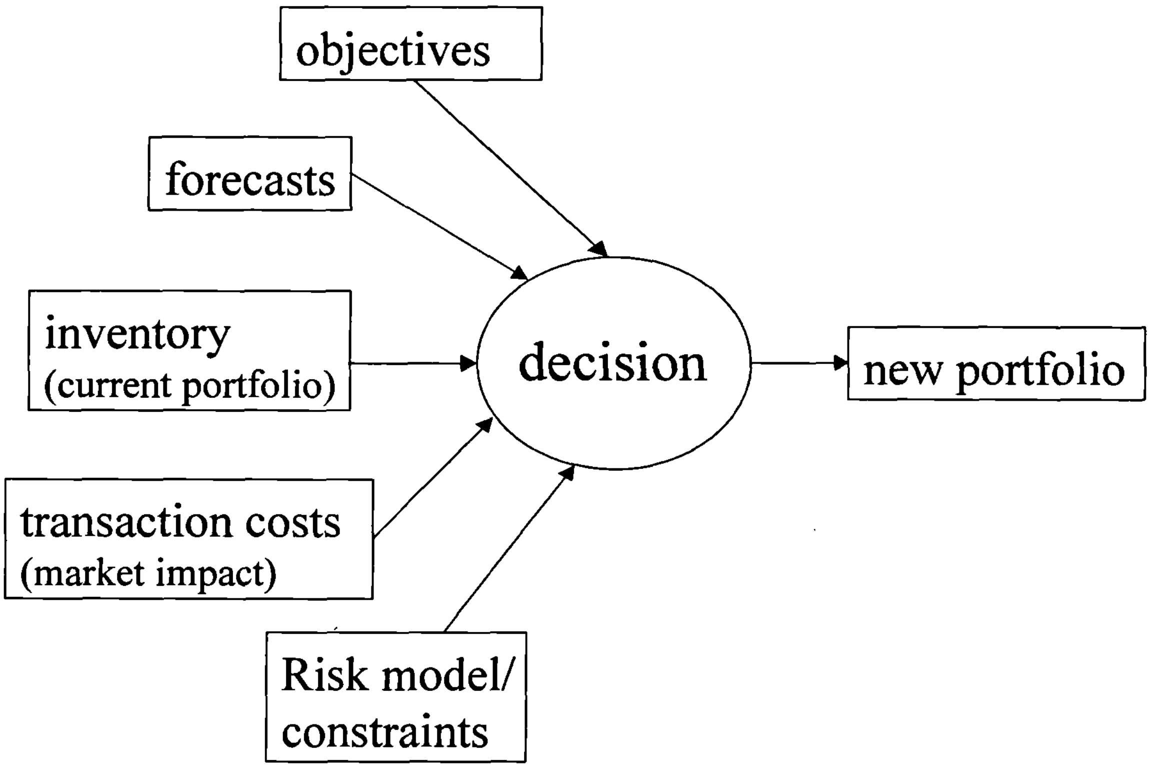


Figure 1.1 depicts the potential inputs into the investment decision-making process.

The complexity involved in developing reliable financial forecasting models has often led to a decoupling of forecasting from the optimisation of the rest of the investment decision-making process. This strong emphasis on forecasting is often confirmed in trivial implementation and optimisation of trading strategies for forecasting models. This is demonstrated by a plethora of over-simplified trading rules, whose purpose is to create "profitability" tests for predictability rather than focusing on whether genuine trading, which utilises predictive forecasts, is economically significant in practical trading environments. This thesis examines the fact that the active management of predicted asset returns involves the two inter-related modelling tasks of decision making and forecasting. We explore the methodological gaps associated with forecast model building, calibration and eventual integration into the decision making process.

The modelling of sequential decision tasks in adaptive, stochastic environments has a rich history within the Decision Science (and Operations Research) literature. This is especially true in robotics and navigation control systems where it is an active and growing area of research in its own right, with theories and modelling techniques for "intelligent" decision making emerging from the field of machine learning. It is from this perspective that we intend to explore the development of a methodological framework to optimise trading strategies for forecasting models, which can be viewed as a specific form of sequential decision making under uncertainty. Applications of control systems often incorporate models for decision making and forecasting. These modules are often viewed as separate tasks and loosely combined by two sequential modelling stages. The first stage is the building of the forecasting model, and the second is the decision phase to convert predictions into actions that, in the case of trading, controls the trading decisions through time. Thus, the overall system performance is influenced by the implementation of both forecasting and decision models.

A dynamic trading strategy can be described as a task that involves making sequences of decisions in order to improve investment performance by exploiting some form of predictable market conditions. In the case where predictability is derived from a statistical forecasting model, the aim of forecasting is to provide valuable information to assist in the active guidance of the trading position to a state with a higher expected value than a passive strategic investment position. This form of investment is complicated by the revision of the trading decisions and the inclusion of transaction costs for changing allocation. In these circumstances, even a forecasting model with a significant level of predictability may not be directly exploitable; if, for example,

- the cost of trading outweighs the expected performance gain, or
- the investment objective significantly constrains trading, or
- the trading strategy is sub-optimal.

These three interrelated issues combine to influence the significance or "value" of the predictive forecasting model to the dynamic trading strategy.

When evaluating forecasting models from a decision analysis perspective, the significance of the predictive model in the context of overall trading performance is therefore not solely dependent on the model's predictive power but also on the characteristics of the trading environment (i.e. trading costs, trading strategy and investment objectives). It is entirely plausible that a forecasting model with a relatively low level of predictive power may prove to outperform, in economic terms, competing forecasting models which although more powerful in terms of statistical predictive power, may, for example, require more frequent trading at the expense of transaction costs.

This issues may be clarified by considering the classical approach to investment management based on modern portfolio optimisation (Markowitz, 1962). Under the Markowitz assumption, the risk adjusted portfolio return is maximised, for a particular time horizon, by changing the portfolio weights given the estimates of expected return, variance and correlation of the individual asset risks. Using mean variance analysis, the expected utility of the portfolio conceptually takes the form:

$$expected\ utility = \sum w * expected\ asset\ returns - \tag{1.1}$$

$$\lambda[\sum w^2 * expected\ asset\ variance + \sum w * w * expected\ asset\ correlation]$$

where w represents the weightings of the assets in the portfolio and λ is the risk aversion parameter.

In equation (1.1) the three different expectations of the individual assets are unknown and so need to be estimated. In general, forecasting in empirical finance is concerned with modelling these three unknowns while decision making is concerned with optimising portfolio weightings given the expectations. Traditionally, the two tasks of forecasting and decision making have been encapsulated in the single equation, described in equation (1.1).

This classical approach of constructing a portfolio with maximum expected return for a given level of risk only considers a single time period. This view is considered to be "myopic" as the effects of decisions are only concerned with maximising risk adjusted return on a period by period basis, and so ignore interactions effects between periods caused by market imperfections, such as transactions costs, taxes, trading restrictions, etc. However for dynamic trading strategies in practical trading environments, decisions are effected by more general trading costs. When these factors are considered the optimal dynamic portfolio is no longer myopic but may be improved by employing multiple time periods where the longer term effects of actions are also taken in account. The sequence of trading positions is then *inter-dependent*, so a "best" decision cannot be estimated myopically (i.e. simply looking over current period) but must be determined from a decision policy (i.e. looking longer term over a sequence of actions). In this case, modelling techniques for sequential decision making need to be incorporated with the predictive information from the statistical forecasting model to approximate the optimal trading strategy.

Another important issue for dynamic trading strategies is the design of the forecasting model. In the building of any forecasting model, regardless of the forecasting method (e.g. time series regression, ARIMA or state space method), there are a number of model design factors that characterise model construction. These include, for example, the choice of the predicted time series, forecast type, model optimisation criteria, data set selection, time horizon and variable estimation/selection methodology. In most practical applications of statistical forecasting, selection of model characteristics is achieved in an ad hoc manner guided by modelling expertise. For example, the "best" forecasting model is usually constructed or selected based on some *a priori* forecast horizon with some model selection criteria (e.g. adjusted R^2, AIC), subject to the usual model integrity tests. This is assumed to provide the "optimal" predictions regardless of how decision making is implemented.

This simplistic *ad hoc* approach to forecast model construction is normally adequate for most decision tasks. For dynamic trading strategies, however, the combination of the highly complex nature of financial markets, relatively low levels of predctive ability, weak *a priori* modelling assumptions and the high sensitivity of trading performance to forecast accuracy means that naïve implementation of the forecasting model may indirectly lead to sub-optimal trading performance. For trading systems, a more general approach seems appropriate, where the multiple characteristics of any forecasting model are optimised for a specific trading strategy. This requires a *multi-objective optimisation* approach which is not part of the traditional approach to constructing forecasting models and motivates an important area of research in this thesis.

The multi-objective optimisation of financial forecasting models and the optimisation of dynamic trading strategies provides the potential to perform a joint optimisation over both forecasting and decision modelling stages to "globally" maximise trading performance. This is a significant departure from the traditional approach to dynamic trading strategies and also from the conventional methodological approach to constructing control systems using forecasting and decision models.

In summary, the implementation and optimisation of dynamic trading strategies for statistical forecasting models is a complex task and typically involves modelling stages for both the tasks of forecasting and decision making. These two tasks have traditionally been modelled separately, with financial forecasting models produced to test predictability and decision models produced from optimise trading given available predictive information. Rarely, in any application, have the interdependencies between the individual construction of the two models been combined in a general framework to optimise performance of a control system. This is the underlying motivation behind this thesis to develop a clear methodological approach to optimise trading strategies by providing a means of jointly optimising both forecasting and decision models, thereby "globally" maximising trading performance.

1.3 Thesis Overview

An overview of our methodology can be described by comparing alternative methods, as shown in figure 1.2.

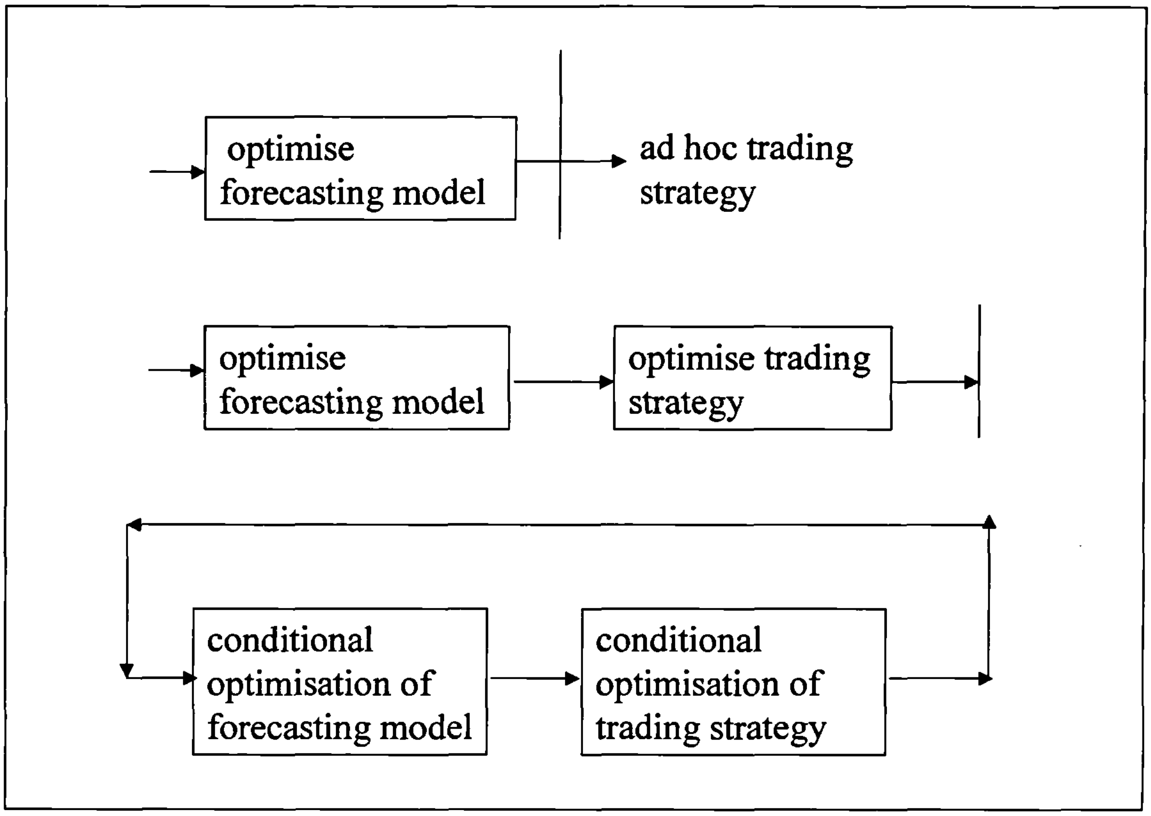


Figure 1.2 compares our methodology (bottom diagram) against the traditional "profitability rule" approach (top diagram) and a single pass optimisation method (middle diagram).

The top diagram shows the traditional approach to approximating the trading potential of a forecasting model by developing a simple, *ad hoc* trading rule, which acts as a profitability test for a forecasting model. In this approach the modelling of the two tasks, namely, forecasting and decision-making, are completed in isolation from each other. The vertical line indicates that the optimisation process finishes with the selection of the forecasting model.

The middle diagram shows a more advanced approach which optimises both the forecasting model and trading strategy. Models for the two tasks are optimised sequentially, with first the construction of the forecasting model and then, given the predicted returns, the optimisation of the trading strategy. The vertical line indicates that the optimisation process finishes after modelling the trading strategy. This approach is better than the one described by the top diagram, however, optimisation of the trading strategy depends on selection of the forecasting model with the optimal characteristics. For trading applications, a forecasting model may have multiple characteristics which influence trading performance, and so affect the optimisation of the trading strategy. However, these design factors cannot be optimised without some information feedback from the trading strategy indicating the value of each model characteristic.

The bottom diagram describes conceptually our methodology for optimising the two models for forecasting and trading. Models for the two tasks are optimised as described for the middle diagram except that information is fed back from the trading strategy to the construction of the forecasting model. This information controls the design of the forecasting model in order to optimise the forecasting model characteristics conditional on the trading strategy. The joint optimisation of the two models is performed by an iterative procedure that repeatedly optimises each task until convergence.

In our methodology, the conditional optimisation of the trading strategy is implemented using two decision-modelling techniques, namely parameterised decision rules and reinforcement learning. The parameterised decision rules provide a means of incorporating a priori knowledge about the trading policy. We show how this technique is appropriate for modelling the trading policy given predicted returns from a forecasting model in trading conditions with fixed transaction costs. The reinforcement learning approach relaxes the assumptions imposed by decision rules to approximate the optimal trading policy in more general trading conditions. These are typically found when transaction costs arise from market impact effects.

The conditional optimisation of the forecasting model is implemented using meta parameters to control the influence of a number of forecast model design factors. We experiment with forecast model characteristics and use simulation to approximate the optimal forecast horizon and two model properties, prediction correlation and predictive smoothness. Joint optimisation is shown to be achievable by repeatedly optimising the two stages in turn until the two models converge to approximations of the "true" optimal forecast and decision models.

The methodology is applied to a particular form of dynamic portfolio management referred to as statistical arbitrage trading. This typically requires the frequent turnover of large portfolios and consequently, the associated returns are highly sensitive to transaction costs and market impact. This application provides an interesting backdrop to empirically test the practical benefits of this methodology.

1.4 Contributions

This thesis builds on work from decision science, machine learning and financial engineering to provide the foundations and methodological developments to optimise trading strategies for statistical forecasting models, which is particularly relevant to the field of "Computational Finance". The contributions fall into the following areas:

Conditional optimisation of a trading strategy given predicted returns

- development of a class of parameterised decision rules for optimising trading strategies for predicted returns. Methodology based on a priori knowledge of the functional form of the optimal policy and extended to "path dependent" trading rules for optimising trading strategies in the presence of stable transaction costs.
- development of an enhanced reinforcement learning algorithm for optimising a trading strategy for predicted returns in the presence of general trading costs. In particular, the work of Neuneier (1996,1998) is extended to develop a "multi-action" Q-learning algorithm which increases learning efficiency.

Conditional optimisation of a forecasting model given a trading strategy

- development of a methodology for optimising the "design factors" of a forecasting model given a trading strategy using meta parameters.
- development of simulation experiments for trading systems in which the forecasting model has two predictive characteristics which both affect performance. We also develop experiments to investigate the affects of forecast horizon and predictive accuracy for trading systems in the presence of transaction costs.

Joint optimisation of a trading strategy and a forecasting model:

- development of iterative methodology to perform the joint optimisation of modelling stages for the two tasks of decision-making and forecasting.
- development of simulation experiments to evaluate the extent to which the isolated optimisation of forecasting and decision models can induce sub-optimal trading.

Simulations and real-world application:

- methodologies developed in this thesis are validated in controlled simulations of trading environments using data with known properties.
- methodologies are applied to statistical arbitrage trading using real-world data. Experiments are conducted on hourly data from the FTSE 100 which was collected over a seven month period. Our results demonstrate that the developed path

dependent trading rules and joint optimisation methodology can improve out-of-sample performance in the presence of transaction costs.

1.5 Organisation

In Chapter 2 we review the theoretical arguments and empirical evidence for predictive ability in financial markets, describe tests for predictability and consider the implications for developing trading strategies.

In Chapter 3 we review modelling techniques for statistical forecasting and sequential decision-making in investment finance. We describe the main classes of financial forecasting methods which are well-suited to capturing any deterministic component of asset price dynamics. We also discuss the development of decision modelling techniques and focus on techniques which are particularly pertinent to developing trading strategies for forecasting models. In addition, we review the literature on modelling techniques for trading systems and financial forecasting.

In Chapter 4 we investigate the requirements for implementing and optimising a trading system in order to exploit the predictability of a forecasting model. We discuss the development of a decision-modelling framework for a trading system and discuss two possible designs: a single model or two models, one for each task. The advantages and disadvantages of the two system designs are investigated and statistical analysis provides important motivation and justification for the two-stage approach. We also employ simulation experiments to investigate the effects of different prediction characteristics, trading strategies and transaction costs under realistic market conditions.

In Chapter 5 we present our proposed methodology to approximate an optimal trading strategy given predictions from a statistical forecasting model. First, we develop a decision modelling technique based on a class of parameterised trading rules and investigate the optimisation of trading strategies for synthetic predicted returns with controllable characteristics. We extend this approach by developing three "path dependent" trading rules which smooth the trading position in order to optimise the trading strategy for a fixed level of transaction cost. Performance improvements are explained in terms of a novel form of the bias-variance trade-off, which balances out the exploitation of predictability against the minimisation of trading costs. Secondly, we develop a sequential decision model based on reinforcement learning to optimise a trading strategy in the presence of variable transaction costs. Specifically, we develop an enhanced Q-learning algorithm which exploits the partial independence between asset prices and trading decisions. In addition, we use a neural network to act as a function approximator in order to generalise to a continuous state-action space. Simulation experiments are conducted to compare the performance of our algorithm against the optimal myopic trading strategy.

In Chapter 6 we present our proposed methodology to optimise a forecasting model conditional on a given trading strategy. We approach this problem by considering that forecasting models have a number of generic design factors that may effect the economic value of the model in the trading environment. A number of design factors are considered including the optimisation criterion, forecast horizon and forecast object. We use synthetic experiments to investigate an optimisation criterion consisting of two characteristics, namely predictive correlation and prediction autocorrelation. We show, using a *meta* parameter, how this criterion may be optimised with respect to a trading strategy to improve trading performance. In addition, we develop simulation experiments to examine the effect of the forecast horizon for a given trading strategy. The results show that the optimal forecast horizon depends on the level of transaction costs.

In Chapter 7 we present our joint optimisation methodology which encompasses the modelling approaches, developed in chapters 5 and 6, for the tasks of forecasting and trading. The proposed method achieves joint optimality by repeatedly alternating between the two modelling stages until convergence. We discuss this process by examining methods which optimise two interdependent tasks by repeatedly alternating between the two processes to achieve joint optimality. We illustrate, using synthetic experiments, how the iterative methodology can be used to control the optimisation criterion of a forecasting model with two characteristics which both influence trading performance. The results demonstrate how performance may be improved by using our joint optimisation methodology in the presence of transaction costs. In addition, we discuss the application of advanced optimisation techniques and use simulation experiments to show how we may optimise the forecast horizon.

In Chapter 8 we apply our joint optimisation methodology to the development of statistical arbitrage trading strategies for predictive models. We discuss the development of a two-stage trading system for a statistical mispricing which is comprised of a forecasting model and a trading model. We conduct extensive empirical analysis of a realistic set of 50 statistical mispricings which are identified within the UK equity market (FTSE 100). Trading systems are optimised and evaluated for different levels of transaction costs. The results demonstrate that parameterised trading rules and joint optimisation can significantly improve trading performance in the presence of transaction costs.

2 Evidence of Predictability in Financial Markets

In this chapter we provide a concise review of the evidence for predictability in financial markets. For more than a generation, the topic of market efficiency has been of widespread interest and the focus of considerable research. This issue has been hotly debated and primarily tested in the form of two extensively studied theories, the random walk and the efficient market hypotheses. In light of this, we do not intend to provide an exhaustive review of market efficiency or even take a stand on the topic itself, but rather focus on the key findings and arguments that are central to the debate and review the statistical methods that have been used to test and measure efficiency.

In section 2.1 we provide an overview of the market efficiency debate and summarise the orthodox view which is epitomised by the random walk and efficiency market hypotheses. In section 2.2 we review the theoretical arguments that have been used to defend and subsequently criticise the orthodox view. In section 2.3 we briefly review the empirical research that has been undertaken to validate the random walk and the efficient market hypotheses. In section 2.4 we discuss the development of tests for predictability and demonstrate how these may be used to go beyond testing market efficiency. In section 2.5 we summarise the main findings and conclude that there is some evidence of predictability in financial markets and that this provides ample justification to investigate the optimisation of trading systems based upon forecasting models of financial assets returns.

2.1 Overview

For over 30 years, researchers have deliberated over the predictive nature of financial markets and proposed theories to explain the underlying mechanism that determines the price dynamics of financial assets. Research has actively focused in searching for predictability in asset returns, with motivation arising from an economic interest in understanding how fluctuations in the economy influence financial markets and also from a practical interest in seeking ways of improving the financial rewards of investing.

Even after this period of much heated debate, no consensus has been reached amongst academics which conclusively explains the behaviour of asset prices. The dominant (or orthodox) view, however, is epitomised by two related theories known by different forms of the random walk hypothesis of asset returns and the efficient market hypothesis. The former theory assumes that asset returns are completely stochastic in nature while the latter theory implies that profit opportunities do not exist in perfectly efficient markets. In essence, these two theories imply that in well functioning markets, asset prices are unpredictable and fully reflect all available information. To allow for practical market conditions, the efficient market hypothesis has been modified for trading costs so that a market price only fully reflects available information to the extent that trading would not exceed the marginal benefit.

In the last ten years, however, theoretical arguments and empirical research have seriously questioned these two theories but there is still no general agreement over the validity of both the random walk and the efficient markets hypotheses. In the next two sections we present an overview of the main controversies over the efficient market hypothesis, stock market rationality, and the existence of systematically profitable trading strategies on the basis of theoretical arguments and empirical research. We include arguments for and against these theories, and provide a concise review of the most compelling empirical research.

2.2 Review of Theoretical Arguments

Initial applications of the random walk hypothesis to financial markets perceived randomness in asset returns as due to large groups of investors continuously seeking ways of increasing wealth (Samuelson, 1965). This linked with the Efficient Market Hypothesis by arguing that perfectly efficient markets would generate asset price changes that are perfectly random. In this view, investment decisions are directly driven by new information which is rapidly incorporated into market prices. The result of this process is to eliminate any opportunities for systematically profitable trading. Furthermore, it was argued that in perfectly efficient markets with no trading costs, these potential opportunities would disappear immediately as prices would instantaneously reflect all available information, so that no information-based profitable trading strategy could exist and thus asset returns must be unpredictable. At first glance these two related theories seem to adequately explain the functioning of financial markets and provide an elegant link to microscopic processes found in physics and other natural sciences. However this unified framework for understanding the dynamics of financial markets has been questioned by a number of authors.

The first most notable critics based their arguments upon explicit examples of financial markets where the Random Walk Hypothesis does not hold (LeRoy, 1973; Lucas, 1978). They showed how this breaks the conceptual link between the two theories. They argued that unpredictability does not imply an efficient market, and conversely, that predictability does not guarantee a poorly functioning market with irrational investors. More aggressive criticisms were put forward by Grossman (1976) and Grossman and Stiglitz (1980) who argued that it is impossible for perfectly efficient markets to exist as there would be no justification for acquiring information. The current price would already incorporate all current information value so there would be little reason to trade as no excess profits can be made. In these circumstances they argued that even small trading costs should cause financial markets to degenerate and eventually collapse!

To defend the Efficient Market Hypothesis from such severe criticism, Black (1986) suggested that financial markets must contain inefficient traders, described as noise traders, who speculate on spurious information and trade, not based on new market information, but for other reasons, for example, due to liquidity requirements. He proposed that these trades must lose on average and therefore provide sufficient profitable opportunities for information-based investors to counterbalance the costs of trading and collecting information. This theoretical efficient market is now non-degenerative, however, in practice, there is nothing inherently inefficient about trading on the basis of liquidity requirements or for any other reason. In addition, no evidence is presented to show the functioning of this delicate balance between noise and information-based traders and the costs of trading and collecting data.

It appears that the main reason for the heated debate amongst financial economists is the idealised nature of the Efficient Market Hypothesis which means that in practice it cannot be proved or refuted. In order to adequately test the theory researchers have attempted to expand the hypothesis to take into account other factors such as trading conditions, information structure, dividends, etc. These changes qualify the original hypothesis, which has the effect of turning it into a conditional or joint hypothesis. These are often difficult to refute which means that it is difficult to reject market efficiency.

As an alternative, a number of revisions to the original efficient markets hypothesis have been proposed. The most notable, the *relative efficient market hypothesis*, states that "the efficient markets hypothesis should be linked with bounded rationality so markets are only efficient and unpredictable with respect to existing available information and not all possible sources of information" (White, 1988). This allows the potential for new information to be exploited to generate excess profits and overcomes the costs associated with acquiring information. It is considered that new sources of information may include advanced modelling techniques that are only available to specialist market participants. This would provide the motivation to drive research into financial forecasting techniques without seriously conflicting with the efficient markets hypothesis.

A more elaborate argument for the relative efficient markets hypothesis, states that "is it not possible to systematically earn excess profits without some sort of competitive advantage (e.g. superior technology, proprietary information, advanced methodology) over other market participants" (Lo, 1999). This definition is analogous to the trading of products in other, non-financial industries, where patents provide a time frame for generating excess profits until some expiry date, at which other market participants are allowed to copy the innovation. However, financial markets have a number of significant differences compared to most other non-financial industries. These are mainly due to the large number of market participants, the intense competition, low entry barriers and the fact that innovations are, in general, not patentable. It is thought that these factors have the effect of making financial markets *relatively efficient*, while the potential rewards associated with further innovation provide sufficient motivation for competitive advantages to be developed. It is considered that this theory closely matches the functioning of financial markets and suggests that some of the technological advances offered by sophisticated modelling techniques provide sufficient motivation to allow specialist traders to have the potential to generate excess profits.

From the perspective of this form of the relatively efficient market hypothesis, it is anticipated that the dynamics of financial markets will change through time as competitive advantages become more widely available to other market participants. In this case, it is likely that the widespread acceptance of any competitive advantage would result in a decline in the associated profits. This would inevitably result in market dynamics moving towards randomness with respect to an old form of competitive advantage. This tallies with the experience of "old" riskless arbitrage strategies that at one time offered a systematic and reliable means of providing profitable trading opportunities but now only offer marginal trading potential.

It is interesting to note that the relative efficient market hypothesis implies that financial markets must have some degree of predictable behaviour and that this predictability is not a symptom of market inefficiency or irrational investors. This is aptly described by Lo (1999) who says that, "*predictability is the oil that lubricates the gears of capitalism*" and thus is an essential ingredient in financial markets. The implication of this revised theory is that predictability will always exist in financial markets but to harness this rare and valuable commodity requires a continuous stream of technological advancements in order to keep one step ahead of most other market participants. In chapter 3 we review the main groups of modelling techniques which can be applied to financial forecasting and trading. Our underlying philosophy is that these or related methods have the potential to be applied to investment decision making and are capable of capturing and harnessing some degree of market predictability.

2.3 Review of Empirical Research

After the random walk and efficient market hypotheses were presented in the 1960's, empirical research began to try an establish the validity for the these theories and so test the viability of typical trading strategies. A vast literature has accumulated over the last 30 years and we present an overview of the most notable empirical studies that have attempted to test for predictability and profitability in financial markets. The random walk hypothesis implies that no statistical regularities should exist in well-functioning financial markets. To test the Efficient Markets Theory (EMT) specific information sets are analysed which has resulted in three forms of the EMT: weak, semi-strong and strong. The weak form of EMT considers information solely from historical prices, the semi-strong form considers all publicly available information and the strong form considers all private and publicly available information.

Most empirical tests have concentrated on the weak form of EMT and these fall into two categories. The first conducts statistical tests on historical data to identify significant patterns and the second examines technical trading rules to determine whether they generate abnormal returns after trading cost. In one sense, these studies are extremely prohibitive given the external forces that are commonly believed to influence asset prices, but provided a good method of analysing market behaviour.

One particular topic which was the focus of much initial work considered analysing the effects of *seasonality* from different days of the week or months of the year for any predictable behaviour. One of the first anomalies of the weak form of EMT was identified as the weekend effect (French, 1980). For example, one study by Gibbons (1981) analysed daily closing data from the New York Stock exchange, and found that Monday's return was significantly lower than other days of the week. A 17 year period (1962-1978) was examined where the annualised Monday return was -33.5%! In a more recent, related study, conducted by Harris (1986), the intra-day and day effects from a 14 month period (1981-1983) were examined and found that the effect was primarily due to the difference between Friday close and Monday open.

In a number of other studies, seasonal behaviour was investigated in monthly stock returns. Fama (1991) examined monthly returns over a 50 year period (1941-1991) on the New York Stock exchange, and discovered that returns in January were substantially higher than returns in other months. The examination of returns showed that returns on small stocks outperform the monthly average by 5.3%. Another comprehensive study examined a wide range of international equity markets and found the existence of a significant *January effect* in 17 countries (Gultekin et al.,1983).

In order to explain the January effect, researchers have proposed various explanations, one of the most notable is that tax advantages could produce market anomalies in January (Kato and Shallheim, 1985). However, subsequent work has discounted the tax selling hypothesis by still finding the January effect in some tax exempt markets (Jones et al., 1987). In other related studies, which analysed trading rules that exploit the January effect, evidence suggests that assets purchased in December and sold at the end of January outperform the market by approximately 8% on average (Reinganum, 1983). The general conclusion from these studies is that the January effect cannot be reconciled with the theoretical concept of an efficient market. It is interesting to note if a sufficiently large number of investors purchased equities in December to anticipate price rises in January then any price rise should occur earlier. Clever investors would then buy in early December to anticipate these price rises, which in turn would lead to even earlier price rises. In principle, the January effect would then move earlier as move investors try to anticipate price rises until the effect is eliminated.

Although these and other similar seasonal effects are well known, and conclusively show that returns are systematically dependent on the time of day, week and even the month of the year, there is little practical evidence that these statistical regularities can and are being exploited for trading. To maintain the efficient markets hypothesis, most proposed explanations have suggested that the observed seasonality is due to artefacts in the data or where transaction costs are too high to counteract any benefits from trading these time effects.

In another area of early research, simple tests for predictability were conducted to examine whether past returns could forecast future returns. A number of these studies have examined the first order autocorrelation between returns over time intervals, ranging from 1 day to 3 months, and over various stock markets (e.g. Fama, 1965; Cootner, 1974). In general, however, results have shown no significant correlations and studies argue that correlations should not be used to examine the efficiency of markets because of the influence of the outlying observations (Fama, 1965; Jennergren and Korsvold, 1975). Other non-parametric tests have been proposed, for example, the Runs Test, which counts the sign of consecutive returns, however, in empirical experiments these tests have only found small positive relationships between returns.

In contrast to these negative results, the random walk hypothesis has been tested by comparing *variance estimators* over different time frequencies (Lo and MacKinlay, 1986). Results from weekly stock returns over a 23 year period (1962-1985) strongly reject the random walk model for various aggregated indices and portfolios. This is primarily attributed to the behaviour of small stocks and not infrequent trading or time varying volatilities. They also showed that the Variance Ratio test is a more powerful test of the random walk model than the traditional Dickey Fuller and Box Pierce tests (Lo and MacKinlay, 1989).

Other tests for predictability have taken a different view by examining *relative* stock returns by subtracting the related market index from the absolute stock returns. Studies have used the CAPM model or other similar models to examine the correlation of these "excess" returns (e.g. Fama and MacBeth, 1973; Galai, 1977; Roll 1970). Results from these experiments, however, show no significant predictive correlation and so no evidence against the efficient market hypothesis.

Another group of empirical studies advocate searching indirectly for *non-linear relationships* in returns data by developing complex trading rules based on historical price movements. A common example is a breakthrough barrier trading rule (Fama and Blume, 1966; Jennergen and Korsold, 1975). This formulates a trading strategy that sells when the asset price breaks through a lower price barrier and buys when the asset price rises above a upper price barrier. Results have shown some evidence of profitable trading rules but profits often disappear with practical trading costs. Other more esoteric rules, such as rules for "head and shoulders" patterns and relative strength rules have been devised (Levy, 1967). A number of similar approaches are often advertised and supported by technical traders or chartists but evidence that trading on the basis on these results can generate excess profits is generally considered inconclusive. However, these forms of investment are similar to empirical research amongst finance academics which has shown some evidence of predictability (recent examples include, Brock, Lakonishok and LeBaron, 1992; LeBaron, 1996).

Other studies investigate forms of cyclical behaviour in financial markets with the underlying assumption that predictability can be attributed to periods of under or over value. The belief is that market participants are susceptible to cycles of investor optimism and pessimism which result in asset prices temporarily moving away from their "true" value. These forms of market dynamics have been described by the "stock market overreaction" hypothesis (DeBondt and Thaler, 1985; Delong et al, 1989; Lehmann, 1988). The theory implies that asset returns are negatively correlated for some holding period which provides predictability in stock returns. This theory has been tested using the contrarian investment rule, in which stocks that have recently increased in value are sold and stocks that have recently decreased in value are purchased. Results from these studies seems to suggest that stock markets do overreact.

Other studies have examined the cross-autocorrelation structure of securities and empirical findings have shown a lead-lag structure between large capitalisation and small cap stocks producing a source of positive dependence in stock returns (Cohen et al, 1986; Lo and MacKinlay, 1990). However, results may also be due to "thin" or nonsynchronous trading between stocks where prices are mistakenly sampled simultaneously, which was first identified by Fisher, (1966). Experiments have been conducted to examine the magnitude of index autocorrelation and cross correlation generated from models of thin trading. Results show that although some autocorrelation is induced the observed correlation structure would require unrealistically thin markets (Lo and MacKinlay, 1999). For a review of numerous other studies and a thorough examination of empirical research into predictability in asset returns, see "The Econometrics of Financial Markets" by Campbell, Lo and MacKinlay.

As an alternative approach, a growing number of more recent empirical studies have developed time series models using statistical forecasting techniques in an attempt to capture any deterministic component of the underlying dynamics of financial markets. These methods form another area of intensive research which is reviewed later in chapter 3, section 3. However there is a growing number of researchers who indicate that statistical forecasting methods have the potential to capture any predictability in the functioning of financial markets which may be effectively harnessed to produce profitable trading.

In summary, this section does not attempt to be an exhaustive review of empirical research of tests for market efficiency or predictability. Its purpose is to show that there a considerable and growing evidence that financial markets may not be truly efficient and that some degree of significant predictability may be identified through statistical analysis.

In the next section we review some of the tests for predictability on the basis of forecasting the future price changes as some function of past price changes. Although this approach is restrictive it provides a framework for constructing forecasting models from past data. We also show how these tests have been applied to detect predictability for more general forecasting models where additional economic variables are used in the modelling process.

2.4 Predictability Tests

The random walk hypothesis of asset prices can be decomposed into three different versions depending on the strength of the underlying assumptions. The strongest form of the hypothesis assumes that asset returns have independently and identically distributed increments. The assumption of independence implies that increments are uncorrelated and also that nonlinear functions of the increments must be uncorrelated. In the case of equities, asset prices are always positive so this definition requires modification. It is usually assumed that log normal prices follow a random walk with normally distributed increments. If asset prices did follow this hypothesis then it would form an elegant analogy with stochastic processes in the natural world.

However, this model of asset prices is not very realistic when compared with the past behaviour of markets. For example, in the long term, markets adapt and evolve through time as economic, political and structural changes take place. This can be seen by observing the time varying variance (heteroscedasticity) of asset returns. This model of identically distributed increments is too strong an assumption to apply this version of the random walk hypothesis to financial data. We therefore do not consider any tests for this version of the hypothesis.

The second form of the random walk hypothesis assumes that increments are independent but not identically distributed. However it appears to be extremely difficult to statistically test for independence without assuming identical distributions. Some non-parametric tests have been developed (e.g. rank tests) but these still require some restrictive assumptions about the distribution. The lack of powerful statistical tests for this version of the random walk hypothesis has led too much empirical research to develop "economic" tests of predictability. The aim of these is to indirectly measure predictability by analysing the performance of a simple trading strategy for the predicted asset return.

In general, economic tests of predictability have taken the form of a simple filter rule that is applied to the asset return series. The total return generated from a dynamic trading strategy implemented on the basis of the trading rule is then considered to be a measure of the predictability of the asset returns. Empirical experiments have been conducted to allow for effects of the market microstructure (e.g. dividends, trading costs, etc...). The results from initial experiments concluded that such rules do not perform as well as a simple buy-and-hold strategy (Fama, 1965; Fama and Blume, 1966).

However, more recently, similar economic tests have been extended to forecasting models of asset returns. The rule forms a simple trading strategy to provide an "economic" measure of predictability by exploiting the predictive nature of the forecasting model. One simple but plausible strategy buys a fixed amount of the asset if the prediction is positive and sells an equal amount if the prediction is negative. This naïve asset allocation rule was first attributed as Merton Measure of Market Timing (Merton, 1981) to provide a simple measure of out-of-sample predictability in terms of trading profitability. Recent studies suggest that simple trading rules used in conjunction with a forecasting model of asset returns may be used to outperform a benchmark (e.g. Refenes, 1995).

In other studies, more advanced forms of trading rules have been developed which have formed into a class of investment management known as *technical analysis* and *charting*. The underlying assumption of these trading methods is that historical prices, trading volume and other market statistics exhibit regularities which form patterns such as head and shoulders or support levels that can be profitably exploited to extrapolate future price movements. These can be tested in a similar manner using economic tests to justify the modelling approach.

The third and weakest version of the random walk hypothesis assumes that asset prices may have dependent but uncorrelated increments at all possible leads and lags. This is tested under the null hypothesis that all autocorrelation coefficients are zero. There are a number of simple but powerful tests that have been developed to test this hypothesis and these are reviewed next.

2.4.1 Autocorrelation Test

The most obvious test for this weak form of the random walk hypothesis is to directly test the null hypothesis that the autocorrelation coefficients of the increments (or first differences) are zero. This is achieved by estimating the autocorrelation coefficients and their associated p–values for a given sample.

These sample statistics, however, can sometimes suffer from biases caused by outliers or the estimation of the sample mean. The construction of correlation coefficients requires the measurement of deviations from the mean and a poor estimate can cause a negative bias in the autocorrelation. However, bias corrected sample autocorrelation coefficients have been presented by Lo and MacKinlay, (1988).

The main deficiency of the autocorrelation test however is that it can only detect predictability associated with a particular lag and not test autocorrelation within the entire time series and so are highly sensitive to noise. To compensate for this weakness, other statistics have been developed, sometimes known "portmanteau" statistics, which are joint tests over the set of individual correlation coefficients. In the next subsections we focus on Q-statistics and the Variance Ratio test.

2.4.2 Q-Statistic

A more powerful test of the random walk hypothesis is the Q-statistic initially developed by Box and Pierce (1970). The Q-statistic at lag k is a test statistic for the null hypothesis that there is no autocorrelation up to order k and takes the form:

$$Q_k = T\sum_{i=1}^{k}\hat{\rho}_i^2 \tag{2.1}$$

where ρ is the sample autocorrelation at lag i and k is the number of lags.

The Q-statistic for k lags is defined as the sum of the first k squared autocorrelations, which is is asymptotically distributed as a χ^2 distribution with degrees of freedom equal to the number of autocorrelations. The test was further refined for small samples by Box and Lung (1978) to account for the number of observations.

The Q-statistic is often used as a test of whether a time series is white noise. It differs from the autocorrelation test by attempting to identify statistically significant predictability by considering the deviations of all autocorrelations from zero. This is a more powerful test as predictability held within a number of lags may not be identified by examining the correlation at one particular lag.

However, there are practical problems associated with this test. Most notable is the selection of the order of the lag. If you choose too small a lag, the test may not detect serial correlation at high-order lags. However, if you choose too large a lag, the test may have low power since the significant correlation at one lag may be diluted by insignificant correlations at other lags. For further discussion, see Ljung and Box (1979) and Harvey (1990, 1993). As this test does not consider predictability at a specific lag it is often used in conjunction with the autocorrelation test.

This statistic also requires modification when applied to statistical forecasting models. For example, if the series represents the residuals from ARIMA estimation, the appropriate degrees of freedom should be adjusted to represent the number of autocorrelations less the number of AR and MA terms previously estimated. In addition, care is needed in interpreting the results of a Ljung-Box test applied to the residuals from an ARMAX specification (see Dezhbaksh, 1990, for simulation evidence on the finite sample performance of the test in this setting).

2.4.3 Variance Ratio Test

A more recent test that is gaining widespread acceptance is known as the Variance Ratio Statistic. For a timescale τ, which defines the time period between innovations, the variance ratio statistic, $VR(\tau)$, takes the form:

$$VR(\tau) = \frac{Var(r_t\ (\tau))}{\tau Var(r_t(1))} \tag{2.2}$$

where r_t is the asset return at time t.

This test exploits the property that the variance of random walk increments must be a linear function of the time interval and result in a ratio close to one. This property has been shown to hold even when applied to the variance of increments that vary through time (Campbell et al., 1997). Extensive studies using the Variance Ratio test have been completed by Lo and MacKinlay (1996), which provide clear evidence that stock markets exhibit non-random walk behaviour.

The variance ratio is directly related to the autocorrelation coefficients and is equivalent to

$$VR(\tau) = 1 + 2\sum_{i=1}^{\tau-1}\left(1 - \frac{i}{\tau}\right)\rho_i \tag{2.3}$$

where ρ is the ith order autocorrelation of the increments of some time series. Equation (2.3) shows that the VR statistic is a weighted combination of the first τ-1 autocorrelation coefficients.

The test for the random walk hypothesis assumes that the variance ratio of any period must be close to one. If the $VR(\tau)$ is significantly greater than one then trending behaviour is detected in the time series. In contrast, if the $VR(\tau)$ is significantly less than one then mean reverting behaviour is identified. The variance ratio statistic has been used for predictability tests to distinguish a variety of deviations from random walk behaviour. For example, the variance ratio statistic has also been extended to a multivariate context and used to identify the dynamics of cointegrating time series related to trending and mean reverting behaviour (Burgess, 2000). In this work, empirical studies of FTSE 100 stocks have been found to contain significant evidence of non-random behaviour.

2.5 Beyond tests for market efficiency

So far, we have only considered the investigation of tests for market efficiency in financial markets. In this section we show how predictability tests may be adapted for purposes beyond tests of random walk behaviour to assist the detection of predictability and form the initial part of a methodology to optimise a trading system comprised of forecasting models.

In financial forecasting applications, there are often a wide number of candidate target series which may be modelled at a range of time steps. The large number of potential time series often make the process of model building complex and highly computer intensive. This problem is compounded by searching for predictability in time series comprised of combinations of assets. Under these conditions the number of candidate time series can grow exponentially within the number of assets. It is then often impractical to construct forecasting models for each possible time series and screening is imposed to filter out series with no detectable predictability.

In the context of trading, the detection of some level of predictability is not the ultimate goal but rather just one process in the construction of the trading system. This process may be considered as an initial screening stage that selects asset returns with a predictable component from a pool of candidate assets and so suitable for developing statistical forecasting models. Further stages may be developed to optimise models to forecast asset returns and select optimised trading positions. In the context of this thesis, the overall process may be described conceptually as shown in figure 2.1

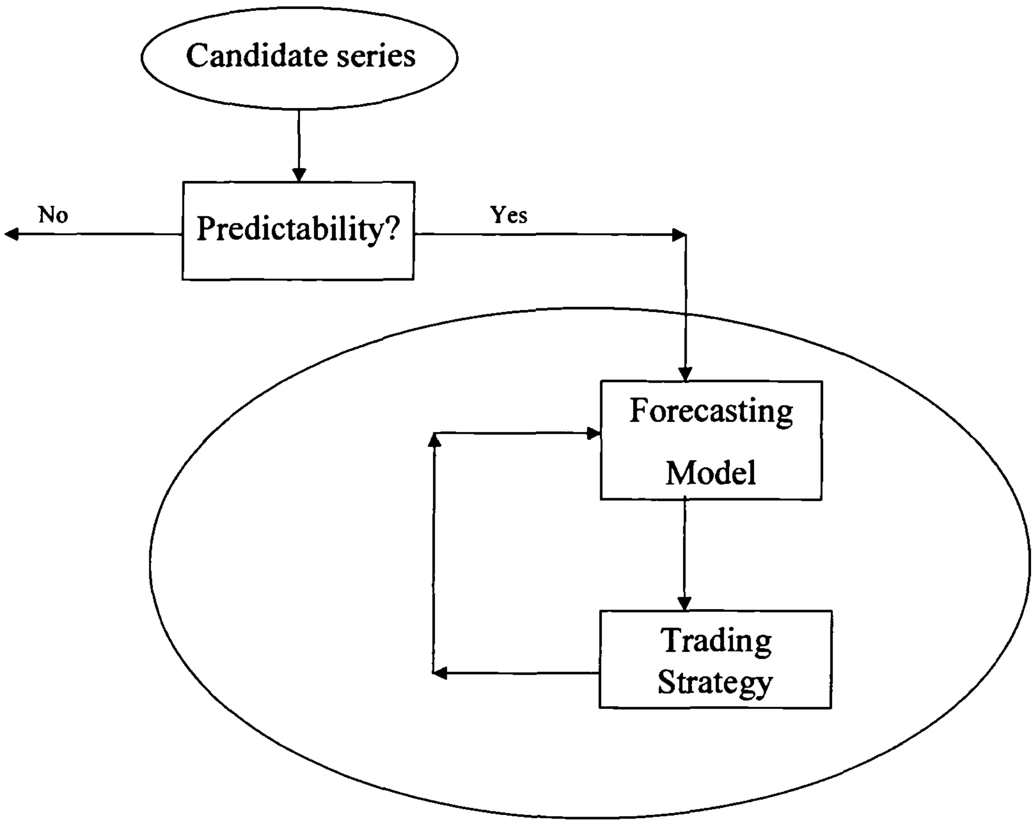


Figure 2.1 depicts a use of predictability tests as an initial screening stage within the construction of a trading system comprised of forecasting and trading models.

In figure 2.1 the candidate series refers to the return series drawn from a candidate set of financial assets. These may form the object of a forecasting model which aims to maximise predictability (or more general forecast characteristics) rather than just identify predictability. The predictability test simplifies the selection of target series from the large pool of candidate assets. It may also be used to choose the forecast horizon of the forecasting model on the basis of the significance of the predictability tests. At this initial screening stage, if no predictability is detected then the asset is discarded.

This conceptual approach has been used by methodologies in financial forecasting that have a large set of candidate target variables. For example, a recent modelling methodology by Burgess (1999) adapts the Variance Ratio statistic to provide novel tests for predictability for financial asset returns and combinations of assets. He shows how these statistics provide a practical means of identifying potential time series which may be candidate target variables of a forecasting model.

2.6 Summary

In this chapter we present a concise review of the evidence for predictability in financial markets on the basis of both theoretical arguments and empirical research. We describe tests for predictability and discuss how they may be used to go beyond market efficiency and form the initial part of a methodology of optimise trading systems comprised of forecasting models of financial assets.

We present a chronological review of research which records a growth in proponents of predictability in financial markets. Evidence for predictability has emerged against the backdrop of a classical view which strongly upholds to the random walk model of asset returns. The empirical evidence for predictability has led to revised theories of the dynamics of financial markets which is encapsulated in the relative efficient market hypothesis. In this view predictability is not only plausible but is a necessary component of financial markets.

If we accept the hypothesis that predictability exists in financial markets then it seems reasonable to ask how this influences market efficiency. The first point to make is that market inefficiency is not necessarily a consequence or a symptom of market predictability and so predictability may not imply that markets are inefficient. The second is that the properties of predictability (i.e. degree, sources and forecast horizon) can be transient so often require sophisticated modelling techniques to capture sufficient predictability for models to be statistically and economically significant, both in-sample and out-of-sample. The final point is that a true test of the significance of a statistical forecasting model requires the incorporation of predictive information into a trading strategy. The performance of the strategy then needs to be tested on out-of-sample or ideally real time data to determine whether trading produces "excess" profits.

The most convincing empirical evidence follows from the underlying assumption that financial markets are dominantly stochastic and only partially deterministic in nature. In addition, the construction of statistical models by identifying interesting linear combinations of stocks and bonds appears to offer the most reliable method of searching for predictable behaviour in financial markets. In general these forecasting models with model parameters, θ, can be described as

$$E[\boldsymbol{\beta}\Delta\mathbf{y}_{t+1}] = f(\mathbf{X}_t;\theta) \tag{2.4}$$

where X is some vector of explanatory variables, β is the portfolio weights, and Δy is the returns of the assets in the portfolio.

In equation (2.4) the explanatory variables depend on the methodology for identifying predictive ability and can represent ex ante macroeconomic variables (Lo and MacKinlay, 1997), cointegration residuals (Burgess, 1998), or CAPM residuals (Bentz et al., 1996). One specific application of this approach is "statistical arbitrage" which exploits statistical relationships between asset prices and is explored in more detail in Chapter 9 where we apply decision technologies to optimise dynamic trading strategies in the presence of realistic trading conditions.

It is worth noting that the topic of predictability in financial markets has wider implications than are discussed within this thesis. For example, in derivative pricing most models assume that asset returns are unpredictable and that prices are described by drift and volatility processes. If stock returns are predictable then the traditional Black Scholes equation for pricing options is misspecified, as suggested by Lo, 1999. This is another active area of research but outside the scope of this thesis and so is not discussed further.

3 Modelling Techniques for Forecasting and Decision-Making

In this chapter we provide a review of modelling techniques that can be applied to forecasting and decision-making tasks in investment finance. In recent years, this topic has been of considerable interest and has attracted intensive research, so we restrict our attention to modelling techniques that are relevant to trading predictability in financial markets.

In section 3.1 we review the development of quantitative techniques in investment finance. In section 3.2 we review four main classes of forecasting methods that are well suited to capturing any deterministic component of asset returns. These classes are smoothing techniques, ARIMA and time series regression models, cointegration and error correction models and adaptive, state space models. In section 3.3 we chronologically review the development of financial forecasting models which have been applied to empirical data. In section 3.4 we introduce the topic of sequential decision modelling and focus on techniques that are particularly pertinent to developing trading strategies such as decision rules and reinforcement learning. In section 3.5 we review the development of modelling techniques to optimise dynamic trading systems.

3.1 Overview

In the last three decades, the rapid expansion of global financial activity, accompanied by growth in computer technologies and the real-time provision of broadcast financial data has had a profound impact on the functioning of financial markets. Market participants are now faced with overwhelming quantities of information and complex choices amongst financial instruments. These changes are gradually affecting the culture of investment management with an increasing reliance on quantitative modelling techniques in the hope that these offer a more principled, "scientific" approach to management. It is widely anticipated that quantitative investment techniques will play an increasing important part of investment management in the years ahead.

In the past 15 years, quantitative techniques have moved from the radical fringe, based on merely intriguing theoretical ideas, to the mainstream of investment finance. To underline the extent of "quant" modelling, Robert Arnott, senior partner in one of the world's largest quantitative managers (First Quadrant), states that "perhaps as much as a half-trillion dollars is committed world-wide to strategies wholly or largely based on quantitative techniques" (Bernstein, 1998). During the last decade, the shift in emphasis to quantitative tools in investment management has opened the way for the development of advanced techniques in both forecasting and decision making using computational and statistical modelling methods. This has led to a number of highly regarded, international journals and conferences in the field of Computational Finance which serves the growing interest amongst both practitioners and academics.

Modelling techniques in investment finance can be broadly broken down into two groups based on the *uncertainty* associated with the underlying assumptions of the asset pricing dynamics. One group assumes a precise understanding of the underlying dynamics, thus enabling pricing to be calibrated and decisions optimised with the assumption of *no uncertainty* (e.g. option pricing, bond pricing and arbitrage). The other group, however, performs asset valuation and decision-making *under uncertainty* (e.g. mean-variance optimisation, risk analysis, equity valuation and statistical arbitrage).

The first group primarily uses a unified theoretical framework based on arbitrage-free pricing theory to value and select of a wide range of financial instruments, including derivatives and fixed income securities, and also for identifying "risk-free" arbitrage opportunities in foreign exchange, fixed income and equity markets. Examples include option pricing using the Derivative Pricing Model (Black-Scholes, 1973) and pricing fixed income securities using yield curve models, such as the Heath Jarrow Morton model (Heath, Jarrow and Morton, 1990). These models determine the arbitrage-free prices of financial instruments, in frictionless and competitive markets, where there is no counterparty risk, by constructing a synthetic asset to generate the theoretical value of the security. Pricing is based on strong theoretical assumptions of underlying pricing dynamics and cash flows and these techniques are typically used to identify arbitrage opportunities or for the hedging of instruments to manage risk exposure.

The second group combines weaker theoretical assumptions with the statistical analysis of empirical data to assess asset valuation and the future performance of decisions *under uncertainty*. The existence of uncertainty means that there is more than one possible outcome to an investment decision, so the return is subject to some degree of risk. Quantification of the risk associated with an investment opportunity implies some expected return distribution. Models have been developed to structure investment decisions, for either individual securities or a portfolio of assets, by controlling the *expected risk adjusted return* in order to find the most desirable assets to hold, given the properties of the individual assets and the investment objectives. These models include Modern Portfolio Theory (Markowitz, 1962), Capital Asset Pricing Model (Sharpe, 1963), and Arbitrage Pricing Model (Ross, 1976). These models assess investment risk and return in different ways, by analysing the combined risk of a portfolio, the individual asset risk relative to a market index or relating the sensitivity of asset price movements to underlying macroeconomic factors respectively. Applications of these different models in investment management typically include stock selection, asset allocation and performance monitoring. This type of investment modelling relies heavily on the statistical analysis of historical price data to estimate future asset behaviour in order to add practical value to investment decision making. The highly stochastic nature of asset price behaviour limits *a priori* modelling assumptions and raises the prominence of statistical inference from past data. The dependence on statistical techniques provides the basis for potential advantages from advanced statistical and computational modelling techniques which assist with the tasks of forecasting and decision-making.

For illustration purposes, consider the introduction of forecasting models into portfolio optimisation using a mean variance framework from Modern Portfolio Theory. The approach can be broken into two stages: first the analysis of the individual securities to quantify likely future asset return and risk and second, the formation of the optimal portfolio given the investment objective. In the classical approach, estimates of the future return and risk of the individual assets are quantified under the assumption that past returns are the best representation of the future expected return distribution. In this case, two statistics capture the attributes of the return distribution from the historical data, namely the mean and the standard deviation[1]. In addition to these two statistics for expected asset return and risk, the relationship between the returns of two assets is estimated using the covariance statistic, which measures the linear association between two return series. These traditional estimates of the expected mean, variance and correlation of asset returns can be traced back to the ideas of the efficient market hypothesis where it is assumed that all available information is incorporated into the current asset price (Refenes, 1997). These efficient market assumptions advocate that positions in investment portfolios are held for long time periods.

The first two moments of financial time series are thus described by the *unconditional mean, unconditional variance and unconditional correlation* and defined mathematically for the time series of two asset returns, x and y, by

[1] This assumes that the return distribution is approximately normal and that the variability is an accurate measure of asset risk. Alternative risk metrics have been developed for skewed distributions, or where investors are interested solely in relative performance, or where risk is considered to be sensitive to the second lower partial moment of the return distribution (for further details see Balzer, 1994; Bernstein 1998).

$$\mu = E(y_t) \qquad \sigma^2 = E\left[(y_t - \mu)^2\right] \qquad \rho = E\left[(y_t - \mu_{y_t})(x_t - \mu_{x_t})\right] \tag{3.1}$$

where the unconditional mean μ, is the average of the asset return series, the unconditional variance σ^2, is the square of the standard deviation of the historical asset returns and the unconditional correlation ρ, is the historical correlation of the two asset returns, x and y.

In the second stage of portfolio theory, optimal portfolios are formed from the estimates of the expected return, variance and correlations of potential assets. If we assume that proportions of an investment are allocated to a number of different assets then the expected return R_P, and standard deviation σ_P, of any portfolio of assets is defined mathematically by

$$R_P = \sum_i w_i \mu_i \qquad \sigma_P = \sqrt{\sum_i w_i^2 \sigma_i^2 + \sum_i \sum_j w_i w_j \rho_{ij} \sigma_i \sigma_j} \tag{3.2}$$

where μ_i and σ are the expected mean and standard deviation of the returns of the individual securities respectively, w_i is the proportion of each security in the portfolio and ρ_{ij} is the correlation between the returns of two different assets, denoted by i and j, where $i \neq j$.

In equation (3.2) the portfolio risk is not equal to the weighted average of individual asset risks but the square root of the total portfolio variance, which is defined by two terms, the weighted variance and the weighted covariance. The formation of a portfolio of assets provides a means by which some of the portfolio risk is reduced through diversification with the selection of assets with low correlations as well as low individual risks. This is the basic premise of modern portfolio theory where the mean-variance analysis is used to construct portfolio weights that achieve the required balance of maximising return while minimising risk.

In practice, however, the portfolio efficiency achieved by conventional mean-variance optimisation rarely achieves its potential performance in future trading. Often, "in sample" optimal portfolios are highly sub-optimal in true "out-of-sample" performance. The main problem with mean variance optimisation arises from the instability of the weights of optimal portfolios to small changes in the estimated inputs.

In order to overcome this problem, techniques have been developed to reduce instability and impose a minimum level of diversification. These methods include the use of *ad hoc* portfolio constraints that limit the minimum and maximum weightings of an asset in a portfolio. Bayesian techniques have also been used which place a prior on the expectations (Satchell, 1998) and resampling statistics (Michaud, 1998) to construct the optimal portfolio on the efficient frontier. These techniques have led to more robust portfolios and have been demonstrated to enhance out-of-sample investment performance (Michaud, 1998). The sensitivity of mean-variance optimisation also indicates that relatively small improvements in the predictability of the expected mean, variance and correlation may lead to significant increases in investment performance.

Given the empirical evidence and theoretical arguments for some degree of predictability in financial markets, any departures in asset returns from the random walk model would provide a theoretical basis for a more general form for the expectations of asset returns. Under this assumption the naïve estimate, based purely on statistics of historical data, is replaced by a general estimate conditioned on the most recent past returns and/or fundamental (or market) factors that influence, to some degree, the future behaviour of the financial time series. The unconditional estimates of expected returns and risks, given in equation (3.1), are substituted by statistical time series forecasting models defined by the *conditional mean, conditional variance and conditional correlation* and, for two time series *x* and *y*, take the form:

$$\mu_C = E\left[y_t | F_{t-1}\right] \quad \sigma_C^2 = E\left[(y_t - \mu)^2 | F_{t-1}\right] \quad \rho_C = E\left[(y-\mu_y)(x-\mu_x) | F_{t-1}\right] \tag{3.3}$$

where the expectations are conditional on some vector of lagged time series variables, F_{t-1}.

A wide range of forecasting techniques may be applied to the modelling of these statistics which describe the expected distribution of future asset returns. The highly stochastic nature of asset returns and the poor understanding of asset price dynamics has led to the development of ever more complex techniques which attempt to model any non-linear or time varying relationships that may be present in financial markets. Until recently, the search for significant forecasting models has proved illusive with some empirical studies suggesting possible predictability but offering very little convincing proof. However, in recent years, with researchers combining economic understanding with modelling expertise, explicit empirical research has emerged which is believed to contain concrete evidence that forecasting models can have some degree of predictive power (see chapter 2 for more details).

In principle, forecasting models offer the potential for more accurate estimates of the future behaviour of asset returns. Increasing predictability has the advantage of providing more accurate estimates into the portfolio optimiser which, in principle, lead to an improved out-of-sample investment performance. The expected performance of the optimised portfolio is then some complex function of the predictive ability of the models for the conditional mean, variance and correlation of the individual asset returns. Often, conditional estimates of future asset behaviour lead to an optimised portfolio that requires the implementation of significant rebalancing. The obvious application of these techniques in managing a portfolio using mean-variance analysis is to rebalance the portfolio to directly improve the expected return-risk performance of the portfolio. The implication of altering the portfolio construction at discrete time steps is to create a sequence of rebalancing instructions to achieve some overall investment objective.

The upshot of this process is to create a dynamic portfolio management approach that requires large changes in allocations to exploit the predictability of forecasting models. These types of trading systems typically require the optimisation of sequences of interdependent decisions to achieve the overall objective and to properly incorporate the effects of trading costs (e.g. transaction costs, market impact, taxes) which are often significant over shorter time scales. Sequential decision tasks under uncertainty have been extensively studied in other subject domains and in the second half of this chapter we review the development of *sequential decision models* and develop links with applications in investment finance.

3.2 Review of Forecasting Methods

In this section we review statistical modelling techniques for forecasting that range from relatively simple moving average techniques to more complex methods, described by ARIMA (Box and Jenkins, 1976) and time series regression methods, cointegration and error correction models (Granger, 1983), and state space modelling techniques (Harrison and Stevens, 1976). We discuss the properties, strengths and weaknesses of these competing methods and the selection of forecasting models.

One important underlying assumption of all forecasting methods is the belief that future behaviour is similar or related to the past. If this assumption is violated then the forecasting model is considered to be unreliable and subject to model breakdown. This assumption is normally referred to as nonstationarity and has been the subject of considerable research in order to maintain the integrity of forecasting models. A comprehensive treatment of issues relating to nonstationarity can be found in Burgess (1998).

3.2.1 Moving Average/Smoothing Methods

One the most pragmatic but commonly used family of methods for forecasting time series are known as the smoothing, or moving average, methods. These have been developed to model the main types of deterministic components (i.e. seasonality, cycles and trends) that may be present in a time series.

For brevity, we focus on the *simple moving average* (SMA) model and the *simple exponential smoothing model (ESM)*. The concept is to smooth the sharp variations in the observed time series but allow more recent values to have greater influence on the forecasts than more distant observations. This is achieved by constructing a time-weighted average of past observations. In the case of the simple moving average an equally weighted window of data is used while the weights of the simple exponential smoothing decays exponentially with the number time periods from the current time step.

The simple moving average model is defined for a time series, y, at time, t, over all future time horizons (denoted by τ), as

$$\hat{y}_{t+\tau|t} = \frac{1}{h}\sum_{i=0}^{h} y_{t-i} \tag{3.4}$$

where h controls the window length of the most recent historical observations.

The simple exponential smoothing model is similar but uses a weighted average to increase the importance of the most recent data. It is normally re-expressed so the future forecasts are equal to the weighted sum of the current observation and the previous forecast and so takes the form:

$$\hat{y}_{t+\tau|t} = \alpha y_t + (1-\alpha)\hat{y}_t \tag{3.5}$$

where α is the weight decay parameter that controls the relevance of the historical observations.

Both models can also be used for estimating the future variance of the asset return by replacing the last observation with the last observed squared residual. More general smoothing models were developed by Holt (1957) and Winters (1960) to model time series components relating to trends and seasonality by using additional model parameters. Other related methods were suggested by Brown (1963) who used the *discounted least squares* method to emphasise the most recent observations through an exponentially weighted decay function. This method has been extended to consider polynomial models. A review of the development in smoothing methods is provided by Gardner (1985).

The parameters of smoothing models are either pre-specified on the basis of some *a priori* belief in regard to the underlying properties of the time series or selected according to some optimisation criteria that measures forecast accuracy on some test/validation data set. The mean squared error was originally shown to give optimal (one step ahead) forecasts by Muth (1960) when the time series is non-stationary. In general, model parameters are typically selected according to the analysis of a number of metrics of forecast accuracy, as discussed in many texts on statistical forecasting (e.g. chapter 12 of Diebold, 1998). Smoothing techniques are widely used as forecasting models in many applications areas outside of investment finance and have the advantage of being easy to understand, require few observations, and parameter selection can be automated. The disadvantages of smoothing techniques are due to the simple, univariate nature of the modelling process and the specification of the model parameters. In next section we progress to more principled modelling techniques.

Many surveys of large-scale empirical studies using forecasting methods (Makridakis and Hebron, 1979; Makridakas et al., 1982; Winkler and Makridakas, 1983) indicate that simple methods (i.e. exponential smoothing) can perform as well as more complex methods on out-of-sample data. These results are primarily due to the underlying modelling assumptions of model selection that the selected model is the true representation of the process and the time series does not undergo any structural changes. Obviously in many real applications this occurs and so results favour simple parsimonious models that do not over-fit the in-sample data.

In investment finance, moving average/ smoothing models are typically used when it is believed that the functioning of markets is slowly changing through time and more emphasis is required on the more recent observations. The conditional mean of a financial time series can either be modelled by transforming the price series into a returns series by taking logs of the differences in consecutive prices and then smoothing the returns series. Alternatively, the price series can be smoothed and log differences calculated from the transformed series. The relative efficiency of financial markets and the lack of obvious deterministic components in financial time series limits the advantage of using smoothing models to estimate the future expected return of an individual asset.

3.2.2 ARIMA and time series regression models

Autoregressive-moving average (*ARMA*) modelling is a classical methodology for estimating statistical forecasting models. It uses two processes, the autoregressive (*AR*) terms and the moving average (*MA*) which describe the lagged values of observed values and forecast errors respectively. These processes are combined to approximate any stochastic time series, where the general equation of the *ARMA(p,q)* model takes the form:

$$\hat{y}_t = \mu + \sum_{i=1}^{p} \phi y_{t-i} + \sum_{i=1}^{q} \theta \varepsilon_{t-i} \tag{3.6}$$

where ϕ and θ are the coefficients of the autoregressive and moving average processes respectively, and where p and q denote the number of lagged values of the observed time series and the error terms respectively.

The development of the *ARMA* process was originally suggested by Yule (1921) and Slutsky (1927) and has formed the basis of a family of forecasting models which has been widely used for modelling economic time series (e.g. Box and Jenkins, 1970). ARMA models have been extended to non-stationary stochastic processes by differencing the time series until it is stationary. This modelling framework was introduced by Box and Jenkins (1970) and is referred to as *ARIMA(p,d,q)*, where d denotes the order of differencing or *integration* within the time series.

In addition to ARIMA models, time series regression models have also been developed to incorporate the effects of exogenous factors that persist over time. Initially, distributed lag models were developed that incorporated lag effects of independent variables into a regression model (see Grilliches, 1967). The finite distributed lag model has the general form:

$$\hat{y}_t = \alpha + \sum_{i=0}^{q} \beta_i x_{t-i} \tag{3.7}$$

where q is the finite number of time periods in which the independent variable x_t has influence and β is the lag coefficient.

These were extended to many types of distributed lag model including unrestricted finite lag models, polynomial lag models (Almon, 1965) and geometrical lag models. The latter models are designed to control effects of multicollinearity by imposing some structure on the lagged variables and also benefit from requiring fewer model parameters. These models were a form of mixed autoregressive-regressive model that could recognise any temporal dependence between variables while capturing autocorrelation amongst errors. (For more details of distributed lag models, see Greene, Econometric Analysis, 1993)

ARMA and time series regression models were later combined by the development of *ARMAX models*, introduced by Hatanaka (1975) and Wallis (1977), that incorporate lagged exogenous variables into the autoregressive-moving average process. Further extensions included the development of multivariate *ARMA* models, with the most common known as the vector autoregressive model or *VAR* model. These models have advantages in not requiring specification of contemporaneous variables as exogenous or independent and have been used to test for causality between variables (Granger, 1969).

The parameters of *ARMA* models are typically estimated by nonlinear least squares estimation, (Box and Jenkins, 1970) or maximum likelihood (Newbold, 1974) and these two methods are discussed in most texts on time series modelling. The methodology for selecting between different *ARMA(p,d,q)* models is commonly accomplished by model identification procedures developed by Box and Jenkins (1970) using autocorrelation functions (*ACF*) and partial autocorrelation functions (*PACF*).

3.2.3 Cointegration and Error Correction Models

Cointegration is a method for describing the long run relationship between a group of variables which exhibit an equilibrium relationship with each other. It differs from correlation, the standard measure of linear association between variables, by relaxing the requirement for time series to be jointly covariance stationary and so avoid the loss of information due to differencing. Financial time series (e.g. Equity market indices) are particularly good examples of non-stationary variables that may exhibit some long run relationships between their prices.

The first step in building a cointegration model is to identify a set of time series that are individually non-stationary and integrated order one, but which drift upwards at roughly the same rate, and form a linear combination which is stationary. Intuitively, this implies that the nonstationary component is a common trend and we can use the other time series to detrend a target series rather than differencing. The time series that satisfy this requirement is said to be *cointegrated* and the coefficients are the cointegrating vector. In general, the cointegrated time series for some target variable, y_1, takes the form:

$$y_{1,t} = \beta_2 y_{2,t} + \beta_3 y_{3,t} + \beta_4 y_{4,t} + ... + \varepsilon_t \tag{3.8}$$

where y_2, y_3,...y_n, are other cointegrating variables, ε_t is the deviation term and β the cointegrating vector.

The cointegrating time series can be estimated using standard estimation procedures (e.g. linear regression) and the cointegration hypothesis tested by applying modified versions of standard stationarity tests to the residuals (e.g. Dickey Fuller, and cointegration regression Durbin Watson).

In investment finance, one application is to identify weak form inefficiencies between markets (Chelley-Steeley and Pentacost, 1994; Choudhry, 1994, Burgess, 1996). In principle, the cointegration perspective is able to detect market inefficiencies or changes in underlying relationships that may otherwise remain undetectable. This is due to cointegration analysis effectively reducing market noise caused by unpredictable events or news information that influences the price of a whole group of assets.

After the cointegrated time series has been specified, a forecasting model of the target time series can be developed by using the residuals from the cointegrating time series. This is known as an *Error Correction Model* (ECM) (Granger, 1983; Engle and Granger, 1987). The model incorporates the deviation term that acts as the error correcting process that forces the asset prices back into some long run equilibrium. The error correction model for a cointegrating time series y_t, with deviations, ε_t, defined from equation (3.8), and lagged variables of the change in the target and cointegrating series, takes the general form:

$$\Delta y_t = \mu + \sum_{i=1}^{p} \theta_i \Delta y_{t-i} + \sum_{k=1}^{n} \sum_{i=1}^{q} \phi_{i,k} \Delta y_{k,t-i} - \gamma \varepsilon_t + e_t \tag{3.9}$$

where θ and ϕ are the model coefficients, which incorporate the influence of past changes in the time series and γ, is the coefficient of the error correcting process defined from cointegration.

The advantage of this modelling approach is that long run relationship (i.e. the common drift) between the time series and the short term dynamics (i.e. the deviation of the time series from the long term trend) can both be taken into account. Error correction models can be constructed using regression analysis (e.g. Granger, 1983; Engle and Granger, 1987; Stock, 1987), or other multivariate methods such as canonical cointegration (Park, 1989), principal components analysis or where there is more than one cointegrating vector using vector autoregression (VAR) models (Johansen, 1988).

3.2.4 State Space Models

In the 1960's, state space modelling techniques were initially developed as a recursive method of controlling multi-sensor systems, such as navigation systems to track spacecraft, (Kalman, 1960). These ideas were initially applied to time series modelling by Stevens and Harrison, (1976) who derived equivalent ARIMA models using the state space representation.

The basic notion of the state space model is to represent the components of the time series by a system of equations. The states of the system, described by "state variables", are then determined from the observed time series. In economics, state variables have been developed to represent time series components such as trends, cycles and seasonality, (Pagan, 1975; Harvey and Todd, 1983; Harvey 1983) and in investment finance, to represent conditional risk factor sensitivities of equity investment management (for review, see Bentz, 1999).

The advantage of the state space representation is the ability to optimally estimate models using the *Kalman filter*, a powerful estimation algorithm (Kalman, 1960). In economics, this has seen the development of structural models (Harvey, 1989), which are time invariant models where the observable variables are decomposed into systematic effects (trends, cycles and seasonality) and stochastic components. Other applications of state space models include the development of adaptive models that allow for time varying or adaptive relationships between variables to be tracked over time (Harvey, 1993). Further details of state space models and the Kalman filter can be found in many textbooks (e.g. Harvey, 1989).

An example of a simple adaptive model is the stochastic coefficient regression model, which can be expressed in state space form, by the general system of two equations. The first is referred to as the *observation equation* and the relates the dependent variable y_t, to the independent variables x_t, through the unobservable state vector, β_t which represents the model coefficients. The *observation equation*, for all time steps t, takes the form:

$$y_t = \boldsymbol{\beta}_t \mathbf{x}_t + \varepsilon_t \tag{3.10}$$

where ε_t is the unobservable *observation noise* term with variance, σ^2.

In general, the elements of β are unobservable but their time structure is assumed to be known. The evolution of these states is assumed to be a Markov process and described by a *transition equation* that for a random walk model takes the form:

$$\boldsymbol{\beta}_t = \boldsymbol{\beta}_{t-1} + \eta_t \tag{3.11}$$

where η_t is the *system noise* term with variance, q^2. The two noise terms are considered to be uncorrelated Gaussian white noise processes.

The time varying coefficients, β, of the stochastic regression are then estimated in the presence of noise using a *Kalman Filter*. This is a recursive method that optimally estimates the coefficients at time t based upon the available observations. The assumed levels of observation and system noise are used by the filter to determine the how much the variation in the dependent variable y is attributable to the system and how much to observation noise. The filter consists of a system of equations that updates the estimates of the coefficients when new observations are available. The state space formulation of the model allows the estimates of the states to be updated based solely on the last observation and still take into account the entire history of observations. A more general definition of the equations used to update the coefficients is given by Bentz (1999) and the derivation of the Kalman Filter by Harvey (1989).

In investment finance, state space models are particularly powerful when they allow for relationships between variables to change through time. In the time varying regression model a large forecast error may be due to either an unpredictable event or a change in the underlying relationship. The purpose of the Kalman Filter is to attribute the innovation to these two sources based upon the relative size of the variances of the noise terms, σ^2 and q^2. Time varying relationships may also be modelled by modifying standard modelling techniques (e.g. regression) and estimating parameters by time weighting the historical data (e.g. time weighted least squares regression, Refenes, 1994).

This completes the review of four statistical forecasting models that may, in principle, be applied to estimating future asset price behaviour in investment finance. Next, we briefly review methods of evaluating and testing forecasting models.

3.2.5 Model Selection Criteria

Forecasting models are typically evaluated on the basis of some performance criteria, which can be grouped into the following categories:

- in-sample accuracy
- out-of-sample accuracy
- application-specific performance

In-sample accuracy measures quantify the degree to which the forecasting model correctly represents the available insample data, subject to model integrity tests. A range of statistical criteria has been developed which measure the trade-off between the goodness-of-fit and the number of degrees of freedom of the model. Model selection criteria include R^2 (coefficient of determination), adjusted R^2, Akaike Information Criteria (AIC), Bayesian information criteria (BIC), Schwarz Information criteria (SIC) and F-ratio. For a discussion of these tests see Diebold (1998). In financial forecasting, these methods can result in sub-optimal models as data is often subject to high levels of noise and relationships may be unstable over time due to presence of nonstationarities in the underlying data generating process.

Out-of-sample accuracy measures quantify the forecasting model on the basis of the forecast accuracy in some out-of-sample data set. This approach is commonly used for stopping the training process of neural network models in order to avoid "overfitting" the in-sample data. In order to use all the data for model estimation, data re-sampling methods have been developed, such as k-fold cross validation, which estimate model performance on randomly selected subsamples of *k* observations with respect to models optimised on all remaining data (for more details see, Efron and Tibshirani, 1993). In financial forecasting, other accuracy measures have been commonly used in an attempt to give some further intuition to the value of the forecasting model, such as percentage signs correct.

Both of these selection methods assume that lowest forecast error will provide the "best" model for any application. In contrast, application-specific performance measures evaluate the forecasting model in terms of the value of the predictive information for a particular decision-making application. The objective of this process is to gain some insight into the value of the predictive characteristics of a model in the application domain. In principle, there is a fundamental difference between the purpose of the forecasting model (i.e. to assist decision making) and the model optimisation/selection criteria. This principle of forecast modelling is discussed in forecasting textbooks, (e.g. Diebold, The elements of forecasting, 1997) where the effective design, use and evaluation of a forecasting model is described as ultimately requiring information of any associated decision making. However, in practice, the construction of forecasting models in relation to decision making is often achieved in an *ad hoc* manner guided by modelling expertise. In financial forecasting, the most common approach has been to apply some trading strategy, often implemented using a simple trading rule, to the predictive signals in order to measure the profitability of the trading system over some out-of-sample period. Typically, performance is measured using some risk adjusted return metric, such as the Sharpe Ratio (average excess return divided by standard deviation of excess returns). These "profitability" tests have proved to be particularly common for empirical studies in financial forecasting.

In this thesis we focus on the development of optimisation/ selection criteria for forecasting models on the basis of application-specific performance. In principle this has the advantage of linking the forecasting model to the ultimate objective of optimising the application of the predictions into the decision problem. For this approach, construction of the forecasting model requires some information from the decision-making application, which describes the value of the forecasting model. In this thesis we develop a joint optimisation methodology to optimise a trading system consisting of a forecasting model and a trading model. In this framework information is fed back from the trading model to the forecasting model in order to provide some measure of application-specific performance. This information is then used to update*meta* parameters which control the design factors of the forecasting model.

3.3 Review of Financial Forecasting models

So far we have discussed forecasting models that may be applied in investment finance. In this section we chronologically review the development of financial forecasting models.

The earliest empirical studies of financial price movements found that autocorrelation structure in asset returns was weak and simplified time series models, based on ARIMA modelling techniques, could not be identified (e.g. Fama, 1980). These initial experiments tended to support the widely held view amongst financial economist that asset prices were almost entirely stochastic, thereby implying that financial markets are effectively efficient and thus unpredictable. Research then concentrated on further empirical tests for market efficiency by searching for evidence of predictability, as discussed in chapter 2.

In the early 1980's, research in financial and macroeconomic time series found evidence that error variances of time series models are heteroscedastic and not as stable as previously assumed (Engle, 1982, 1983; Cragg, 1982). Results showed that forecast errors appear in clusters, in such a way that the variance of the forecast error depends on the size of the previous error. The autoregressive conditional heteroscedastic model (ARCH) was presented as an alternative to conventional analysis of variance (Engle, 1982). This topic has subsequently developed into a whole family of models for predicting asset price volatility (for more details see Engle et al., 1987, Bollerslev, 1986).

In the late 1980's, evidence from the machine learning community showed that artificial neural networks could predict non-linear time series with chaotic behaviour which appeared random to traditional modelling techniques (Lapedes and Farber, 1987). The similarity in the dynamics of chaotic and financial time series led some researchers to investigate the possibly that asset returns that were unpredictable from a linear modelling perspective may be partially deterministic from a non-linear modelling view. Empirical studies of asset prices did indeed show some evidence of predictability, (Dutta and Shashi, 1988; Schoenenberg, 19990, Bosarge, 1990, Refenes, 1992) although interesting, similar research with an econometric bias failed to detect any evidence (White, 1988). It now appears that the conclusions of these initial experiments were overly optimistic by suggesting that asset returns were highly deterministic from a non-linear perspective.

In the early 1990's, more sophisticated modelling techniques were developed that could relax some of the traditional modelling assumptions, in a controlled manner, and have the potential to allow non-linear and time varying relationships. These computational modelling techniques aimed to learn relationships from data without *a priori* assumptions thereby, in a sense, permitting a data driven form of modelling that allowed the "data to speak for itself". An increased understanding of financial time series led researchers to examine ways of developing these forecasting methods to compensate for the stochastic properties of financial time series. This has proved a particularly interesting area to test these new methods with rich sources of data and a poor understanding of the underlying dynamics of the system (as recognised by international conferences and journals on Computational Finance).

Methods were developed to control model complexity in neural networks, using weight elimination and variable pruning techniques, and empirical studies achieved some notable forecasting performance, with significant out-of-sample predictability and potentially profitable trading. These studies were conducted in foreign exchange markets (Weigend, 1992; Abu-Mostafa, 1990,1993, 1995), corporate bond ratings (Moody and Utans, 1992) and daily returns of the German stock index (Choey and Weigend, 1996), daily prices of European Indices (Burgess, 1996). Other studies indicated the presence of non-linear relationships between economic variables and asset returns (Zapranis, Utans and Refenes, 1995) while Timmerman and Pesaran (1995) used simulation to show that forecasting models of monthly US stock returns could have significant predictability. A recent paper by Refenes et al. (1998) uses a variety of computational and statistical techniques (e.g. neural networks, state space models, cointegration) to show that predictive methods can be applied to the empirical forecasting of financial time series. However it is worth noting that empirical studies using statistical forecasting models have a tendency to over-sample the data set and so find a statistical significant model with no predictive ability. This was first pointed out by White and referred to as data snooping (White, 1991).

A landmark study showed that it is possible to build a forecasting model to maximise the predictability of constructed portfolios of stocks and bonds with respect to a set of *ex ante* observable macroeconomic variables (Lo and MacKinlay, 1995). In these models individual assets are combined to form a synthetic asset. The statistical models are shown to have predictive power even after controlling for data snooping biases. The out-of-sample measures show the predictability is authentic and profitable even for simple trading rules. The evidence is now considerable "that levels of predictability are now statistically significant, even after controlling for data snooping biases" (Lo, 1995).

This approach taken by Lo (1998) is similar to other promising studies of predictability which seem to appear in models of unobservable or synthetic quantities. These include the prediction of risk measures (Rosenberg, 1976), cointegration residuals (Burgess, 1995), the returns of investment funds (Bentz et al., 1996), and implied volatilities (Gonzales-Miranda et al., 1995). Intuitively, the reasoning behind this breakthrough may to some extent be due to difficulty in detecting market inefficiencies in synthetic assets without the framework of a statistical model and so offers a competitive advantage over other market participants using conventional analysis methods.

The recent development of predictive models has led researchers to consider their economic significance in investment strategies. It soon became clear that ultimately, to take advantage of predictive ability, forecasts need to be incorporated into some form of dynamic trading strategy. Studies began to show evidence that financial forecasting models required optimisation in relation to the decision policy and not solely the minimisation of forecast errors.

This was first indicated by applications of financial time series prediction that suggested that MSE may not be the best error measure and experiments showed that profits from trading are not strongly correlated with either the MSE or MAE (Leitch and Tanner, 1991). This was further emphasised by Satchell and Timmerman (1995), who showed that there is not always a direct relationship between the predictive ability (measured in terms of mean squared forecast error, (MSFE)) of a forecasting model and the profitability of the associated trading strategy. In this work, they developed a theorem that showed that there is not necessarily a monotonic relationship between the size of the MSFE and the probability of correctly forecasting the sign of a asset return.

Other researchers have also recognised that optimising different performance metrics could lead to significant changes in trading performance and led to the development of a wide range of comparative model selection criteria. Numerous extensions to the MSE (or L2 norm) were proposed with researchers advocating MAE and MAPE (Doboek, 1994; Refenes, 1995) with related modelling techniques developed from the field of robust statistics (Bolland, 1998). Some researchers have tried to overcome this problem by predicting the direction movement of asset returns rather than predicting the actual values by using a directional symmetry measure (Caldwell, 1995; Refenes, 1995). This has led to the development of optimisation methods for forecasting methods using neural networks that employ non-differentiable performance functions (Refenes, 1995).

The application of financial forecasting models highlights the difficulties in trying to directly integrate forecasting and trading. This was aptly described by Caldwell, (1995) who stated that "prediction and trading are two entirely different tasks, with entirely different goals". Methods have however been developed that attempt to bridge this gap by optimising the forecasting model with respect to trading. These models are reviewed at the end of this chapter.

3.4 Review of decision modelling techniques

The science of decision-making is difficult to pin down as it covers a wide range of problems and disciplines. Research is spread across many fields including operation research, computer science, control theory and decision science as well as occurring in the underlying principles of many business related topics such as economics, finance and marketing. We therefore, do not attempt to tackle the entire literature of decision theory, instead we focus on the main foundations and the aspects that are relevant to the specific problems involved in developing modelling techniques for sequential decision making under uncertainty. This part of decision theory addresses sequential decision tasks in which both short-term and long-term consequences of decisions must be considered. In this context, emphasis is placed on modelling techniques that are particularly pertinent to sequential decision tasks in investment finance, especially in the development of dynamic trading strategies for statistical forecasting models.

This section starts with an introduction of decision theory followed by a brief historical background to decision modelling. We then describe a general framework for modelling and solving sequential decision problems using stochastic dynamic programming, and two learning methods, parameterised decision rules and reinforcement learning. Finally we review the literature for modelling techniques for trading systems.

3.4.1 Introduction to Decision Theory

Decision-making can be simply defined as the process of making choices among alternatives that offer different consequences. The objective of decision theory is then to quantify the overall reward from different possible actions and so enable the best decision to be determined. In practice, however, many decision tasks face some degree of uncertainty over the rewards received from alternative actions. Under these conditions, the goal of decision theory is to incorporate statistical knowledge of the uncertain aspects of the problem within a theoretical framework in order to optimise decision making.

Decision theory operates, in an analogous way to statistical forecasting methods, by decomposing the task, where possible, into separate components. These are modelled using a mathematical framework that is used to find the optimal solution.

In general, this process involves breaking the decision task into *states*, *actions*, *utility function*, *prior information* and *historical data*. In this approach, system states describe the possible behaviour of the observed system and *actions* describe the set of possible decisions. The utility function describes the perceived reward received from a selected action in a particular state, and prior information and historical data describe the information available about the uncertainty that may influence a reward. These basic concepts cover nearly all decision modelling problems and can be expanded to model sequential decision tasks by adding state-action transition probabilities. These define the chance of moving to a new state given an action in the particular state, and also extending the definition of the utility function to describe the sequence of rewards received from following a sequence of actions, often denoted as a decision policy.

In decision theory, the system state is defined as the condition of a system at a particular time and is used to determine all aspects of future behaviour independently of how the state was reached. For this condition to hold the current state must contain all information of the past operation of the system that is relevant to the future behaviour. This is known as the *Markov property* (memoryless) and in stochastic systems is used to determine the probability of all aspects of future behaviour of the system. The assumption of complete state information is an indispensable property and underpins the modelling of decision making in dynamic systems and has fundamental implications to the optimisation of models to control decision making.

Decision theory has been applied to a wide range of decision tasks ranging from single horizon, deterministic environments to complex, multi-step, dynamical systems where decisions must be selected in the context of an optimal decision policy. In these more complex problems, decision models and forecasting models are used when solutions cannot be determined analytically and optimal policies can only be approximated numerically.

In investment finance, the elements of decision theory play an important underlying role in the development of modern portfolio theory (Markovitz, 1952). The mean-variance framework of portfolio theory uses analysis of the expected return and risks of individual to optimise asset allocation according to an investment objective. In this framework the state describes the current portfolio position, the actions describe the possible portfolio weights, the utility function describes the perceived return risk preference of the investor, the prior information and data describe the estimated returns, risks and correlations of the individual assets. This model is then used to determine the optimum portfolio which is the "best" action for the investor. This model of investment decision making only considers optimising investments over a single horizon and takes no account of sequences of interdependent decisions to optimise investment performance.

In this chapter we introduce decision models to optimise sequences of decisions to optimise some overall objective. These are more appropriate for dynamic trading strategies that typically take place over many time periods where trading effects lead to interdependent decision making. In the next section we discuss the historical background to decision modelling with particular emphasis on modelling sequential decision problems.

3.4.2 Historical Background to Decision Modelling

The historical background to decision theory has been traced back as early as the 19^{th} century with some of the basic concepts dating back to problems investigated in the early 17^{th} century (Cayley, 1875). However, a significant landmark in solving discrete-time sequential decision processes was the Principle of Optimality presented by Bellman (1954). This principle applies to general Markovian (memoryless), multi-period decision problems, where states and actions can be defined at each stage. The principle states that "an optimal policy has the property that, whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision" Bellman (1951). Thus, the optimal decision policy consists of a sequence of optimal decisions from an initial state.

Following this breakthrough, optimisation methods for determining optimal policies developed, known as Dynamic Programming (DP), which have been used successfully to solve many sequential decision tasks. DP methods were shown to be efficient at finding optimal policies in sequential decision problems with a complete understanding of the state-action dynamics. This knowledge included all state transition and reward probabilities of the underlying decision system. The solution then involved searching through the set of all state and action sequences to determine the optimal policy, often implemented for discrete state-action spaces using a decision tree representation. These key ideas were applied to both finite and infinite horizon sequential decision tasks and shown in many cases to accurately approximate the optimal decision policy. The theoretical foundations of decision theory led to further work into formal proofs of convergence (Blackwell, 1965), other modified iterative algorithms and continuous-time models to solve a wider variety of stochastic sequential decision tasks. For further reading on the development of dynamic programming, see Bellman (1957), Jacobs (1969), Nemhauser (1966), White (1969) and for stochastic dynamic programming (Puterman, 1995).

During the 1950's and 60's, considerable research, independent of decision theory, was undertaken in artificial intelligence, from a computational and psychological perspective focused at understanding learning approaches to decision making. At the end of this time, a synthetic learning approach known as "reinforcement learning" began to appear which could learn interactively from rewards or punishments received from selected actions (Mendel, 1970). The learning process proceeded by trial and error where actions followed by good or bad outcomes are strengthened or weakened respectively. In addition, other evolutionary methods (e.g. Genetic algorithms) showed promise in searching for optimal decisions (Holland, 1965). These computational learning approaches contrasted to other research in machine learning that considered a supervised style of learning that suited pattern recognition tasks but involve no direct interaction with an environment (for more details see Bishop,1995).

In the 1970's and 1980's researchers began integrating ideas from reinforcement learning with decision theory to provide a theoretical framework for understanding stochastic decision systems. Contributions to integration were primarily made by Werbos (1977, 1987), Sutton (1984), and Watkins (1988). These algorithms were explicitly related to dynamic programming and known by terms such as *heuristic dynamic programming* (Werbos, 1977), *incremental dynamic programming* (Watkins, 1988) and also *adaptive heuristic critic* (Sutton, 1984). These algorithms are distinguished from traditional DP methods by relaxing the objective of optimality in exchange for heuristic search methods that approximate the optimal solution and achieve relative computational efficiency for problems with large numbers of states which had proved impractical for standard methods.

Arguably, the most important breakthrough was the development of the Q-learning algorithm (Watkins, 1989), which directly linked temporal difference learning and dynamic programming (DP). Temporal difference (TD) learning (Sutton, 1988) is an incremental learning method that solves the problem of predicting the expected reward from a given state. Combining DP with TD enabled the on-line estimation of the value function without depending on reward expectations and transition probabilities. In the 1990's, research has focused on extending reinforcement learning algorithms to solve issues related to *generalisation*, *exploration*, rate of *convergence* and *on-line updating methods*. Before this point much of the theory of reinforcement learning had concentrated on discrete Markovian environments which cannot be easily used to model many practical tasks. Next, these issues are addressed in more detail.

In general, continuous state-action spaces require discretisation of states and actions. A fine encoding of a continuous space may be impractical due to the curse of dimensionality. However, too coarse an encoding may lead to significant inaccuracies in value function approximation and result in sub-optimal policies. Subsequently, universal function approximation methods were developed to *generalise* discrete action-state formulations to continuous state-action spaces by parameterisation of the value function. A number of techniques have been developed including neural networks (Rummery et al., 1994) and radial basis functions (Van Roy and Tsikislis, 1996). The topic of parameterised reinforcement learning is sometimes referred to as neuro-dynamic programming (Tsikislis, 1998). This is still an open area of research with much work building on the foundations of neural network techniques (Lin, 1992; Sutton, 1988; Anderson, 1993; Thrun, 1994; Tesauro, 1992; Boyan, 1992; Rummery, 1994). At present, methods for value function approximation only guarantee convergence for discrete states or when using linear models to approximate a continuous state space (Rummery, 1994).

The optimisation of sequential decision tasks involves the process of investigating the value of actions in each possible state. This process involves exploring the possible actions and gathering information about the expected rewards. For DP methods this process is accomplished off-line by efficiently searching through the space of possible states and actions and then determining the optimal policy. For reinforcement learning this process is more complex as the search for the optimal policy cannot be accomplished off-line. *Exploration* must be balanced with exploiting the current knowledge to choose the best actions. If the space of actions is not explored then the system may not find the optimal policy, however while exploration is taking place the system will be performing sub-optimality. A variety of methods have been developed to select non-greedy actions for discrete state-space systems (Thrun, 1992) and also for continuous function approximators (Rummery et al., 1996).

For problems with large state-spaces that are deemed computationally impractical, real-time dynamic programming was developed (Barto et al., 1993) to focus on learning only regions that are visited during normal operation. Other methods for speeding up convergence involve alternative Q-learning algorithms such as modified Q-learning (SARSA) (Rummery, 1994; Sutton, 1994), summation Q-learning (Rummery, 1994), Q(λ) (Peng and Williams, 1994). These methods incorporate a stochastic policy procedure (i.e an "ε-greedy" algorithm) to reach convergence of the value function. The idea is to separate exploration from exploitation of the reward information by allowing a deviation in the current optimal policy while switching off the update procedure. The selected action is then stochastic and can only be used to estimate the value function following the policy decision. This enables exploration while maintaining integrity of the current estimated policy. Methods that use these update rules to control exploration have shown to be more robust and increase the convergence rate for the Race Track problem (Barto, Bradtke and Singh, 1993).

To achieve value function *convergence*, reinforcement learning algorithms have been shown to guarantee convergence for finite discrete states given a finite probability of visiting all states repeatedly and provided an optimal policy exists (Bertsekas 1987; Bertsekas and Tsiksiklis, 1989, Watkins and Dayan, 1992; Jaakola, Jordan and Singh, 1993). However these results do not give any indication of the *rate of convergence* which is important to compare alternative methods for a given sequential decision problem.

The most well known application of Q-learning is to the game of backgammon where a computer model was able to successfully compete in international competitions (Tesauro, 1992). Other successful applications include the optimisation of an elevator (Crites and Barto, 1996) and job shop scheduling (Zhang, 1995). Examples of reinforcement learning methods applied to investment finance problems include temporal difference learning to approximate a policy function to optimise investment trading given selected explanatory variables (Moody et al., 1998), and temporal difference learning to price high-dimensional exotic derivatives (Van Roy, 1998).

In the next section we introduce a general model for solving sequential decision tasks that encompasses a wide range of applications and demonstrates the concepts involved in developing decision models.

3.4.3 Sequential Decision Model

Consider a general sequential decision task that involves making decisions in a series of discrete time periods[2]. In this class of problems the model must take into account outcomes of the current decision and future decision making opportunities that arise as a consequence of the current decision. This is symbolically represented for two points in time by figure 3.6.

[2] This restricts our attention to innovations that correspond to discrete points in real time separated by some time interval.

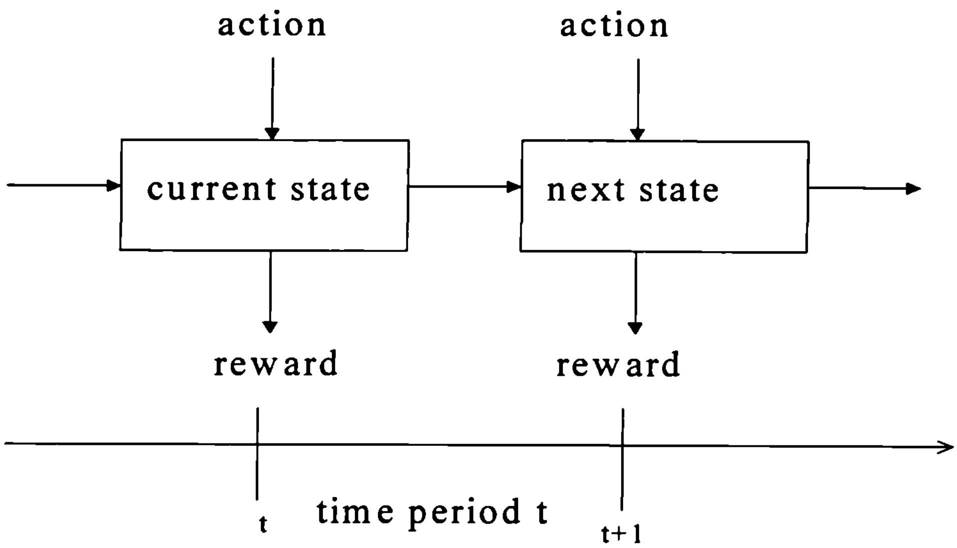


Figure 3.6 graphically depicts a representation of a sequential decision problem.

In this representation, at a point in time, say t, the system is observed and the current state determined. Based on the information contained in the state the decision maker selects an action which produces a reward. The value of the reward is dependent on the current state, the selected action and possibly some random disturbance. The system then makes a transition, during time period, t, to the next state, observed at time point, $t+1$, determined by the current state, the selected action and possibly some random disturbance. Upon observation of the next state the decision maker selects another action, receives another reward and the system makes a transition to another state. This cycle of events, namely, state-observation, action, reward, state transition, repeats for each time period.

The purpose of the sequential decision model is to assess the value of alternative actions and so enable the selection of actions that maximise some pre-specified function of the rewards over time, defined as the *total reward*[3]. The process of evaluating alternative actions is complicated by the interdependence between subsequent actions and states so the consequence of an action cannot be considered in isolation from future or past actions. This connection prohibits action selection based on simply maximisation of the immediate reward (i.e. "myopic" optimisation) but instead requires optimal actions to anticipate the opportunities for rewards in future system states. This is because some actions give high immediate rewards but may result in system states whose only allowable actions produce low rewards. Hence, overall, the sequence of actions might produce a total reward that may be lower than otherwise possible. Conversely, an action that produces short term low reward may give an opportunity for longer term high rewards. Thus, an optimal decision cannot be made myopically but must account for both short and long

[3] The total reward function is usually but not necessarily implemented as the sum of individual rewards.

term consequences of actions. The optimisation of the total reward is therefore dependent on the number of time periods (denoted as the *horizon* of the decision task), the sequence of actions and states that occur over the time periods, and the stochastic factors that influence the rewards and state transitions.

This description of the decision system allows the development of methods that select sequences of actions, known as a decision *policy*. This may be specified, for example, as some form of decision rule, which associates an action with each possible system state. Thus the total reward for a state depends on the consequences from a sequence of actions specified by the decision policy over the horizon of the decision task. If the decision task is subject to random disturbances then the value of a policy is defined mathematically by the expected total reward. The objective of the decision model is then to find a policy given an initial system state that maximises the expected total reward, predefined by some performance criterion. The best policy is then known as the optimal policy.

Although there are many special cases of applications that do not fit precisely into this general sequential model many practical applications can be constructed to meet these parameters. Examples of practical application include route planning, resource allocation, gambling, and many robotic and control systems problems as well as investment management. Next we discuss how decision models can be used to find optimal solutions to sequential decision tasks.

3.4.5 Solving Sequential Decision Tasks

The formulation of sequential decision models for practical decision tasks has led to the development of an extensive literature devoted to finding optimal decision policies. These methods can be categorised on the basis of whether or not a complete model of the decision system is assumed.

The numerical method most commonly used to solve stochastic decision tasks when a complete model of the decision system is available (including all state–transition probabilities and reward probabilities) is known as stochastic dynamic programming. The algorithm performs an efficient search through all possible state sequences generated by all possible policies to find the optimal policy. DP methods can be computationally expensive for long sequences of actions with large state spaces and convergence to the optimal policy may be impractical (due to the "Curse of Dimensionality", Bellman, 1957). In investment finance, stochastic dynamic programming has been applied to pricing of American style options for stocks based on Black Scholes option price formulation (Black, Scholes, 1979).

In the option pricing application, system state transition probabilities describe the dynamics of the stock price which is assumed to follow a random walk process. American options have the possibility of exercising at any point and so the DP algorithm searches for the optimal policy that maximises the value of the option. However, if the *a priori* assumptions do not constitute an accurate model the DP method will mispecify the optimal policy and may lead to incorrect valuations of the American options.

If a complete model of the decision environment is not known *a priori* then there are two potential options for solving the decision task:

- either construct a model from experience (system identification) or
- use a method that does not require a model (real-time learning).

The first method requires the construction of a representation of state-transitions and reward probabilities for different actions. These can be calculated from experience by performing all possible action-state combinations of sufficient frequency to accurately estimate the probabilities. This process can require considerable system interaction for complex tasks with large state-action spaces. Once the state transitions and reward probabilities have been approximated a complete model of the system is known and so DP methods can be applied to determine the optimal policy.

The second method requires the sequential decision model to be solved without a complete model of the decision system. Reinforcement learning methods describe a collection of learning or adaptive methods for finding optimal policies in the absence of a complete model of the decision task. The optimal policy is constructed by *directly* learning about the system while interacting with it. The method adjusts the policy as a result of observed consequences and not indirectly from either *a priori* assumptions or a model of the state-transition probabilities and reward probabilities. In this approach, only actions that are performed can be evaluated, so the choice of actions needs to be fully explored and consequences observed while adjusting the policy to converge to the optimal policy. The consequence of an action is reinforced during the learning process, and so this method was aptly named *Reinforcement Learning* (Mandel and M^c^Laren, 1970). These two methods are not mutually exclusive and have been combined to optimise decision making (for more details and a discussion, see Watkins, 1989).

3.4.6. Mathematical Framework

To construct a mathematical framework for a discrete time sequential decision model we assume that the system occupies a state, s_t, at discrete points in time, from the set of possible states, S_t. The system observed in state, $s_t \in S_t$, may then have an action, a_t, from the set of allowable actions in this state, $A_t\ (s_t)$. The complete set of possible actions, A, is then the union of possible actions in the possible states, denoted by, $A = \cup_{s \in S} A_t\ (s_t)$. The sets for actions and states may be finite or a continuum represented by arbitrary countable infinite sets.

Many applications of the sequential decision model can be formulated as an infinite horizon problem by using a discounting factor. In this form the explicit time dependency of state and action sets is unnecessary and so notation can be simplified without impacting the mathematical framework. This approach establishes a *stationary policy* so a decision is associated with a state for any point in time. For simplicity, we use this assumption to refer to the observed system state, at any time t, as $s \in S$, as having an action a from a set of allowable actions, $A(s)$ without any loss of generality. (In later sections where this assumption does not hold the explicit time dependency is included.)

As a result of performing an action a, in state s, at time t, the decision system makes a transition from state, s to a new state, say s', observed at time, $t+1$, with some probability $P_{s's}(a)$ with an associated reward r, The reward received, for performing an action, $a \in A(s)$ in some observed state, $s \in S$ at time t, can be defined by some function $r_t(s,s')$. If the reward is negative then it is regarded as a cost. The theory of Markov Decision Process can simply be extended to the case of delayed rewards (Puterman, 1993).

The objective of constructing the decision model is to find the policy for selecting actions that is optimal in terms of maximising some predefined *performance criteria*. The removal of time dependency on action and states, described previously, allows the specification of a policy, denoted by π, as a mapping that assigns an action to each possible state. The stationary policy then selects an action given the state, at time t, defined as

$$a_t = \pi(s_t) \tag{3.16}$$

where π is some function of the state variables.

Given the definition of a policy, the total discounted future rewards received from a policy over an infinite number of time periods takes the form:

$$r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots + \gamma^n r_{n+1} + \ldots \tag{3.17}$$

where γ, is the discount factor subject to the constraint, $0<\gamma<1$. The value of the discount factor influences the longer term consequences of actions and is application dependent. If $\gamma = 0$ then only the immediate consequences of an action are considered and the optimal decision is made myopically.

Given the reward sequence of a policy, the expected value of a decision policy, π, in an initial state, s, takes the form:

$$V^{\pi}(s) = E^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s\right] \tag{3.18}$$

where V^{π} is the *value function* that assigns an expected value to each system state, s.

For a stochastic dynamical system, the value function for the policy, π, in an initial state, s, as given in equation 3.18, can be rewritten using the transition probability as

$$V^{\pi}(s) = \sum_{s' \in S} P(s \mid s', a)\left[r(s,s') + \gamma V^{\pi}(s')\right] \tag{3.19}$$

assuming a discrete state space. For a continuous state space, the summation is simply replaced with an integral over all possible states.

In equation (3.19) the value function associates the value of all possible states with a specific decision policy. This is achieved by combining the expected immediate reward and the discounted expected future value of all possible future states from following the decision policy. The value function is therefore not dependent on how the system arrived at the current state, only on following the policy thereafter. The value function can be thought of as a *prediction* of the total reward for a policy.

The development of a value function forms the key concept in analytic methods to solve the Hamilton-Jacobi-Bellman equation for continuous time stochastic control problems and also to organise and structure the search for the optimal policy using numerical methods such as dynamic programming and reinforcement learning.

Now that the value function has been defined for all decision policies, we can define the optimal policy and the optimal value function for a state. Suppose there are two policies, π and π', with associated value functions $V^{\pi'}(s)$ and $V^{\pi}(s)$. If the policy π' is an improvement over π then we would expect

$$V^{\pi'}(s) \geq V^{\pi}(s) \qquad \forall s \in S \tag{3.20}$$

where the strict inequality holds for at least one state.

A policy is then considered the *optimal policy*, denoted by π^*, if there is no other policy that improves the expected value, as denoted by equation (3.20). The value function of the optimal policy is defined as the *optimal value function*, and normally denoted by V^*.

By defining the value function for a policy, all the mathematical infrastructure is defined to develop methods to compute the value function and approximate the optimal policy. From the definition in equation (3.19) it is not immediately obvious how the value function for a policy in a state can be computed even if all the transition probabilities and expected rewards are known, without even considering how to calculate the optimal policy or optimal value function. In the next section we describe a collection of methods for identifying optimal policy, namely, stochastic dynamic programming, parameterised decision rules and reinforcement learning methods.

3.4.7 Dynamic Programming

The formulation of dynamic programming is traced back to Bellman's Optimality Equation (Bellman, 1957) which states the necessary and sufficient condition for a value function to be optimal in each state. For a discrete state space, the value function takes the form:

$$V^*(s) = \max_{a \in A} E\left[r_a(s, s') + \gamma V^*(s')\right] \tag{3.21}$$

From equation (3.21) the action that maximises the value function defines the optimal action when the system is in state s, under the assumption that the optimal policy is used for all future actions. Assuming a complete model for the system dynamics, so that all the possible transition probabilities for all action-states and reward probabilities are known, it is straight forward to select the optimal action if the optimal value function in the next state, $V^*(s')$ is known.

The goal of dynamic programming is then to compute the optimal value function for each state and so determine the optimal actions that constitute the optimal policy. The discount factor, in equation (3.21), allows for an infinite time horizon but the procedure for practical reasons requires this series to be truncated to some sufficiently long time horizon in order to still accurately approximate the value of the state.

Conventionally, the method proceeds by forming a backward recursion which results in an optimal decision associated with each state at each time period. The general rule for determining the value function takes the form:

$$V_{n-1}^{\pi}(s) = \sum_{s' \in S} P(s' \mid s, \pi(s)) \left[r(s,s') + \gamma V_n(s') \right] \quad \forall s \in S \tag{3.22}$$

for all system states s.

Equation (3.22) can be explained by starting from some terminal (or truncated) state and applying the policy π in state s, to perform action $a=\pi(s)$ with the known associated expected reward. Applying this analysis to all possible states $s \in S$ completes the value function for the final time period, denoted as $V_n^{\pi}(s)$ for all states, s. The algorithm proceeds by iteratively computing the value function for the policy at the previous time periods given the expected rewards and probability transitions for all actions and states, until the current state is reached.

This approach can easily be modified to approximate the optimal value function $V^*(s)$, in a similar way to equation (3.22) by calculating the maximum expected total reward at each time step. The rule for computing the optimal value function is known as *value iteration* and for stochastic programming takes the form:

$$V_{n-1}^{*}(s) = \max_{a \in A} \left[\sum_{s' \in S} P(s' \mid s, a) r(s,s') + \gamma V_n^{*}(s') \right] \quad \forall s \in S \tag{3.23}$$

for all system states s.

Using the optimal value function, defined in equation (3.23), the optimal policy π^*, can be determined as the actions that maximise the value function for each state. If V^* is computed with sufficient accuracy the process followed by equation (3.23) should produce the policy that approximated the true optimal policy which maximises the expected total reward defined by the decision performance criterion.

An alternative method to value iteration is *policy iteration* (Howard, 1960; Ross, 1983) which generates a sequence of policies, of which each is an improvement over the previous policy, until the optimal policy is reached. This method employs the criteria for policy improvement, described in equation (3.20), to converge to the optimal policy. The method begins by selecting an arbitrary initial policy π^o, and computing the value function V^{π^o} for all states using the general rule given in equation (3.22). In the next time period, the policy is modified by selecting the action that maximises the total reward and using the current optimal policy, π^o thereafter. If the revised policy is an improvement over the previous optimal policy then the policy is replaced and the new value function computed. These two phases of computing the value function for a fixed policy and then iterating the policy are repeated until the policy cannot be improved and so the optimal policy is assumed to be reached. This method is often more computationally expensive, and so less efficient, than value iteration.

Both these methods are used for solving sequential decision tasks that can be formulated as a Markov decision process with complete information and a discrete state-action space. Although these methods compare favourably to exhaustive search procedures, computational work still grows exponentially with the number of time periods, states and actions so that large complex tasks remain intractable. The advantage of dynamic programming methods is that the solution is guaranteed to converge to the optimal value function, assuming an optimal policy exists.

One of the deficiencies of dynamic programming methods is the requirement of complete knowledge of the system, including all transition probabilities, which are not available in most practical sequential decision problems. To solve this problem the policy iteration algorithm was first adapted to an environment with no transition and reward probabilities by the development of *Actor-Critic* methods (Barto, Scott and Anderson, 1983, Sutton, 1984). These are similar to the two stage process involved in policy iteration, with the *Actor* selecting actions from the stochastic policy and the *Critic* updating the policy with respect to the current value function after the action has been selected. In these methods the value function is updated using an incremental update equation that takes the form:

$$V(s) \leftarrow V(s) + \rho\left[r(s,a) + \gamma V(s') - V(s)\right] \tag{3.24}$$

where ρ is a learning rate parameter.

Next we describe more advanced methods that learn the optimal policy without requiring a complete model of the system dynamics.

3.4.8 Direct Policy Optimisation using Parameterised Decision Rules

As stated before, a stationary decision policy is a mapping that assigns an action onto each system state. This can be simply implemented as a decision rule that fixes the relationship between actions and states. If the decision rule is a member of a specific class of rules, then each rule may be specified by selecting values from a set of parameters, denoted as *parameterised decision rules*. This effectively models the functional relationship, given some encoding of the input information (i.e. the system state) and the output value (i.e. the action). The policy can be represented for each possible action, a and state, s as

$$a_t = \pi(s_t;\theta) \tag{3.25}$$

where θ are the decision rule parameters.

This approach does not require the computation of a value function and is similar to methods for parameter estimation in other fields (e.g. pattern recognition, neural networks) where parameters are adjusted to improve performance for a sample of training data. However this approach differs from classification or prediction methods (sometimes referred to as supervised methods) by not having an observed target (or correct) value of the dependent variable.

One difficulty in implementing parameterised decision rules is the selection of parameters that specify an appropriate class of decision rules that encompass the optimal decision policy. Other potential problems arise from the use of *a priori* knowledge, choice of performance measure and parameter estimation procedures to compare the performance of different decision rules specified by different parameter values. The appropriate use of *a priori* knowledge can enhance the development of parameterised decision rules by guiding the selection of the class of decision rules and the representation of the decision rule parameters.

Given a parameterised decision rule, as described in equation (3.25), the total reward for a policy is the consequence of the choice of decision rule parameters. The optimal decision rule parameters, denoted by θ^*, are then the parameters that maximise the expected reward function and can be expressed as

$$\theta^* = \arg\max_{\theta \in \Theta} E[R(\theta)] \tag{3.26}$$

where R represents the total reward and Θ is the space of possible parameter values.

The adjustment of the decision rule parameters can be viewed as a type of *learning* as it does not require a model of the decision task. The process operates by applying the decision rule to each system state, before analysing the value of the performance criterion to update the decision rule parameter values. Optimisation techniques for maximising a function can then be applied to update the parameter values to find the optimal decision rule, as described in equation (3.26).

This is a standard optimisation problem for maximising an objective function of several variables, possibly subject to restrictions on the values of the variables defined by a set of constraint functions. Routines for solving these problems are widely available for nonlinear, quadratic objective functions and nonlinear constraints. If the reader is interested in more details see, NAG library Chapter E04, (1997) or for more detailed texts, see Gill et al. (1987) or Fletcher (1974).

As an alternative, the explicit optimisation of the decision rule parameters may be achieved by applying a simple incremental learning method implemented using gradient ascent with learning rate ρ that takes the form:

$$\Delta\theta = \rho \frac{dR(\theta)}{d\theta} \tag{3.27}$$

This method requires the reward function to be a smooth continuous function and the initial decision rule parameters to bound the global maximum of the reward function.

3.4.9 Reinforcement learning

Reinforcement learning combines computational learning with the main concepts behind dynamic programming to learn actions to solve a task by trial and error. In this subsection we focus on *Q-learning* (Watkins, 1988) which is the most widely used form of reinforcement learning. Q-learning uses a different form of value function, known as a *Q-function*, to define value with respect to both the state and action instead of just the state. The *Q*-function is similar to the value function, defined in equation (3.19), and takes the form:

$$Q^{\pi}(s,a) = \sum_{s' \in S} P(s' \mid s,a)\left[r(s,s') + \gamma V^{\pi}(s')\right] \tag{3.28}$$

assuming a discrete state-action space.

The Q-function, in equation 3.28, for a policy π in state s, is directly related to the value function, defined in equation (3.19), by

$$V^{\pi}(s) = \max_{a \in A} Q^{\pi}(s,a) \tag{3.29}$$

When the state transition probabilities or reward probabilities are unknown the Q-function is incrementally updated is a similar manner to equation (3.23) by substituting equation (3.29) which gives

$$Q(s,a) \leftarrow Q(s,a) + \rho\left[r(s,a) + \gamma \max_{a \in A} Q(s,a) - Q(s,a)\right] \tag{3.30}$$

This is known as the *one step* Q-learning algorithm. After learning the Q-function, the optimal policy is simply determined by the sequence of actions that maximises the Q-function at each state.

The one step Q-learning algorithm, equation (3.30), was first combined with *TD* learning by Watkins (1988). The generalised *TD* learning approach enables all previous information to be used to update the Q-function and so potentially improve convergence. The n-step backup form of this on-line Q-learning is then defined as

$$Q(s,a) \leftarrow Q(s,a) + \rho \sum_{k=t}^{\infty} \lambda^{k-t}\left[r_k(s,a) + \gamma \max_{a \in A} Q_{k+1}(s,a) - Q_k(s,a)\right] \tag{3.31}$$

where λ is the parameter that controls the relevance of past differences.

3.4.10 Parameterised Reinforcement Learning

Standard reinforcement learning techniques store the value function using a look up table. However, function approximation techniques can be incorporated into reinforcement learning to approximate the value function. This is particularly important for considering continuous or high dimensional state problems. The aim is to use the information generated from the rewards received from certain action-states to generalise across all possible states. In this case the model of the value function is a function approximator which is updated or reinforced after each observation.

Reinforcement learning techniques for modelling the value function are similar to computational learning techniques applied to supervised learning problems, such as forecasting or pattern recognition. In most cases, the function approximator learns from a fixed data set by repeatedly updating the model parameters to estimate the appropriate mapping from the input variables to the output (or target variable). One of the disadvantages of these modelling techniques is the possibility of over-fitting, where the model learns spurious relationships in the training data that are not present in new data. The ability to control fitting is known as generalisation and is particularly important in highly stochastic environments where controlling noise effects is an important factor in model performance.

The choice of a suitable parameterisation requires some knowledge of the nature of the value function that is to be approximated. Factors that typically effect the choice of parameterisation include: the ability to learn on-line from individual training examples as well as from batches of data, good scaling with the dimensionality of the input space, ability to provide a mapping between continuous inputs and outputs and generalisation of data to approximate the value function across all states in a continuous state space. A general multi-purpose modelling technique that fulfils these requirements and has proved particularly popular is neural networks. However, other techniques have also been used for parameterisation, such as linear functions and Radial Basis Functions (RBF).

3.4.11 Neural Networks

Neural networks have been extensively studied over the last ten years so in this subsection we only provide a concise review focusing on how they may be applied to approximate the value function for reinforcement learning.

A neural network model can be considered to consists of three layers (input, hidden and output) of elementary processing units (sometimes called nodes) that are connected together by weights which act as the model parameters. Each unit performs a weighted summation of the input signals to the unit and applies some basis function. An input signal is simply the weight multiplied by the output from the connecting unit in the preceding layer.

In the context of function approximation, the functional form of a fully connected neural network with h hidden units and p input variables is typically given by

$$f(\mathbf{x};\mathbf{w}) = \sum_{j=1}^{h} w_j^{(2)} \phi\left(\sum_{i=1}^{p} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right) + w_0^{(2)} \tag{3.32}$$

where $\mathbf{x}$ is the vector of input variables, $\mathbf{w}$ is the weight vector and ϕ is the basis function.

To model non-linear relationships, basis functions are normally restricted to monotonically increasing, differentiable functions, such as *tanh* or sigmoid functions. The neural network is essentially a non-parametric regression model which consists of linear combinations of basis functions. The functional form of neural networks is capable of very general function approximation and, with sufficient hidden units, has the flexibility of a universal function approximator (Hornik et al. 1989). However, increasing the number of hidden units increases the number of weights, which in turn raises the potential number of degrees of freedom of the model (Bolland, 1998). This has the effect of increasing the chance of overfitting a given empirical data set. The weights of the neural network are modified to approximate a function, for a given empirical data set, by a particular training algorithm. Many algorithms have been proposed for training neural networks and they can be grouped into either batch (or off-line) methods or on-line methods.

On-line learning methods update weights after each observation and are more suited to dynamic systems operating in real-time with data arriving almost continuously and then discarded. This method often requires large quantities of data before the weights converge, as the data is not recycled. In contrast, off-line learning methods optimise network weights to reduce an error function for an entire training sample. This process involves only updating the weight after all training samples have been presented to the network, before repeating the process until the weights have converged. This method is more suited to learning value functions in controlled, possibly simulated environments with fixed size data sets.

In the following work we consider the error back-propagation algorithm (Rumlehart *et al.*, 1986) which is based on a gradient descent rule. Other methods include quasi-Newton, conjugate gradient methods, etc. (for reviews see Press *et al.*, 1986). Back-propagation is widely used, as it is the least computationally demanding, although it can exhibit slow convergence and sensitivity to local minima, and in the context of value function approximation, can be applied to both off-line and on-line estimation. The algorithm proceeds by computing the direction of the derivative of the error and then updating the weights in the direction of the error.

In the case of reinforcement learning, there is no direct cost function to define the error, although the target value can be approximated by the immediate reward and value of optimal action in the next time period, on the basis of TD learning. The error is then half the squared difference between the target and estimated value, with the difference, *d*, defined as

$$d = r(s_t, a) + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a) \tag{3.33}$$

where r is the immediate reward from action a in state s.

The learning rule to update a weight is defined as

$$w_{ij} \leftarrow w_{ij} + \alpha \frac{\partial E}{\partial w_{ij}} \tag{3.34}$$

where α is the learning rate parameter and $\partial E / \partial w_{ij}$ is the partial derivative of the error with respect to the weight.

For on-line learning the error derivative is calculated from current observation while for off-line learning the error derivative is the average value calculated from all training samples.

Convergence of the back-propagation algorithm is often accelerated by the use of a momentum term, in which the weight change is a weighted sum of the most recent gradients and often gives a better indication of how to change the weights. The weight update rule is changed to

$$m_{ij} \leftarrow \eta m_{ij} + \frac{\partial E}{\partial w_{ij}} \qquad (3.35)$$

$$w_{ij} \leftarrow w_{ij} + \alpha m_{ij}$$

where η is the momentum parameter, $0 < \eta < 1$, which controls the averaging of the weight updates.

In reinforcement learning, the momentum terms can be replaced by an eligibility trace which acts in a similar way but keeps track of previous output gradients. For connectionist reinforcement learning, the weight update rule is changed to

$$e_{ij} \leftarrow \lambda \gamma e_{ij} + \frac{\partial Q}{\partial w_{ij}} \qquad (3.36)$$

$$w_{ij} \leftarrow w_{ij} + \alpha d e_{ij}$$

where λ is the parameter, $0<\eta<1$, which controls the averaging of the weight updates, e is the eligibility trace and d is the difference which is defined in equation (3.33).

3.4.12 Comparison of Modelling Techniques

We have described three numerical modelling techniques for approximating the optimal policy for sequential decision problems, namely, dynamic programming, parameterised decision rules and reinforcement learning. Each has been successfully applied to different practical sequential decision problems.

The three methods are closely related to each other and use the same mathematical framework of the discrete time sequential decision model to describe the interaction of the agent (or decision maker) with the system in terms of states, actions and rewards. The goal of all three methods is to approximate the optimal decision policy through a directed search of all possible policies. The underlying mathematical framework and use of a priori knowledge distinguishes them from other optimisation methods, such as evolutionary methods (e.g. Genetic algorithms) that search in policy space, guided only by scalar evaluations of entire policies.

Dynamic programming requires the strongest underlying assumptions with a complete model of the dynamics of the decision system including state transition probabilities and reward probabilities for all possible actions and states. Parameterised decision rules require some *a priori* knowledge of the set of possible policies in order to appropriately parameterise the decision policy. Reinforcement learning requires no underlying assumptions of the state dynamics (i.e. transition probabilities, rewards) and has the ability to learn the optimal policy solely through learning from interaction with a dynamic environment.

In general, dynamic programming and reinforcement learning methods use value functions to provide a key computational step in the search for optimal decision policies. Value functions enable reinforcement learning algorithms to take advantage of the experience of individual actions and so are generally thought to be inherently more efficient than heuristic optimisation methods. Reinforcement learning has the advantage of being able to optimise the decision policy online to allow for time varying model dynamics that cannot be achieved using dynamic programming or parameterised decision rules. Parameterised decision rules have the advantage of defining a policy over continuous state-action spaces efficiently through model parameters. Reinforcement learning has been extended to continuous state-action spaces using function approximation methods but may suffer from poor rates of convergence, whereas dynamic programming is restricted to a discrete state space. Dynamic programming has the advantage of guaranteeing that it will find the optimal policy if one exists, unlike reinforcement learning which has no guarantee of convergence in a finite number of updates or parameterised decision rules that may get stuck in a local maximum.

	Reinforcement Learning	**Parameterised Decision Rules**	**Dynamic Programming**
Modelling assumptions	None	Functional form of policy	Complete system dynamics
Continuous state space	Requires flexible parameterisation	Yes (also parameterised)	No
Optimal solution	Not guaranteed	Not guaranteed	Guaranteed, assuming accurate modelling assumptions
Adaptive	Yes	No	No

Table 3.1 comparison of the three sequential decision-modelling techniques

So far, we have considered the conventional approach to the development of forecasting models and sequential decision models which views these two modelling tasks independently. In subsequent chapters we will investigate the ways of combining forecasting models with decision models to improve decision performance. Under the described framework for the sequential decision model there is no explicit term for forecast information. In the later chapters of this thesis we discuss this issue and consider forecasts to be a special type of state information. This is justified by noting the definition of a system state, as described in the introduction to decision theory, that the state describes the condition of a system at a particular time and determines all aspects of future behaviour independently of how the state was reached. For trading systems, the forecast gives a description of the asset price dynamics that impacts the investment return. Ideally the forecast is precisely the compiled view of all the past information which is relevant to future price dynamics.

However, we consider the situation where states and state transition probabilities for forecasts are unknown (i.e. the probability and value of future predictions beyond the next time step). This means that there is not a complete model of the dynamics of the system and so dynamic programming methods cannot be directly implemented with forecasting models determining part of the system-state. This restriction is not necessary for parameterised decision rules and reinforcement learning. In future work we exploit the similarities and differences between forecast and state information to develop decision models for trading strategies based on the learning methods described in this chapter. This completes the introduction to modelling techniques for sequential decision making, in the next section we review the use of sequential decision models for financial trading systems.

3.5 Review of Decision Models for Trading Systems

In investment finance, asset allocation strategies are based upon an investor's insight into future market behaviour that is expected to offer some perceived advantage over other market participants.

The development of statistical forecasting models of financial time series with statistically significant levels of predictability has led many researchers to test the economic significance of their models using dynamic trading systems. Initially, the optimisation of trading policies for statistical forecasts of asset returns was overlooked and implementation trivialised. In many studies the decision policy was implemented using simple heuristic trading rules, for example, a fixed buy/sell strategy given the sign of the predicted return (e.g. Refenes, 1993, Rauscher, 1997, Kollias and Metaxas, 1997, Gonzalez Miranda, 1995, Bramate, Colombo and Gabbi, 1997). The aim of these studies was to create a predictability test based on trading profitability, which would give more credibility to the standard statistical forecast accuracy tests and illustrate the potential effectiveness of the model.

During the last decade, the interest in profitability tests for financial forecasting models has continued to grow as they present clear intuitive understanding of potential value in a trading environment. Forecasting models with poor R^2 values, which for some applications would be considered irrelevant, have been shown to give potentially highly lucrative profits (Burgess, 1999). The apparent substantial profits from simple trading rules shows the sensitivity of trading performance to even low levels of predictability. The connection between more realistic trading evaluation began with the introduction of performance criteria that measured the risk adjusted return of trading strategy. These metrics had been widely used in investment finance and were a natural extension of the profitability test. Typical metrics include the Sharpe Ratio, Sterling Ratio, Maximum drawdown, etc, (e.g. Refenes, 1995, Moody, 1997, Weigend, 1997). Financial forecasting models were also linked with modern portfolio theory to provide a more formal framework for optimising portfolios conditional on statistical forecasts of expected asset return, volatility and correlation (e.g. Refenes, 1996).

More recently, modelling techniques for sequential decision making have been developed to optimise trading systems given some predictive information. These propose different methods to optimise the financial performance criterion and so improve expected trading performance. A number of studies provide evidence that separate optimisation of forecasting and decision models can lead to sub-optimal performance in financial trading systems (Neuneier, 1997; Bengio, 1997; Moody et al. 1997; Choey and Weigend; 1997).

Neuneier (1997) presents portfolio management using a Markovian framework and argues that practical asset allocation can be optimised by applying dynamic programming or neural network based reinforcement learning. He demonstrates how RL and DP methods can be used to optimise asset allocation and shows using an artificial example that the resulting policies are identical, although RL requires considerable more computing time. In addition, he considers a real world task of actively investing in the German stock index (DAX). The state vector consists of the level of the DAX plus 11 market variables and two neural networks are used to store the value function for the binary decision between investing in equities or cash. Results are compared to a prediction model and a passive strategy solely investing in the DAX with the RL policy significantly outperforming the two other strategies.

Bengio (1997) proposes using a differentiable trading policy so the forecasting model parameters can be tuned to optimise a particular financial performance criterion. He demonstrates the optimisation of a forecasting model to maximise trading performance for a heuristic trading rule that allocates funds between a portfolio of assets. Results show that a neural network forecasting model trained to optimise the financial criterion yields better out-of-sample performance than the model trained with a mean-squared-error prediction criterion.

Moody (1997) proposes to optimise the trading using a single model rather than to modularise forecasting and decision making. In this model the explanatory variables act as direct inputs into the model where parameters are optimised to maximise a path-dependent objective function. A novel form of recurrent reinforcement learning is used to optimise model parameters. The method approximates the current Sharpe Ratio using a Taylor series expansion to compute the influence of the portfolio return on the Sharpe Ratio in a single time period. A differential Sharpe Ratio is defined and the model optimised on-line to maximise expected trading system performance.

Choey and Weigend (1997) also propose that trading be optimised using a single model based on an artificial neural network. They present a method that explicitly maximises Sharpe Ratio so the output from the neural network is the proportion of funds allocated between a risky and a risk free asset. Novel iterative update rules are derived and compared to alternative approaches that only optimise profits. The technique demonstrates the need to explicitly control both the expected return and risk of a trading strategy using synthetic and empirical data.

These three studies highlight the performance improvements that can be gained by optimising trading systems using modelling techniques and by optimising models with respect to trading performance. There are a number of important issues that arise from the optimisation of trading systems. These include the development of trading systems based on either a single "large" decision model or two "smaller" individual models for forecasting and decision making. For example, if the inputs to an n dimensional trading system could be separated into two independent groups and smaller models generated then the combined number of states would, in theory, be smaller and so lead to a more parsimonious system. There are other issues relating to the integration or modularisation of models in a trading system. Potential advantages of modularisation include greater system transparency and integrity checks thereby assisting identification of model breakdown and misspecification, reuse of existing forecasting models for new objective functions, and also that the system is less prone to implementation difficulties that limit joint optimisation given non-differentiable, constrained objective functions. These topics are discussed in more detail in section 4.3.

In this thesis (chapter 5) we develop advanced trading rules to optimise trading systems given statistical forecasts. These are based on developing a parameterised class of trading rules that encompass a range of plausible trading rules and have the advantage of selecting the optimal trading rule given the trading objective (Towers, 1998). This approach allows the decision policy to be optimised given characterisation of the forecasting model and the decision environment, and some a priori restrictions to be functional form of the decision policy. These parameterised decision rules have been extended to path dependent decision rules (Towers, 1999) to optimise decision policies for a fixed level of transaction costs. Rules have been demonstrated on trading strategies using synthetic data and empirical forecasts based on statistical arbitrage. Results show considerable improvement in performance through optimisation of the trading system. More details of these methods are given in chapter 5.

In this thesis (chapter 5) we also develop an enhanced reinforcement learning method to optimise trading systems for predictions from forecasting models in the presence of more general trading costs, which may include market impact. Specifically, a Q-learning algorithm is tailored to exploit the properties of the financial domain. In particular, the learning process of the standard Q-learning algorithm is modified to take account of the partial independence between asset prices and trading decisions. This enables learning to proceed across all possible action-states in each time period, which we describe as "multi-action" learning. This approach effectively allows full exploitation and exploration in each time period and so significantly speeds up the learning process. Experiments are conducted, using Monte Carlo simulation, to show that performance can be improved against the myopically-optimal solution and that the rate of convergence is significantly increased compared against a standard Q-learning implementation. Further experiments are conducted to investigate the effects of market impact in the form of a trading restriction. Empirical results of an equity trading system demonstrate that the enhanced Q-learning can significantly increase performance for a 60% trading restriction (Towers, 1999).

In this thesis (chapter 7) we also propose a framework for performing a joint optimisation over a trading strategy and a forecasting model in order to maximise trading performance. An iterative algorithm is developed to optimise the construction of the forecasting model with two characteristics which both influence trading performance (Towers, 1999). Specifically, a *meta* parameter is used to control an optimisation criterion comprised of two characteristics, namely predictive correlation and prediction autocorrelation. The two modelling stages for forecasting and trading are then repeated until trading performance converges to the goal of joint optimality. Empirical evaluations (chapter 8) are conducted on a set of 50 identified "statistical mispricings" from the constituents of the FTSE 100. The results show that joint optimisation can significantly improved trading performance in the presence of transaction costs as low as 10 basis points.

3.6 Summary

In this chapter we have reviewed the main classes of modelling techniques for statistical forecasting and decision-making in investment finance. These methods are considered to be particularly relevant to the development of trading systems comprised of forecasting models and optimised trading strategies. We have discussed a range of forecasting methods that are commonly used in finance forecasting including smoothing techniques, ARMA modelling, cointegration, regression based modelling methods and adaptive techniques. These methods are suitable for capturing deterministic components of asset price dynamics. We have also reviewed the literature for the development of financial forecasting models which have been applied to empirical data. In addition, we have discussed the main sequential decision modelling techniques which are particularly pertinent to the development of trading strategies, such as decision rules and reinforcement learning. Finally, we have reviewed the literature for the development of models for trading systems which have been applied to empirical data.

In this chapter we have shown that there is a wide range of modelling techniques that have been developed for the two tasks of forecasting and trading. In particular, we have reviewed trading systems comprised of a single model, which effectively combines the two tasks into one model, or two models, one for each task. From our review we can see that there is conflicting evidence for the efficiency of these two comparative approaches. It is also not clear what level of predictability is required in order to construct a profitable trading strategy for a given level of transaction cost. In the next chapter we investigate the development and design of trading strategies for exploiting predictability and examine issues including system design, transaction cost, prediction characteristics and performance measurement.

4 Investigating Trading Strategies to Exploit Predictability

In chapters 2 & 3 we reviewed the evidence for predictability in financial markets and the development of modelling techniques for trading systems and financial forecasting models. In this chapter we investigate the requirements for implementing and optimising a trading strategy to exploit the predictability of a forecasting model. We focus on the key issues of system design and the effect of forecast model characteristics, trading costs and the trading performance criterion. After we have explored these issues we outline our proposed methodology for performing the joint optimisation of a trading system comprised of a forecasting model and a decision model.

The term "trading strategy" is widely used in the investment finance community, so in section 1 we describe what we mean by a trading strategy. In section 2 we use the foundations of decision theory, as presented in chapter 3, to develop a more formal definition of a trading strategy based on a Markovian decision framework. In section 3 we discuss two possible designs for implementing this system: the first using only one model to directly optimise trading and the second using two modelling stages, one for forecasting asset returns and the second to optimise the trading strategy given the predictions. The advantages and disadvantages of the two designs are investigated and shown to provide important motivation and justification for the two-stage modelling approach. In section 4 we assess the requirements of a trading system comprised of a forecasting model and a decision model. We develop a synthetic-trading environment with a controllable data generating process to emulate actual and predicted returns for a generic risky asset. This process provides a means of abstracting away from the construction of a specific time series model to directly assess forecast model requirements. We assess the trading performance of different trading rules and also to investigate the effect of two prediction characteristics, namely, predictive accuracy and prediction smoothness, in different trading environments.

In section 5 we discuss the costs that may arise during dynamic trading strategies and discuss the development of market impact models. We illustrate how the optimisation of trading under market impact requires decisions to take account of longer-term trading implications rather than simply optimise the short-term reward. In section 6 we discuss different trading performance criterion that are widely used in the finance industry and in section 7 we discuss the development of our methodology to tackle some these issues.

4.1 Description of a Trading Strategy

In investment finance, the term "trading strategy" is widely, and often loosely, used so in this first section we describe what we mean by a trading strategy to clarify the role of our research within the context of investment management.

In its most general form, a trading strategy is the implementation of an investment policy, which describes the allocation of funds over time within a universe or portfolio of assets. From this view, the asset allocation process quantifies what proportion of the fund that is allocated to which assets. The universe of assets forms part of the specification of the investment policy which is typically defined within the prospectus of a fund. The management of a fund can be typically broken down into two investment styles: strategic or tactical

Strategic investment consists of holding particular assets on the basis of some long term underlying investment strategy. In general, trading positions are held over a long period and no attempt is made to exploit any perceived market inefficiency through short term trading to generate greater wealth. For example, an index-tracking fund would be classified primarily as a form of strategic asset allocation that periodically tracks the weights of a specific index. In this case, trading only occurs when the weightings of the market constituents change significantly. This type of trading is also referred to as passive management as decisions are not controlled on the basis of some expert knowledge, that is designed to add value to the investment process, but only to replicate some market return. This type of fund often produces consistent or at least easily attributable performance with relatively low trading and management costs.

In contrast, tactical investment involves the analysis of a universe of assets in the attempt to identify a method of outperforming other market participants. This is often due to the experience and skill of a manager or some other competitive advantage. This style of trading is also known as active management and involves exploiting opportunities in asset prices often through short-term variations around some base allocation. For example, arbitrage strategies can be viewed as a form of tactical investment that exploit the price differentials between equivalent assets to generate excess profits.

In general, managed investment funds are composed of both active and passive components. The passive component explains the variation in performance due to the long-term mix of assets which is often related to investment characteristics, such as risk preference and cash requirements. Important active components include market timing and asset selection, which may be taken on a sector or individual basis. Market timing is normally attributed to short-term changes in allocation based on reducing exposure to "overvalued" assets while increasing exposure to "undervalued" assets. The active component can be viewed as the deviation of the fund from the passive mix and so acting as an overlay on the strategic investments.

As an example, an investment fund may wish to improve the performance of index tracking by including an actively trading component to exploit any identifiable market inefficiencies. Usually, this form of investment fund is best served by separating out the passive and active allocation tasks and controlling the proportions of the fund allocated to each component with risk control. In this thesis we assume that it is possible to separate out investment management into strategic and tactical components and so solely concentrate on the actively managed component of trading.

In active fund management, market-timing signals may be implemented using forecasting models that provide predictions of asset returns. In this thesis, we refer to a trading strategy as the

> *active component of an investment policy, which involves buying and selling assets, from within a universe of possible assets, based on statistical models of predictive signals or forecasts.*

This description assumes that investment management of a fund can be decomposed into active and passive components and efficiently recombined to optimise the trading objective. This enables the modelling of trading strategies for predictive forecasts without considering the implications of managing the passive component. Thus, in principle, independent "pockets" of predictability can be exploited individually. In some cases, there may be some nonlinear interaction effects between active and passive components but these are considered small and not the focus of this thesis.

In this thesis, rather than focus on specific trading nuances, which, in our opinion, remain firmly in the domain of practitioners, we instead take a methodological approach to implementing trading systems, focusing on methods of developing trading strategies. Despite focusing on methods, the style of trading that we consider being most appropriate for our approach is *arbitrage* trading which typically includes trading systems that are able to buy and sell asset positions cheaply. In chapter 8, we consider statistical arbitrage trading strategies for predictive models and construct a realistic application for equity trading.

4.2 Markovian Decision Framework for a Trading System

In this section, we build on the definition of a trading strategy to develop a trading system within the framework of a Markov Decision Process (MDP) in order to develop decision models to implement and optimise trading strategies. In chapter 5 we extend this work to explore specific methods to optimise a trading strategy for a given forecasting model. In the MDP framework a trading system can be described graphically as shown in figure 4.1.

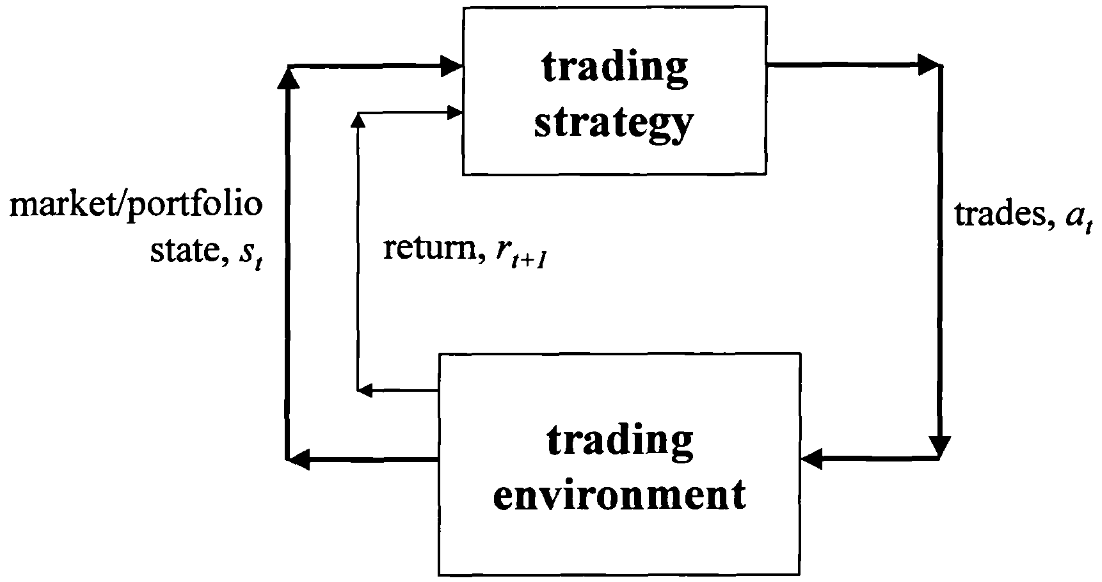


Figure 4.1 describes the trading strategy – trading environment interaction.

In figure 4.1 the *trading strategy* describes the process of optimising the investment decisions and the *trading environment* describes the status of the financial market and the portfolio. These two interact with the trading strategy selecting asset allocation decisions and the trading environment specifying trading costs and updating the asset price dynamics. The trading environment also determines the reward associated with each trading decision. The dynamics of the system progress with the trading strategy receiving some representation of the system state, and on this basis choosing a trading decision. In response to the action, the trading strategy receives a numerical reward and is presented with a new system state.

For simplicity, we consider the case of a trading environment consisting of a single risky asset and a portfolio consisting of a single trading position. In this trading system, the observed price of the risky asset can be monitored in continuous time but the trading position can only be rebalanced at a sequence of discrete time steps, t=0,1,2,3... as illustrated in figure 4.2.

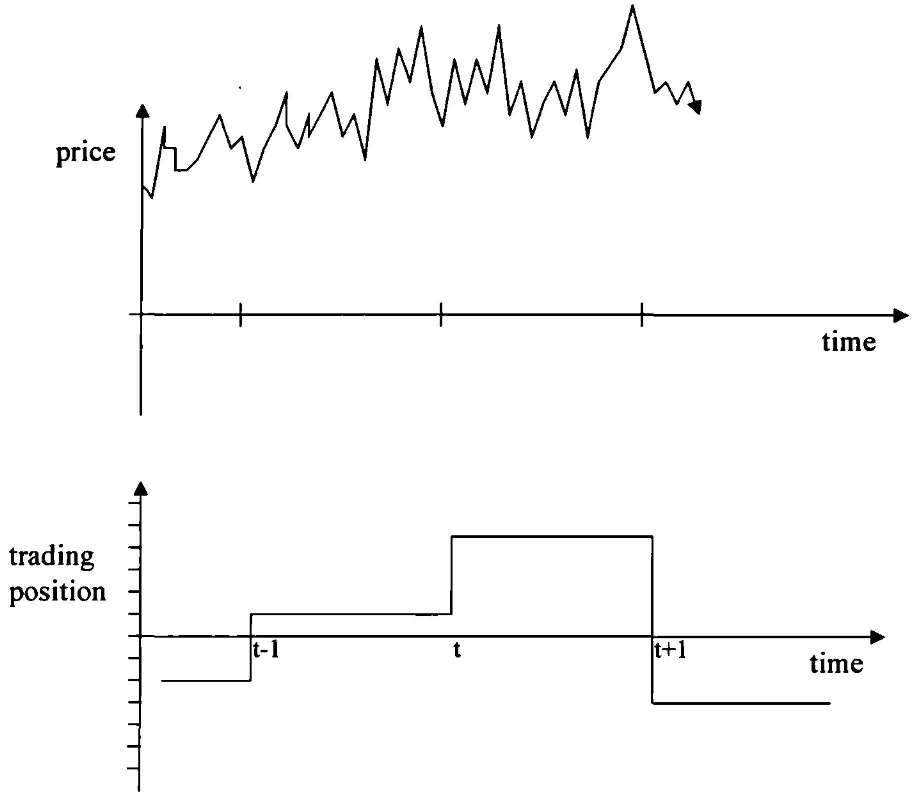


Figure 4.2: shows conceptually the two observed elements of the simplified trading system, with the top panel illustrating the price of a single asset and the bottom panel showing the trading position with rebalancing actions at three discrete points in time.

In figure 4.2 the trading position represents the fractional allocation of the fund to the risky asset, which can be both positive and negative. Negative trading positions represent short selling or, in the case of an overlay, an underweight allocation. A trading position of zero represents a risk neutral or benchmark trading position.

We can further simplify this system by assuming that the trading strategy does not require continuous monitoring of the asset price but only the condition of the trading environment at the time of rebalancing the trading position. In principle, higher frequency price data can be stored as a state variable. In this framework, the trading strategy describes a mapping from the system state to an action state. Methods for implementing the trading strategy will be discussed in detail in chapter 5. In the next three subsections we provided a detailed description of actions, state and rewards for a trading system.

4.2.1 Actions

In this Markovian framework for a generic trading system, we define the action state to represent the set of possible trading positions. Note that the action state is <u>not</u> defined as the change in the trading position, which may typically describe rebalancing instructions, but the rebalanced level of the trading position. This specification allows the previous trade to represent the state of the portfolio and so avoids the problem of converting action states into trading positions.

In theory, MDP's require the action state to be finite, however for most trading systems assets can be purchased in small discrete bundles in comparison to the size of a typical fund. Thus possible trading positions are typically thought of as a continuum between some imposed trading limits or constraints, which are often defined by the size of the fund and possibly the trading style. For example, arbitrage-trading strategies typically require simultaneous trading of a number of assets, taking a mixture of short and long trading positions, while maintaining a market neutral portfolio. For these types of trading strategies, continuous trading positions may be approximated by quantisation of the action-state space. However, for realistic approximations a large action space is typically required which is often impractical for employing dynamic programming methods. This highlights one of the benefits of employing reinforcement learning or approximate dynamic programming methods which include a function approximator, such as an artificial neural network. This allows the finite MDP restriction to be relaxed to allow a continuous state-action space which more accurately models the decision problem.

Trading positions at some time step are thus defined by some fixed finite or continuous action space, $a_t \in A$. However, we may also consider situations in which the space of possible trading positions is dependent on the state of the system. For example, if we consider transaction costs that arise from market impact then one simple model may be to define a trading restriction, which limits trading from the previous trading position. Trading outside the trading restriction is considered to generate prohibitively large transaction costs. In this case, the action space is defined as $a_t \in A(s_t)$, where s_t is the state, which defines the previous trading position, and A is the universe of possible trading positions. The modelling of market impact is discussed in more detail later in this chapter. From this definition of an action state we can assume that every admissible action is defined by some stationary policy, defined as $a = \mu(s)$, where μ is some function which maps states to actions.

4.2.2 State

In section 3.4.1 on decision theory, we stated that, "the system state describes the condition of the system and determines all aspects of future behaviour independently of how the state is reached." For this to hold, the state must contain all information that is relevant to future behaviour.

More specifically, for a Markovian framework of the trading system, the state consists of the previous trading position, a_{t-1}, the current price of the asset, p_t, plus any other trading information, I_t, relevant to future trading behaviour, which at time t, takes the form:

$$s_t = \{p_t, a_{t-1}, I_t\} \quad where \quad s_t \in S_t; a_t \in A(s_t) \tag{4.1}$$

The information set I_t can be expressed as

$$I_t = \{p_1, p_2, ..., p_{t-1}; a_1, a_2, ..., a_{t-2}; \mathbf{x}_1, \mathbf{x}_2 ..., \mathbf{x}_n; r_1, r_2, ..., r_{t-1}\} \tag{4.2}$$

where p_i is past asset price, a_i is the trading position and r_i is the investment return with lag i, and x_i is some vector of arbitrary variables, which may typically consist of exogenous variables or transformations of prices.

The arbitrary variables fall into two groups depending on which aspect of future behaviour they influence. For example, a macro economic variable may influence the price of the asset while current trading volume may influence cost of changing the trading position. We partition these into two groups of variables, x_i^F and x_t^C, which represent variables affecting forecasting and the cost of trading respectively. We assume that forecasting variables do not influence the cost of trading and that trading costs variables do not influence the asset prices. This partitioning is considered natural due to the inherent difference between forecasting and trading.

Under this assumption, the state may then be compressed by constructing a forecasting model of the asset price, $\hat{p}_{t+1}$, with variables drawn from the sub-state describing the forecasting variables and the past asset prices, which, in general, can be expressed as

$$\hat{p}_{t+1} = f(p_1, p_2, ..., p_t; \mathbf{x}_1^F, \mathbf{x}_2^F, ... \mathbf{x}_n^F) \tag{4.3}$$

where f is an arbitrary function.

The construction of a forecasting model has the advantage of reducing the system space and thereby decreasing the number of degrees of freedom, which is directly related to model variance. This is particularly effective if the forecasting method includes some variable selection method that is capable to removing statistically insignificant variables (Burgess, 1999). This issue is discussed in more detail in section 4.3. For a trading system that includes a forecasting model the state can be rewritten as

$$s_t = \left\{\hat{p}_{t+1}, a_1, a_2, ..., a_{t-1}; r_1, r_2, ..., r_{t-1}; \mathbf{x}_t^C\right\} \tag{4.4}$$

where $\hat{p}_{t+1}$ is the predicted asset price.

Thus the sub-state which does not affect trading has been compiled by the forecasting model into a single forecast. In addition to definitions of action and states, a Markovian framework also requires definition of state-action transition probabilities, $P(s_{t+1} \mid a_t, s_t)$ which define the probability of moving to a new state given an action in the current state. In our framework we assume that state-action transition probabilities are, in general, unknown. This is typical of complex decision systems containing single step forecasts and is one reason for developing a computational modelling framework for optimising trading strategies in chapter 5, rather than relying on analytical or dynamic programming methods.

4.2.3 Rewards

The reward associated with each trading position represents the investment return received at the next observation of the system. In trading systems, rewards from trading decisions are not instantaneous but delayed until the next price is observed. We therefore attribute a reward to the selected action at time t, although the actual time the reward is received is at $t+1$.

More specifically, the reward r_t at time t is influenced by the trading position, a_t, the change in the asset price, Δp_{t+1} from time t to $t+1$, and the trading costs, T, and may be expressed as

$$r_t = \Delta p_{t+1} * a_t - T(\Delta a_t, x_t^C) \tag{4.5}$$

where T is some transaction cost function specifying the costs due to the change in the trading position and the state of the market.

In equation (4.5), the first term equates to the immediate profit or loss assuming no trading costs and the second term approximates the transaction cost due to changing the trading position. In the simplest case of a fixed transaction cost rate, the transaction cost function is proportional to the magnitude of the change in the trading position. This rate can be specified if we assume that costs are approximated by the average observed bid/ask spread of the traded asset. In this case, if the asset price is the "mid price" then the transaction cost rate is half the observed bid/ask spread. In practice, however, transaction costs may also arise from a number of other sources such as market impact, price slippage, taxes, etc. In section 4.5 we discuss the influence of practical trading costs in more detail and discuss the modelling of market impact.

In our Markovian framework, the trading strategy only revises the trading position at specified time intervals. If we assume that trading returns are only determined at each trading opportunity then the overall investment reward is an function of the sequence of trading decisions. If we also assume that the trading strategy continues for an infinite number of time periods then the expected profitability *R*, can be expressed as the discounted sum of immediate rewards *r*, defined as

$$R = E\left[\sum_{i}^{\infty} \gamma r_{t+i}\right] \tag{4.6}$$

where γ is a discount factor.

For a trading system, equation (4.6) represents the expected trading performance of a risk neutral strategy. In many trading situations, however, investment considerations must also be given to risk or uncertainty. In a Markowitz mean-variance framework this gives rise to optimising a risk-adjusted trading strategy, $\mu^*(s,\lambda)$, in the form

$$\mu^*(s;\lambda) = \arg\max_{\mu}\left[V^{\mu}(s) - \lambda\sigma^2(V^{\mu}(s))\right] \tag{4.7}$$

where λ is the risk aversion parameter, $V^{\mu}(s)$ is the expected value of following a trading policy, μ, from state s, and $\sigma^2(V^{\mu}(s))$ is the variance of this value.

For dynamic programming methods, it is difficult to compute the variance of the value function at a given state, due to the non-recursive nature of variance estimation. Decision modelling techniques, however, have been developed for optimising trading strategies for risk adjusted performance metrics. Examples of modelling techniques that optimise Sharpe Ratio include direct reinforcement learning (Moody et al., 1998), neural networks (Choey and Weigend, 1997) and parameterised trading rules (Towers and Burgess, 1998). It is anticipated that these techniques can be simply modified to optimise other measures of risk adjusted return such as the Sterling Ratio, which takes account of cumulative drawdown.

To further clarify the relationships between asset price changes, trading positions, investment rewards and trading performance, we devised a synthetic trading system, based on a simple reversal strategy, where the trading position is restricted to long and short positions of fixed, unit magnitude. This trading system is illustrated for 20 time steps in figure 4.3.

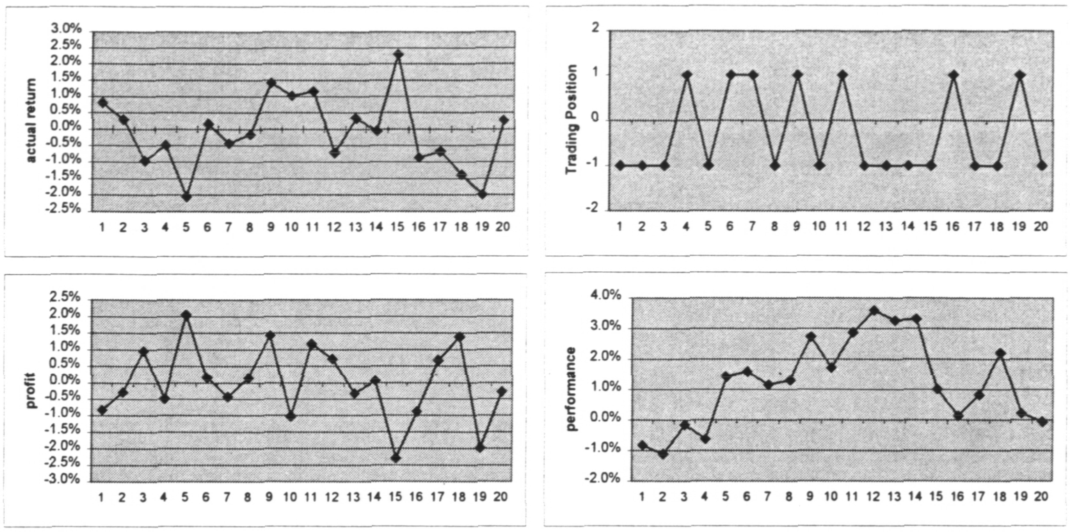


Figure 4.3: Synthetic asset returns (top left panel), trading signals (top right panel), profit per time step (bottom left panel) and trading performance measured as cumulative sum of profits (bottom right panel). Profits assume no trading costs.

The top left panel in figure 4.3 shows the actual asset returns, the top right panel shows the simulated trading positions, the bottom left panel shows the profit from each trading position and the bottom right panel shows the trading performance, in terms of cumulative profit. The top right panel provides an example of how the trading position is reassessed at each time point and held constant over a next time step. The trading position would be controlled by some trading policy based on the observed state of the system and the predicted return.

In this section we have developed a general Markovian framework for a trading system. In the next section we discuss two possible designs of the trading system, either a single model for the two tasks of forecasting and trading, or two modelling stages, one for each task. We also statistically compare the two designs and provide motivation for the two-stage modelling approach.

4.3 Trading System Design

In the last section we described a Markovian decision framework for a generic trading system. In this section we consider alternative methods of implementing this system using forecasting and decision models. Two designs are explored into detail, either a single decision model to optimise both forecasting and trading tasks or two separate models, one for each task, with a forecasting model to predict the asset returns and a decision model to optimise trading given the predictive information. These two approaches are illustrated in figure 4.4.

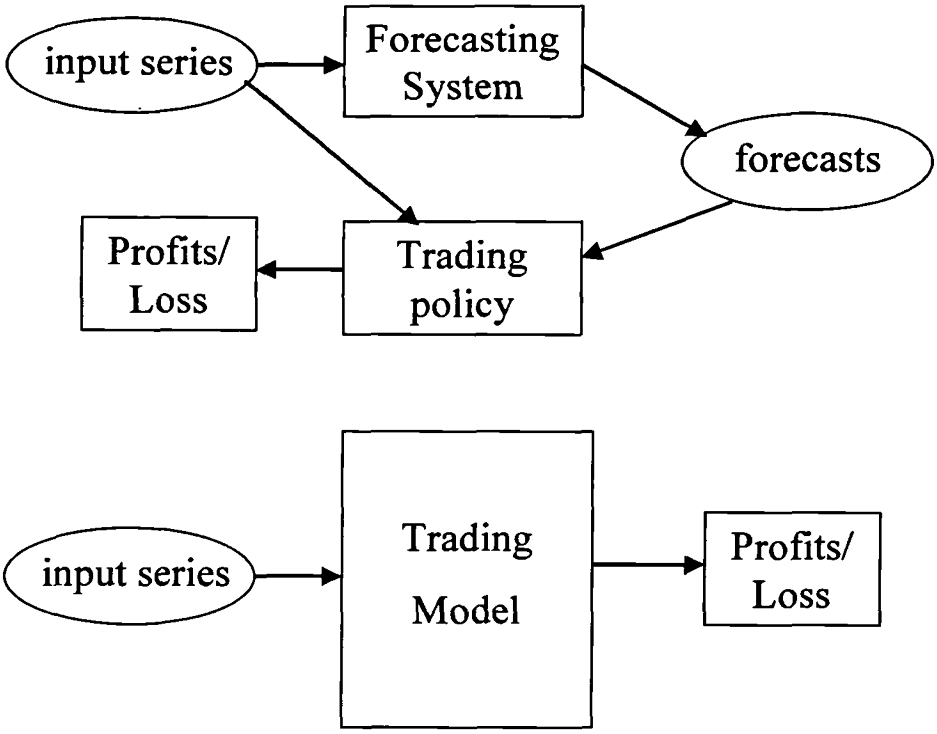


Figure 4.4 illustrates the two trading systems designs: the top diagram depicts the two stage trading system and the bottom diagram shows the single model trading system

In the single model approach the input series, as described in equation (4.1), are explanatory variables into some modelling framework. In the dual modelling approach the input series are effectively partitioned into explanatory variables for the forecasting model, as described in equation (4.3), and state variables for the trading model. The output of the forecasting model is considered another state variable, as described by equation (4.4). There are a wide range of methodologies with which to implement forecasting models and decision models to accomplish these tasks (as described in sections 3.2 and 3.4) and both require some performance criteria to optimise the construction of the models.

Given these two designs, which one provides the optimal trading system?

The single model approach is intuitively appealing with only one optimisation criterion that can be applied to the construction of all models. The two-stage approach, however, has the added complication of having two different optimisation criterion for each of the two tasks of forecasting and decision-making. For instance, in the case of two models, the <u>ultimate</u> objective is to maximise trading performance, but this can only be evaluated in the context of a trading system that already consists of a forecasting model. Thus, whilst trading performance can be used as a criterion for optimising the decision model, estimation of the forecasting model must be performed before the decision model is known and hence performed using an alternative criterion, typically to minimise prediction error.

This issue was first highlighted by Moody (1998), where he proposed that trading systems consisting of two modelling stages created a "forecasting bottleneck" as potentially useful information may be ignored and so result in a sub-optimal trading system. In particular, as only a single forecast is transmitted to the decision model, other information may be discarded which may have proved otherwise useful for optimising the trading strategy. This feature of a forecasting bottleneck can be highlighted for a trading system which combines a predictive model and a trading policy. In this case the trading system includes a forecast module with adjustable parameters, θ, followed by decision module with parameters, θ', as illustrated conceptually in figure 4.5.

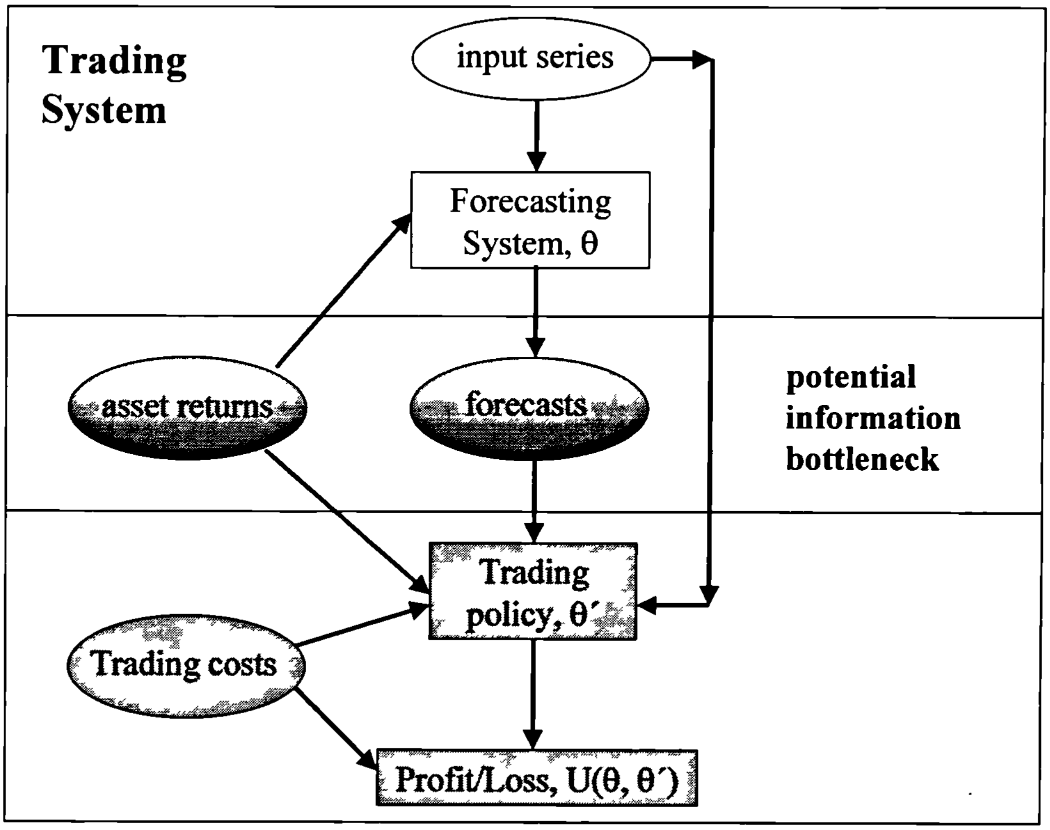


Figure 4.5 is an illustration of the "forecasting bottleneck" with separate optimisation of the forecasting system and trading strategy. The bottleneck indicates that potentially useful information may be lost as it is only transmitted to the trading policy in the form of a single forecast.

In figure 4.5 the performance criterion, $U(\theta, \theta')$, is indirectly optimised by forecasting the future asset returns and then selecting actions on the basis of the trading policy. The forecasting system is constructed using a methodology to select the optimal set of inputs from the set of candidate variables. The forecast model is adapted by varying the parameters, θ, to minimise the forecast error (typically based on mean squared error). The trading policy is then constructed with forecasts and state information as inputs, by varying the decision model parameters, θ', to maximise the expected performance criterion.

The performance is now dependent on two sets of parameters, θ and θ'. This method has the disadvantage that it cannot simultaneously vary the two model parameters and may have a loss of information from "compressing" the explanatory variables into a single forecast. It is suggested that the performance of this type of system is likely to be sub-optimal. One method of overcoming the forecast bottleneck is to discard the forecasting model and use just a single model. In this approach the trading model integrates both modelling stages into one. The model directly optimises trading on the basis of some explanatory variables that act as input variables of the system so the model can directly optimise the trading performance criterion. This approach avoids any loss of information from implementing a multistage modelling system by implicitly performing a joint optimisation of forecasting and decision models via one model and so directly adjust one set of model parameters, θ, to optimise trading performance. In chapter 7 we proposed a methodology to overcome the limitation of the two-stage optimisation approach by effectively "closing the loop" in the model design process and so removing the "forecasting bottleneck". The basis of our approach is to jointly optimise the two model components by developing an iterative methodology, which both evaluates and improves the two modelling stages in terms of expected trading performance.

However, the optimisation of a single model has a number of potential difficulties that explain why it may still be preferable to consider generating a separate forecasting model and a decision model for the design of a trading system. Next, we explore these aspects in more detail.

The first issue is the selection of the input variables to describe the possible state space described in equations (4.1) and (4.2). In trading systems the system state space may consist of a vast number of exogenous variables that may influence the performance of the trading strategy. The construction of a single model means that we cannot directly identify any relationships between the explanatory variables that are considered to drive the price of financial instruments. The development of a forecasting model has the benefit of being able to test the statistical significance of explanatory variables using well-defined model selection procedures. Typically decision models have no variable selection method so it may be appropriate to compress explanatory variables by building a forecasting model with input selection.

Another issue is the implementation of a single trading model to optimise the performance criterion. Unlike forecasting models, decision models have no observed target variable, which typically makes the modelling process more complex. Decision modelling techniques, which employ function approximation techniques, are difficult to optimise and require large quantities of data. For most methods, complexity increases sharply with additional input variables so often making the single model impractical.

Another advantage of separate optimisation is that *a priori* knowledge of the relationship between forecasts and actions can be applied to the modelling process. For example, a positive predicted return should generate a positive or overweight trading position and conversely, a negative predicted return should equate to an underweight or negative trading position. This knowledge of the decision task enables structured decision modelling techniques to be applied to the trading system problem (Towers, 1999).

A single model also lacks transparency with no forecasts of asset returns to check model adequacy or model reliability. This often proves to be important in finance modelling situations where high noise to signal ratios often lead to misspecified models and where model transparency acts as a useful "safety check".

In general, the advantages of two modelling stages are as follows:

- are the ability to divide the information set into two subsets (with a consequent reduction in model complexity),
- potential inclusion of a priori knowledge,
- use of existing forecasting models,
- trading strategies with different objectives from the same forecasting model
- a more transparent trading system.

In addition to these arguments for and against the two model designs in the next sub-section we consider these two modelling approaches for a statistical modelling perspective. We compare these two competing designs from the perspective of model generalisation for different modelling techniques and also with the inclusion of irrelevant input variables.

4.3.1 Statistical comparison of Single and Dual Modelling Approaches

In this section we compare the two trading system design approaches, namely single or two models, from a statistical modelling perspective.

In general, the purpose of constructing a model is to represent the underlying data generating process of the system. However models typically fit both the underlying deterministic signal plus the noise that influences the observed value. The ability of a model to estimate values for previously unseen data is usually referred to as *generalisation*. This property is related to the complexity or design of the model building process. A model with too few coefficients exhibits poor generalisation since it has too little flexibility, however too many coefficients also leads to poor generalisation since the model fits too much of the noise, which is specific to the existing data and will not be same in the future.

The influence of model complexity on generalisation reflects the need to optimise the trading system design. The complimentary properties of bias and variance are typically used to decompose the *generalisation error* (Geman et al, 1992). A large bias reflects too simple a model while a high variance reflects too much model flexibility with respect to particular data set. The aim of model design is to minimise generalisation error by controlling the effect of model bias and variance. The simplest approach of assessing generalisation performance of estimated models is to use a validation (or out-of-sample) data set. The performance of the model over this period is considered as unbiased measure of generalisation. Other measures have been developed based on a prediction error *PE*, which consists of two terms, in-sample error and a complexity term, and can be defined as

$$P.E. = \frac{E}{N} + \frac{\gamma\sigma^2}{N} \tag{4.8}$$

where E is the sum of squared errors, N is the number of observations, γ is the number of free parameters, and σ^2 is the variance of the noise, which is $1-R^2$ for linear regression.

In equation (4.8), the complexity term has three factors, number of model parameters, observations and noise variance. If we increase the number of parameters of the model or variance of the noise then we effectively increase model complexity. However, if we increase the number of observation then we decrease model complexity. Procedures have been developed to approximate the parameters in equation (4.8), most notably the *generalised prediction error* which estimates the *effective* number of parameters in a neural network (Moody, 1992). For a more detailed treatment of generalisation and the bias variance trade-off, see Chapter 6 and 9 of Neural Networks for Pattern Recognition by Bishop (1995). In order to compare the two model designs for a trading system, we can assume that the system with the fewest number of degrees of freedom will be the most parsimonious model, if all other factors are equal.

In order to compare the two competing model designs consider a simple modelling approach where models are implemented using a bin-smoother, which acts a statistical lookup table. These represent a simple class of models that divide the input space into a set of boxes, which act as "bins" to collect the observed values. Lookup tables are commonly used in dynamic programming (and reinforcement learning) methods to approximate value functions for a discrete state space. Bin smoother models are capable of representing both linear and non-linear relationships, however function discontinuities at bin boundaries cause jagged estimated functions. A parameter is required to specify the bin width, which is selected on the basis of a trade-off between a smooth function approximation and model complexity. A small bin width produces a smooth response function but has high dimensionality while a large bin width leads to irregular functions but of low dimensionality.

In order to determine the number of free parameters in the two systems we assume that a model has a discrete number of bins (or states), denoted by s, for each input variable. We also assume that the total number of inputs can be partitioned into two groups for forecasting and trading, as discussed previously. In this scenario the sets k and m represent the inputs variables into the forecasting and trading modules respectively. If we use a single model to integrate across all input variables and simultaneously optimise forecasting and decision making, then the model has a number of parameters equal to $s^{(m+k)}$. Alternatively, if we use two models, one for each modelling phase, then the number of parameters is equal to $s^m + s^k$.

If we now consider the number of parameters required for the two systems then the two stage modelling process outperforms the single model from the following inequality:

$$s^m + s^k \leq s^{k+m} \tag{4.9}$$

for all values of k, $m > 1$.

Equations (4.8 and 4.9) show that the integrated single model will lead to a higher model variance than the two stage approach as illustrated in figure 4.6.

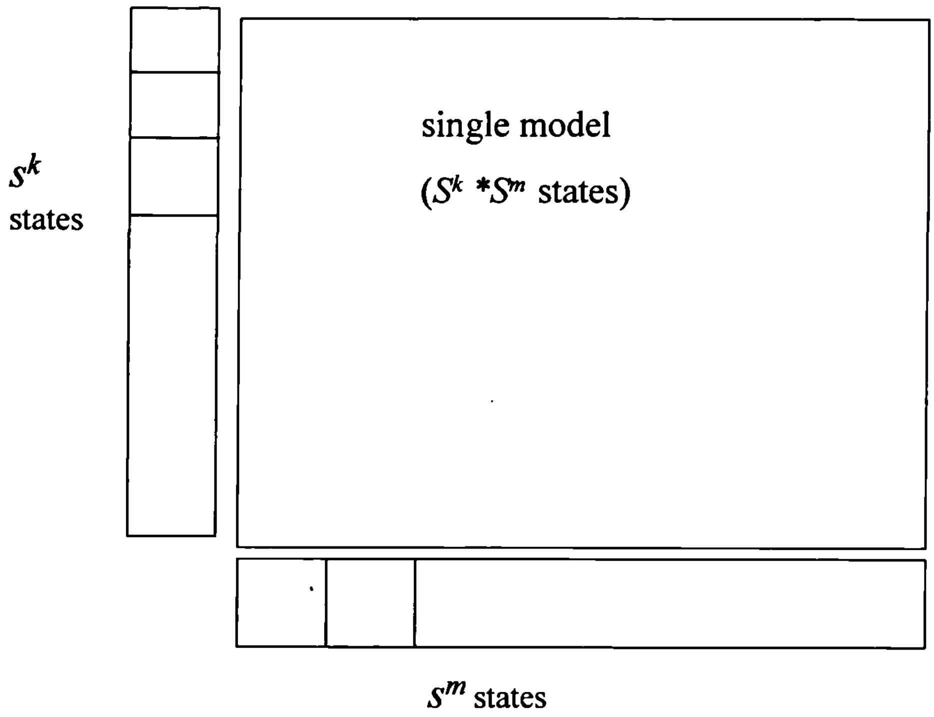


Fig 4.6 conceptually illustrates the number of parameters for a trading system with either one or two models implemented using a bin-smoother model. The two grey boxes depict the number of parameters for a system using two models and the white box shows the number of parameters required for a single model.

Figure 4.6 illustrates the difference in the number of free parameters and subsequently the model variance between the one and two stage modelling approach. The grey boxes highlight the number of states required for a modular trading system and the white box depicts the multiplicative nature of the parameter requirements of the single model. As well as bin smoother models, these results are also indicative of similar classes of function approximators including polynomial and Radial Basis Function networks. For these two designs we can assume that the representational ability and the bias are the same whilst the variance of the single model is higher provided the data partition assumption is correct.

If we consider other modelling types, for example, linear function approximation methods, then the number of parameters for the two modelling approaches is almost equivalent with the number of parameters for the single and dual models equal to $n+m+1$ and n+m+3 respectively. The two additional parameters in the two-stage process are due to a coefficient for the output of the forecasting model and an extra constant term. This comparison holds for similar modelling approaches including neural networks, which do not suffer from the curse of dimensionality. In this case, it appears that the single model has a slightly lower model variance than the two-stage approach.

To complete the study of these two competing systems we consider a system with a number of irrelevant variables. This problem is particularly common in modelling financial data, as there is generally a low level of predictability, which can easily lead to variable selection bias even for the most robust variable selection methods. The aim of this further study then is to discover which design has the best generalisation when irrelevant variables are included.

A synthetic framework was devised consisting of a target or dependent variable that is the result of a data generating process which is partially deterministic and partially stochastic. The deterministic component is constructed from two explanatory variables, one of which is known and the other is within a group of candidate variables. Any other variables within the candidate set are then considered as noise variables. This system is similar to a trading system which has one variable that influences trading and also some predictive variable hidden within a group of candidate explanatory variables. Given this scenario, we can devised an artificial system with a pool of candidate variables and a target variable, y, defined as

$$y = x_1 + x_2 + \varepsilon \tag{4.10}$$

where x_1 and x_2 represent the predictive variables for trading and forecasting respectively and ε is the contaminating noise.

A model of the system can either be constructed using a single, joint model or two separate models as previously discussed. The single model may be defined as

$$\hat{y} = c + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_5x_5 + b_6x_6 + \ldots.. \tag{4.11}$$

where b_1, b_2, b_3 , b_4 are the estimated model coefficients and the input variables x_3, x_4 ,x_5 , x_6 represent additional irrelevant variables that cannot be distinguished *a priori* from the two explanatory variables, x_1 and x_2.

The alternative approach is to construct two separate models, one for "trading" and the other for "forecasting". The output of the first model, the forecasting model, is then an input into the second, the "trading" model. If we assume the trading variable, x_1, can be partitioned from the predictive variable, x_2, and the irrelevant variables, then the two models are defined as

$$\begin{aligned} \hat{y}_1 &= c_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_5x_5 + b_6x_{61} + \ldots. \\ \hat{y} &= b_1x_1 + b_f\hat{y}_1 + c_2 \end{aligned} \tag{4.12}$$

where b_f is the estimated coefficient and $\hat{y}_1$ is the output of the forecasting model.

Simulation experiments were conducted to compare these two synthetic trading systems, as described in equations (4.11) and (4.12). We used 30 observations for the in-sample data set and estimated the models for the two designs, using linear regression, for a range of irrelevant variables from 1 to 4. The models were then evaluated over a test data set of 100 observations in order to generate a measure for their ability to generalise to previously unseen data. For consistency, simulations were repeated 100 times and the results over the test data sets summarised in table 4.1

Average R^2	**Number of Irrelevant Variables**				
	0	**1**	**2**	**3**	**4**
Single	47.93%	45.91%	43.04%	42.23%	39.28%
Two	47.93%	47.05%	45.83%	45.29%	43.96%
Difference	0	1.14	2.79	3.06	4.68

Table 4.1 shows the average out-of-sample R^2 for the two competing modelling systems with up to four irrelevant input variables.

Results from these experiments show that the two stage modelling approach consistently outperforms the single model for systems which contain additional noise variables. Where no noise variables are considered the out-of-sample performance from the two systems is equivalent. When additional noise variables are added the performance of both systems degrade but the performance of the single model system deteriorates faster, as shown by the difference row. In the case of irrelevant variables, the results suggest that the two-stage approach has a lower model variance than the single model.

This analysis reinforces the view expressed previously that a two stage modelling system can be justified for a trading system. Both pragmatic arguments and statistical analysis suggest that trading systems comprised of separated forecasting and trading models are often a more appropriate system design then the single joint model.

In the next section we focus on investigating trading systems constructed using this two stage approach. We investigate the effects on trading performance of the properties of forecasting models, the implementation of the trading strategy and different sources of trading costs. A synthetic trading system with controllable characteristics is devised to demonstrate the effects of these factors.

4.4 Synthetic Trading System

In this section we develop a framework for a synthetic-trading environment which simulates a trading system based on a statistical forecasting model. The system is designed to assess the profitability of a financial forecasting model and to act as a benchmark for the investigation for different prediction characteristics, trading strategies and transaction costs.

The system is evaluated using a data generating process that simulates the output from a forecasting model of a single risky asset. The framework allows for the control of two characteristics of the predicted returns, namely prediction accuracy and prediction smoothness. The prediction accuracy is specified as the correlation between the predicted return and actual return and the prediction smoothness is specified as the first order prediction autocorrelation. These two characteristics are representative of the features present in predicted returns from typical forecasting models and are both expected to influenced trading performance. The development of a data generating process is particularly useful as it enables a generic forecasting model to be emulated without actually constructing a specific model. The trading strategy for the predicted returns is implemented using simple but commonly used trading rules that convert the forecast into a trading position. In addition, the influence of typical transaction costs, from bid/ask spreads, associated with different assets are incorporated into the trading system.

To formalise this process, let some asset return Δy_t be comprised of a deterministic component and a stochastic component which are represented by variables x_t and ε_t respectively and defined as

$$\Delta y_t = \beta x_t + \varepsilon_t \qquad \textit{where } \varepsilon_t \sim NID(0, \sigma_\varepsilon^2) \tag{4.13}$$

$$\textit{and } \; x_t = \varphi x_{t-1} + \eta_t \qquad \textit{where } \eta_t \sim NID(0, \sigma_\eta^2)$$

where β and φ are coefficients and ε_t and η_t are two Gaussian processes with variance σ_ε^2 and σ_η^2 respectively.

In equation (4.13), the coefficient β represents the strength of the deterministic variable x_t, while the coefficient φ introduces autocorrelation into the time series. To specify the variance of the two Gaussian processes, suppose that the variance of the explanatory variable is equal to the variance of the actual asset returns. If we standardise the distributions of the two variables then the variances of the two Gaussian processes are defined as

$$\sigma_\varepsilon^2 = 1 - \beta^2 \tag{4.14}$$

$$\sigma_\eta^2 = 1 - \varphi^2$$

where the coefficients β and φ are subject to the constraints, $\varphi < 1$ and $\beta < 1$.

Using this formulation for the variance of the two processes means that the coefficient, φ, controls the level of autocorrelation in the explanatory variable x_t, and the coefficient β, controls the correlation between the asset return and the explanatory variable.

Now, if we assume that the deterministic component of the asset return can be captured in a statistical forecasting model (i.e. the model is maximally predictive), then the predicted return series $\Delta\hat{y}$, is defined as

$$\Delta\hat{y} = \beta x_t \tag{4.15}$$

Given this formulation of the predicted return, the coefficient, β, controls the level of correlation between predicted and actual returns, which is considered to be the measure of prediction accuracy. Similarly, the coefficient φ, controls the level of autocorrelation in the predicted returns, which is considered to the measure of prediction smoothness. We have now defined a data generating process for a forecasting model, with parameters β and φ which independently control the degree of prediction accuracy and prediction autocorrelation in the predicted return series respectively.

Now, given a predictive asset return, a simple but plausible trading strategy might be to buy a fixed amount of the asset if the prediction is positive and sell an equal amount if the prediction is negative. This naïve asset allocation rule was first attributed as the Merton Measure of Market Timing (Merton, 1981) to provide a simple measure of out-of-sample predictability in terms of trading profitability. The rule is explained as follows: if the asset return in the next period is forecast to exceed the risk free rate then invest completely in the stock, otherwise invest in a risk free asset such as cash. This rule can be generalised to include negative trading positions which allow for short selling.

More formally, let the fraction of the asset allocated at time period t be denoted by a_t and the asset allocation strategy defined as

$$a_t = \begin{cases} m & \text{if } \Delta\hat{y}_t > 0 \\ -m & \text{if } \Delta\hat{y}_t \leq 0 \end{cases} \tag{4.16}$$

where $\Delta\hat{y}_t$ is the predicted return in excess of the risk free rate and m is the fixed trading position.

In equation (4.16), the trading position only takes two values depending the sign of the predicted return. This trading rule doesn't respond to the magnitude of the predicted return but only to the sign of the predicted return.

Alternatively, we may define a trading rule that takes account of the sign and the relative size of the predicted return. For example, we may define a trading rule so that the size of the trading position is linearly proportional to the predicted return, in which case:

$$a_t = m\Delta\hat{y}_t \tag{4.17}$$

where m is the magnitude coefficient that controls the size of the trading position relative to the size of the predicted return. It is assumed that m is non-zero and positive.

In equation (4.17) the trading position may take any value depending on the predicted return and the magnitude coefficient, m. Under practical trading conditions, the trading position is typically constrained by market liquidity, which specifies the minimum tradeable quantity and results in some quantisation of the trading position. However for the purposes of this trading system we assume that there is sufficient market liquidity for a continuous space of trading positions.

We have now defined a controllable data generating process for predicted and actual asset returns and two trading rules, which can be used to convert predicted returns into trading positions. Trading returns for a particular rule can now be determined given some approximation of transaction costs. For an given asset, transaction costs can be estimated from the bid-ask spread, which is observed in exchange traded markets and may vary according to market conditions. For the purposes of this simplistic trading environment we assume that the transaction cost is constant over the duration of trading (i.e. ignoring any time varying component of market impact, taxes, etc.). In equation 4.13, the data generating process for the asset price return assumes a standardised variance. To simulate the behaviour of a particular asset we need to multiply the synthetic returns by the estimated volatility of the asset. This can be estimated by measuring the standard deviation of the asset returns using an empirical data set. If we assume no market impact and a "linear" transaction cost rate, then the approximation of the percentage trading return, *r*, for holding a trading position over time period *t* can be defined as

$$r_t = a_t \Delta y_t - \frac{T}{\sigma}\left|a_t - a_{t-1}\right| \qquad (4.18)$$

where a_t is the percentage of the fund invested in the asset y, T is the estimated percentage transaction cost, Δy_t is the standardised asset return and σ is the asset volatility.

In equation (4.18) the first term determines the profit or loss from trading and the second term the cost of changing the trading position. The volatility parameter is dependent on the asset type and the trading horizon and so will increase for longer time horizons and decrease for short-term trading. In this equation we have assumed only a fixed transaction cost rate and no costs from market impact, illiquidity, slippage, etc. In section 4.5 we discuss the inclusion of nonlinear market impact terms. The transaction cost parameter is simply defined by half the typical bid/ask spread, which has relatively low variability for large capitalisation assets in well-established markets. However, bid/ask spread often increase in periods of low trading volume (e.g. near the close of the trading day) or times of higher than usual market volatility.

To estimate typical bid-ask spreads, we inspected a range of asset classes across different well-established markets at different times of the trading day using a Reuters terminal. The bid-ask spreads are summarised in table 4.2.

	Bid/Ask Spread
Equities	5-50 b.p.s
Bonds	5- 25 b.p.s
Equity Index Futures*	3-15 b.p.s
Bond Futures*	2-10 b.p.s
Interest Rate Futures*	1-5 b.p.s

Table 4.2 shows the typical range of observed transaction costs of different asset classes in basis points. Note that futures do not have bid/ask spreads but a cost per future which can be converted into an equivalent bid/ask price given the value of the underlying asset.

In this section, all the basic properties of a synthetic trading system to exploit predictability have been developed. These include a data generating process to emulate predicted returns from a forecasting model with controllable characteristics, trading rules that convert forecasts into trading positions, and investment returns from trading that include transaction costs. In the next section we use this system to investigate, using Monte Carlo simulation, the effects on profitability of two prediction characteristics and trading different asset classes.

4.4.1 Simulation Experiments

In this section we describe simulation experiments which were devised to test the effect on profitability of trading different asset classes on the basis of simulated forecasting models with controllable prediction characteristics. Different trading scenarios were devised for the trading system, developed in section 4.3, based on the typical transaction costs and price variability associated with trading of different asset classes. In this setting, an asset type is classified by one parameter, T/σ, which is the ratio of the transaction cost rate to asset return variability. As the asset volatility increases over longer timescales, the parameter T/σ represents the effect of not only trading different asset classes but also trading over different time intervals.

Three scenarios were set up with the ratio of transaction cost to asset variability, T/σ, set to either, 0.1, 0.5 or 1.0. These three values represent typical values of T/σ for daily trading of equities, long government bond futures, and interest rate futures respectively, with transaction costs approximated from table 4.1 and volatility estimated using the standard deviation over historical data. The two trading rules, that exploit the "sign" or the "magnitude" of the predicted return, as defined in equations (4.16) and (4.17), are labelled by $k = 0$ and $k = 1$ respectively. The predicted returns are simulated with controllable levels of predictive correlation and prediction autocorrelation, using equation (4.3). The average profit for simulating each of the three scenarios over the range of the range of the two predictive characteristics is shown in figures 4.7, 4.8 and 4.9.

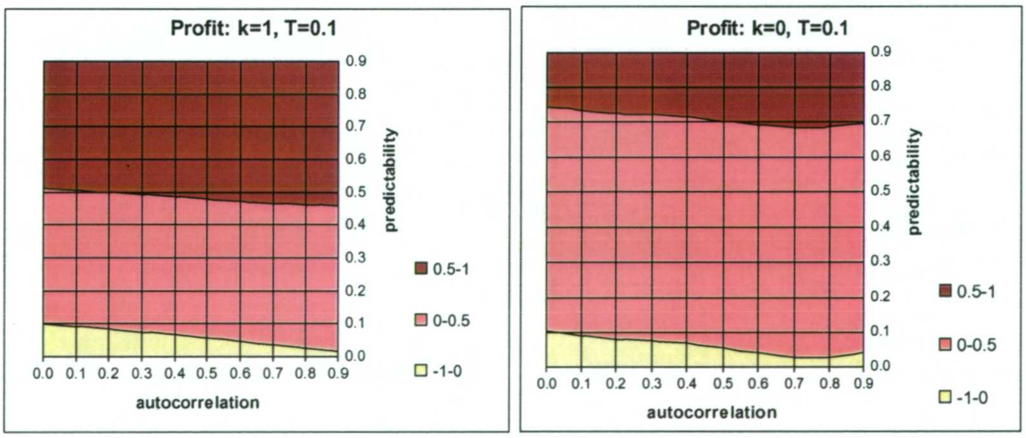


Figure 4.7 depicts the contour profile of average profit for transaction cost $T=0.1$ for two trading rules, where k =1 represents the "magnitude" rule (equation 4.17) and k=0 represents the "sign" rule (equation 4.16).

Figure 4.7 shows that the average profit increases as the prediction accuracy of the forecasting model increases. Positive profits are achieved for all forecasting models with prediction accuracy above 0.1. The horizontal contours show that the effect of autocorrelation is small. Increasing prediction autocorrelation by 0.1 is approximately equal to increasing prediction accuracy by 0.01. The magnitude rule slightly increases profits compared to the sign rule although the pattern is similar.

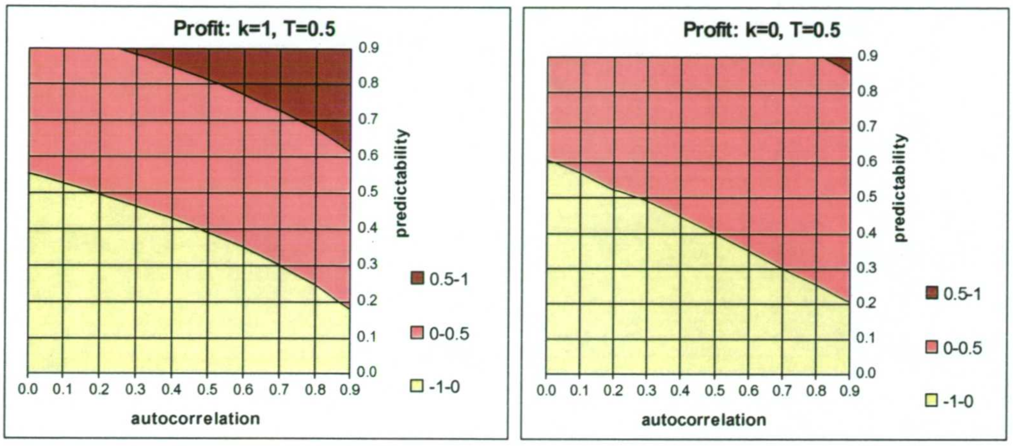


Figure 4.8 depicts the contour profile of average profit for transaction cost *T=0.5*, for the two trading rules, where k =1 represents the magnitude rule (equation 4.17) and k=0 represents the sign rule (equation 4.16).

Figure 4.8, shows the results for the higher transaction ratio. In this scenario a prediction accuracy of at least 0.2 is required to achieve a profit. The angled contours show that prediction autocorrelation is significant with both variables influencing profit. A prediction accuracy of approximately 0.6 is required to achieve a profit if the autocorrelation is zero, for both trading rules, compared to only 0.2 when the autocorrelation of 0.9. This shows that prediction autocorrelation is an important factor as relative transaction costs increase. This is because higher prediction autocorrelation has the effect of reducing the turnover the trading position thereby improving profitability. Again the magnitude rule marginally improves performance compared to the sign rule.

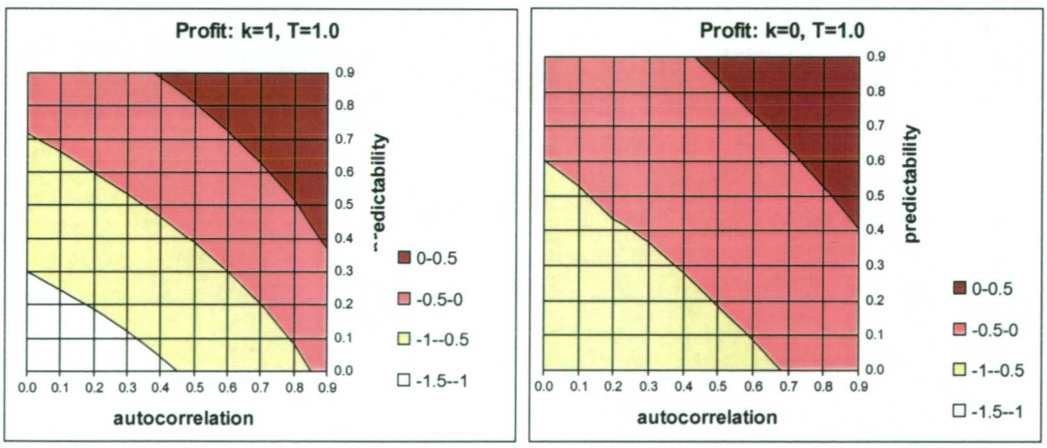


Figure 4.9 depicts the contour profile of average profit for transaction cost, $T=1.0$ and the two trading rules, where k =1 represents the magnitude rule (equation 4.17) and k=0 represents the sign rule (equation 4.16).

Figure 4.9 shows that for both of these trading rules prediction accuracy of at least 0.4 is required to achieve a profit. The contours show that profitability is affected almost equally by prediction autocorrelation and prediction accuracy. For instance, the expected average profit for autocorrelation of 0.6 and predictability of 0.3 is almost equal to autocorrelation of 0.3 and predictability of 0.6. These results show that, for this level of relative transaction costs, prediction autocorrelation is as important a factor as predictive correlation.

These three experiments serve to illustrate the interaction between the two simulated properties of the predicted returns, namely predictive correlation and prediction autocorrelation, two trading rules and the transaction costs, from the observed bid/ask spreads, for three different classes of assets based on trading over a daily time horizon. These results can also be extrapolated to the trading of particular asset classes over shorter or longer time horizons, where shorter intra-day time intervals will increase the relative transaction cost and time intervals greater than a day will effectively decrease relative transaction cost.

From these simulation experiments we have deduced that for asset classes will low transaction costs to price variability, predictive accuracy is the most important feature of the predicted returns and is the key factor to effect profitability. In this case, even low levels of predictive accuracy can be exploited to achieve significant profitability. However, for asset classes with relatively high transaction costs, the prediction smoothness of the predicted returns becomes an increasing important feature and has a significant effect on profitability. In this case, high levels of predictive correlation our required to achieve profitability (in some cases above 0.6) although this is offset by higher prediction autocorrelation. In the next section we extend our analysis of a trading system to examine the influence of trading costs and, in particular, consider the effect of market impact.

4.5 The Influence of Trading Costs

In this section we discuss the costs that can arise during trading in financial markets and devise some simple effects of market impact. We demonstrate, using an illustration, how market impact, in the form of a "liquidity threshold" can effect profitability. Under these trading conditions, we show how a trading system requires the development of a sequential decision model to take account of both the immediate and longer consequences of trading.

In investment finance, the standard assumptions of modern portfolio theory assume away the effect of market imperfections such as transaction costs, market impact, taxes and trading restrictions. Under these conditions the optimal portfolio can be considered "myopic" in that it does not depend on the longer-term consequences of trading. Typically market imperfections are considered as an aspect of the financial market's microstructure and so can be ignored when considering the "efficient markets" perspective, as these costs do not significantly effect longer-term, passive investment strategies. However these trading costs often have a significant effect on the performance of shorter-term, dynamic trading strategies.

In practice, trading opportunities must include all costs associated with transacting a trade to determine if the marginal cost of execution outweighs the potential profits, which may include:

- *Commissions* : direct payments to brokers acting as agents
- *Bid-ask spreads*: these are indirect payments to market makers, with the quoted *ask* price the higher price at which they are willing to sell, and the *bid* price the price at which they will buy.
- *Administrative costs*: these include the direct costs of confirming, documenting, reconciling and clearing trades.
- *Taxes*: these include stamp duty on equities and withholding tax on fixed income securities.
- *Market Impact*: additional costs that influence bid-ask spreads and arise from the relative size of the trade and the state of the market.

Trading costs are driven by the market microstructure, which is due to the organisation and control of the trading process in financial markets and is associated with trading through both market makers and order driven exchanges. In most well established markets these structures are designed so that investors can buy or sell significant quantities of securities quickly, anonymously and with relatively small market impact. Usually this liquidity is maintained by investors incurring a trading cost, usually in the form of the stable bid/ask spread on the price of a security. However in times of low trading volume or when trading relatively large quantities of assets, sufficient liquidity cannot be maintained cheaply by the market and so the trading of specific assets may have significant market impact. In these circumstances the very act of trading can move the price so that trading costs are proportionally more as the size of the trade increases (for more details see, Hausmann, Lo and MacKinlay, 1992). Typically, costs associated with market impact are determined by two types of factor:

- Trade information (percentage size of trade, market capitalisation of stocks)

- State of the market (volume, volatility)

Under these conditions, the minimisation of execution costs typically involves breaking down large trades into smaller blocks to spread trading over a number of time periods (Bertsimas and Lo, 1998). The optimal trading strategy is no longer myopic but now depends on optimising sequences of trades over a given time period. The optimisation of a trading strategy in the presence of market impact requires the development of a sequential decision model. In chapter 5, we develop a methodology for managing a trading strategy to exploit predictability, which takes account of execution costs.

4.5.1 Modelling Market Impact

The purpose of modelling market impact effects is to explain the wide variations in the realised transaction costs from simultaneously rebalancing portfolios of assets. Typically, the transaction cost incurred relative to the mid price covers half the bid-ask spread plus the compensation to the broker for taking on market impact risk, stock specific price risk and market risk. One simplified modelling approach, which we employ in this thesis, is to assume that transaction costs arising from market impact are only related to the size of the trade. Under this assumption we ignore the effects of market-wide trading volume and volatility. Thus, we simulate the market impact cost to the investor using a function dependent on the change in the trading position.

More specifically, let the investment return, r_t be determined by the trading position from a_t and the asset return Δy_t and some transaction cost function C which is due to some market impact effect, which is related to the size of the trade, and takes the form:

$$r_t = a_t \Delta y_t - C(a_t, a_{t-1}) \tag{4.20}$$

As an example, the transaction cost function may be defined as

$$C(a_t, a_{t-1}) = \begin{cases} 0 & \text{if } |a_t - a_{t-1}| \le \lambda \\ c & \text{if } |a_t - a_{t-1}| > \lambda \end{cases} \tag{4.21}$$

where λ is a tolerance parameter and c is some additional cost.

For this simplified transaction cost model, the cost of trading is significantly different from the proportional cost associated with stable bid-ask spreads and is motivated by the fact while small trades have negligible market impact, large trades cannot be executed without incurring extra additional costs. If we consider c to be large, we penalise moves more than λ sufficiently to consider it as a trading restriction around the movement of the existing trading position. For this trading system, trading opportunities are limited depending on the existing trading position and the parameter, λ, which can be considered a "liquidity threshold". Although this model is an oversimplification of practical transaction costs it is motivated by the fact that practical trading conditions often restrict trading on the basis of existing trading positions. An interesting effect of this trading restriction is that the trading environment is now path dependent. This has the effect that expected trading profits benefit from considering both the immediate and longer-term effects of trading. This effect is highlighted in the next section and used in chapter 5 to demonstrate the potential of a sequential decision model implemented using reinforcement learning.

In principle, more sophisticated market impact models can be developed which take the general form:

$$C = f(\sigma, v, S, \kappa) \tag{4.21}$$

where σ is the market volatility, v is the trading volume, S is the relative size of the trade and κ is the capitalisation/liquidity of the stocks in the trade.

The specific function form of equation (4.21) is then dependent on the financial instruments and the specific market in which they are traded. These transaction cost models are becoming increasingly important as practitioners address the significant market impact costs that can be associated with portfolio trading (e.g. Cook, 1999).

4.5.2 Illustration: The effect of a trading restriction

To demonstrate the concept of market imperfections we devised a simple illustration of the effects of a trading restriction on trading performance. The trading restriction represents an investment limit dependent on the current trading position.

Suppose we wish to control a simple dynamic trading system for a single risky asset. The trading position starts at zero and is restricted to three possible states, either -1,0 or +1, and may only increase or decrease by one unit in each time step. We know that in the next two time periods, there is a 55% probability that the stock price will increase by one unit in the first time step and then only a 10% probability of an increase in the second step, as illustrated in figure 1. If we assume there are no other trading costs and the objective is to maximise profits then we can calculate the optimal policy for the "myopic", two period and also an unrestricted trading view.

In the case of the unrestricted trading view, we know from standard portfolio theory that the "myopic" policy is optimal so the best trading position only requires knowledge of the expected trading return in the current period. Thus the optimal policy is to increase the trading position to +1 in the first step and decrease the position to –1 in the second. For unrestricted trading, the expected trading profit over the two periods is calculated from the price probabilities and the selected trading positions and has the value of 0.9 (i.e. 0.55*1 - 0.45*1 + 0.90*1 - 0.1*1 = 0.9).

In the case of restricted trading, the optimal policy for the "myopic" view is to increase trading position to +1 in the first step. However, due to trading restrictions, the position can only be decreased to 0 in the second step. The overall expected reward for the myopic view under restricted trading is only 0.1. In the optimal two period view the first trading decision considers expected rewards over both periods. Thus the trading position remains at zero in the first step and is decreased to -1 in the second, with an expected investment reward of 0.8. These three simple trading policies are illustrated in figure 4.11.

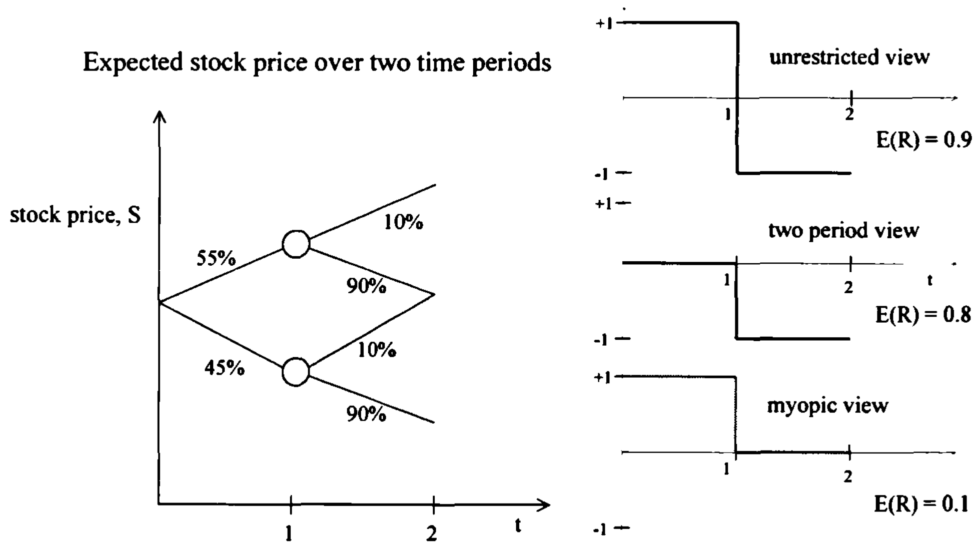


Figure 4.11. The left hand graph depicts the predicted stock movements over the two periods and the right hand graph shows the trading positions for the optimal unrestricted, restricted myopic and two period trading policies.

This simple illustration shows that multi-period or "forward looking" approaches can improve the overall trading performance compared to the trading strategy which is optimal in a myopic sense.

Furthermore, we can extend the illustration to the more general case where the predicted return is unknown in the second time period. In this case, the expected value of trading over the two time periods can be defined given the predicted return $\hat{y}_1$ and the trading position a_1 as follows:

$$V(\hat{y}_1, a_1) = \hat{y}_1 a_1 + E[\hat{y}_2 opt(a_2)] \tag{4.22}$$

where $opt(a_2)$ is the optimal myopic trading position in time period 2.

If we assume that the actual and predicted returns are described by equations (4.13) and (4.15) then equation (4.22) can be re-expressed as

$$V(\hat{y}_1, a_1) = \hat{y}_1 a_1 + \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-x^2/2} \beta x opt(a_2) dx \tag{4.23}$$

where β is the measure of prediction accuracy and x is a random variable which describes the deterministic component of a financial time series and follows a standard normal distribution.

If we impose limits on the trading position of (-1,1) we can assume that optimal myopic trading policy in time period 2, denoted by $opt(a_2)$, is defined as:

$$\begin{aligned} opt(a_2) &= \min(1, a_1 + \lambda) \quad & \text{if } x > 0 \\ &= \max(-1, a_1 - \lambda) \quad & \text{if } x < 0 \end{aligned} \tag{4.24}$$

where λ is the trading restriction.

If the trading position is initialised to zero and we assume that market impact imposes a trading restriction of λ then we can define the value of each trading position a_1 given a predicted return $\hat{y}_1$ as follows:

$$V(\hat{y}_1, a_1) = \hat{y}_1 a_1 + \frac{\beta}{\sqrt{2\pi}} \left[\int_{0}^{\infty} e^{-x^2/2} x \min(1, a_1 + \lambda) dx + \int_{-\infty}^{0} e^{-x^2/2} x \max(-1, a_1 - \lambda) dx \right] \tag{4.25}$$

Using equation (4.25) we can solve for the optimal trading strategy in time period 1 given a predicted return $\hat{y}_1$, the prediction accuracy of the forecasting model, β, and the trading restriction, λ. If we assume that the trading restriction is 1 unit and the predictive correlation equals 0.2, then the optimal trading policy in time period 1 is described in figure 4.12.

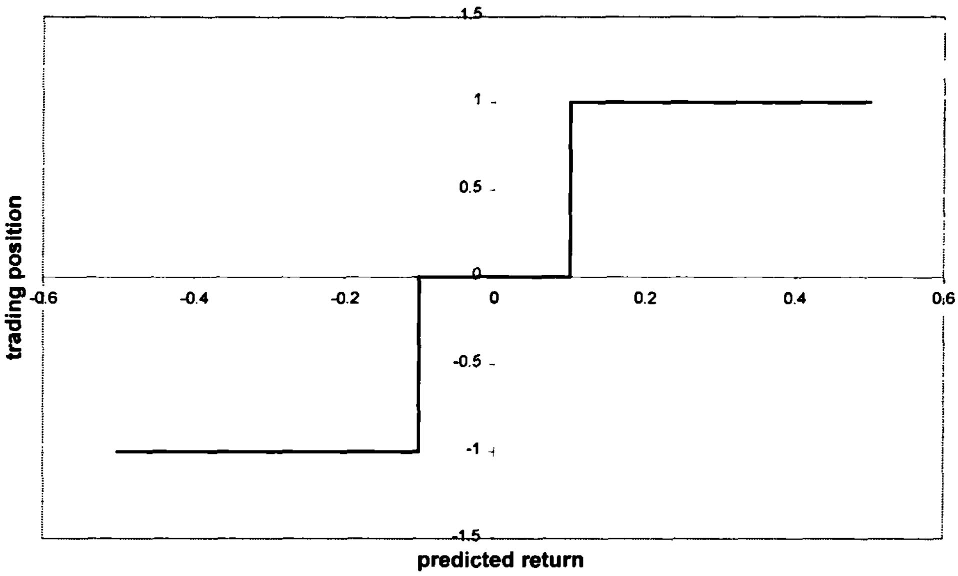


Figure 4.12 shows an example of the optimal trading strategy for two periods.

For this example, the trading position is maintained at zero, even if the predicted return has small positive or negative values (up to 0.12). Only if the predicted return is above (below) 0.12 does the trading strategy change the trading position to 1 (-1), which is the extreme of the trading restriction. In this case of a trading restriction, the optimal strategy is a trade-off between exploiting predictability in time period 1 and a holding a trading position which can exploit the potential predictability in period 2. A trading position of zero offers the best opportunity to exploit potential predictability in period 2.

These experiments serve to illustrate that the optimisation of a trading strategy requires information about not only the state of the trading position and the predicted return but also the prediction accuracy of the forecasting model and some approximation of the transaction costs, which may arise from market impact. In chapter 5 we develop computational modelling techniques to approximate the optimal trading strategy in order to exploit predictability captured by a forecasting model in the presence of variable transaction costs.

4.6 Trading Performance

In this section we provide a concise review of techniques for judging trading performance concentrating on the main issues involved in the optimisation and evaluation of performance and the economic value of trading systems that exploit predictability.

In general, the development of a trading system involves maximising some performance criteria, which describes the investment goal or objective. Typically, a trading goal is to maximise some objective function, such as profit, an economic utility function or some risk-adjusted measure of returns, which depends on the investment return-risk preferences. This is obviously different to the goal of optimising the forecasting model to capture predictability which typically minimises prediction error.

For dynamic trading systems, the obvious objective function for a risk insensitive investment strategy is to maximise profit. This may be described as additive profits, if the investment involves a fixed number of shares or contracts, or as multiplicative profits if the investment at each time period is a fixed fraction of the current accumulated wealth.

For risk sensitive investment strategies, trading performance needs to be modified to take account of the risk involved in holding assets. Typically, investment strategies are risk adverse, which means that some potential gains are forfeit in order to lower investment risk. Modern portfolio theory provides a range of utility functions that capture various degrees of risk sensitivity, which typically involve estimating the expected portfolio variance. A commonly used performance metric is Risk Adjusted Return (*RAR*) which is based upon mean-variance analysis from modern portfolio theory, and is defined as

$$RAR = N\left(\bar{r} - \frac{\sigma_r^2}{T}\right) \tag{4.26}$$

where N is the number of trading periods per year, $\bar{r}$ and σ_r^2 are the mean and variance of the trading returns and T is the risk preference parameter.

Another popular measure is the Sharpe Ratio (Sharpe, 1966; Sharpe, 1994), denoted by *SR*, which is defined as the ratio of average annualised excess returns to annualised standard deviation of returns:

$$SR = \sqrt{N}\left(\frac{\bar{r}}{\sigma_r}\right) \tag{4.27}$$

where N is the number of trading periods per year and $\bar{r}$ and σ_r are the mean and standard deviation of the trading returns.

An advantage of the Sharpe Ratio over *RAR* is that the risk preference of the investor does not need to be specified. The Sharpe Ratio can also be used to indicate the appropriate level of leverage that can be applied to a trading strategy. Another well-used performance metric is the

Sterling ratio, defined as the ratio of the average excess return to the maximum cumulative drawdown.

Risk adjusted performance metrics can also be used for optimising trading systems using both batch and on-line optimisation techniques, as discussed by (Moody, 1996; Choey and Weigend, 1996; Towers and Burgess, 1997). A survey of performance metrics for trading systems is presented in Refenes (1995) and Moody (1997).

The selection of the correct objective function can be important to the optimisation of a trading system and may in practice be described by kinked or complex utility functions that take account of performance related management fees or contract termination criteria. We consider that these issues are specific to individual practitioners and so outside the scope of this thesis. However we consider that our methodology is capable of being adapted to optimise trading strategies for arbitrary performance measures. In this thesis we develop methods for trading strategies that can optimise both risk neutral and risk adverse performance metrics. In particular, we focus on the Sharpe Ratio as it is considered the "industry wide standard" of performance measurement.

4.7 Summary

In this chapter we have investigated the components of a trading system which is constructed to exploit the predictability of a forecasting model of asset returns. We have developed a formal definition of a trading strategy, based on a Markovian decision framework, and compared two conceptual approaches to trading system design. The first uses only a single modelling stage to optimise the trading strategy and the second use two modelling stages, one for the two tasks of forecasting and decision-making. We have discussed the advantages and disadvantages of the two designs and described the potential "forecasting bottleneck" of the two-stage approach. We discussed how this provides motivation for the development of a joint optimisation procedure which can optimise over both modelling stages. Furthermore, we have discussed the two designs from the statistical perspective of generalisation by considering the number of degrees of freedom and effect of irrelevant input variables for comparable models. This analysis was shown to provide further justification and motivation for the two stage modelling approach.

Next we developed a synthetic trading system comprised of the data generating process to emulated the output from a generic forecasting model with controllable prediction characteristics and a decision phase with two simple trading rules that converted predicted returns into trading positions. Simulation studies were conducted to assess the effect on profitability of the two prediction characteristics, namely predictive accuracy and prediction smoothness, under different trading conditions. These experiments provided an illustration of how the two predictive characteristics can influence trading performance depending on the transaction cost and the particular asset class. The influences of practical trading costs have been described and possible transaction cost functions for market impact discussed. A simple example of a "nonlinear" transaction cost has been developed based on a liquidity threshold, which has the effect of restricting the change in the trading position. We used a simple illustration to show how the optimal trading strategy in the presence of a restriction is no longer myopic but depends on optimising a sequence of trades. Finally we have discussed the measurement of "profitability" using risk adjusted performance measures.

4.8 Methodological Overview

On the basis of this investigation, the rest of this thesis is devoted to the development and evaluation of a joint optimisation methodology for a trading system comprised of two modelling stages, one for forecasting asset returns and the second for optimising the trading strategy, as described conceptually in figure 4.13.

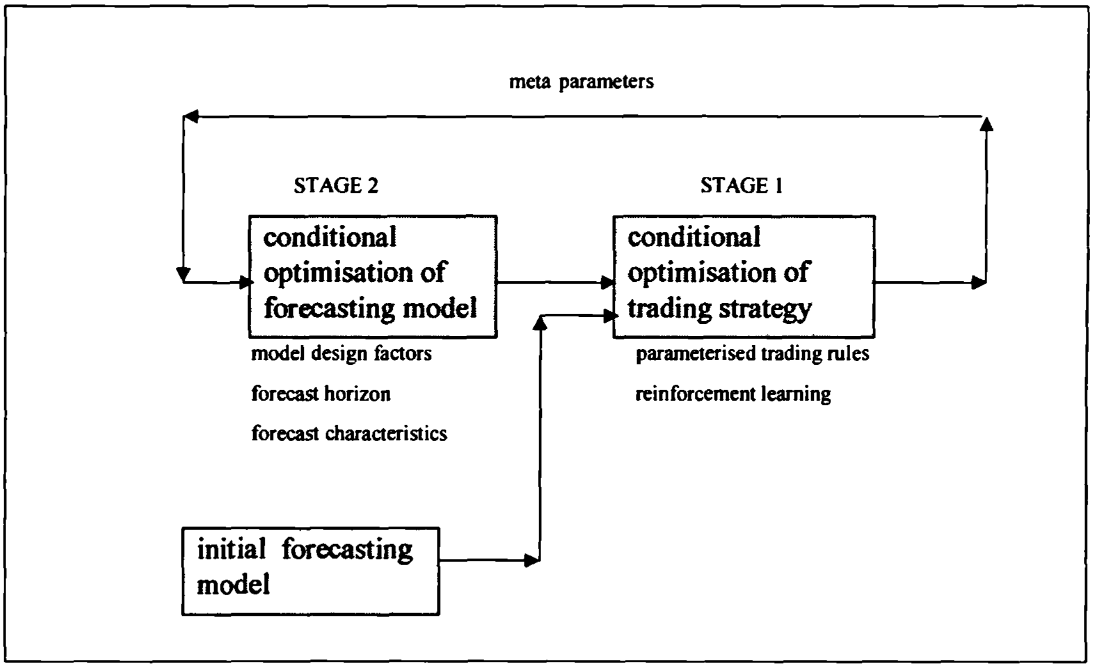


Figure 4.13 depicts the joint optimisation methodology for a trading system comprised of a forecasting model and a decision model.

Figure 4.13 depicts the process of optimising a trading system comprised of two modelling stages for the two separate tasks of forecasting and trading. The optimisation process is initialised by some preliminary forecasting model, which is constructed on the basis of identifying some predictable component of asset price dynamics.

In chapter 5 we develop the first stage of our methodology which involves the optimisation of the trading strategy given the predicted returns from a forecasting model. We develop two modelling approaches depending on the whether transaction costs are fixed (i.e. stable bid-ask spreads) or variable (i.e. market impact effects). When trading in the presence of stable bid-ask spreads, structured decision modelling techniques are shown to be most suitable. We develop parameterised trading rules that are able to combine *a priori* knowledge with a parameterised model in order to approximate the optimal trading strategy (Towers, 1998,1999). This modelling approach is a generalisation of the simple two trading rules described in this chapter with the added abilities of optimising the response of the trading position to the predicted return while also taking account of transaction costs in a parsimonious manner. For more complex trading environments, which include costs arising from market impact, less structured decision models are required that allow for greater flexibility in the functional form of the trading strategy. This is achieved by developing a model of the trading strategy using reinforcement learning. This method has been successfully applied to complex sequential decision tasks to other fields that are intractable for dynamic programming methods. We develop an enhanced Q-learning algorithm which is tailored to optimise trading strategies for forecasting models in the presence of general trading costs (Towers, 2000).

In chapter 6 we develop the second stage of our methodology which involves the conditional optimisation of a forecasting model. In this stage the economic value of the forecasting model is considered conditional on a given trading strategy. This approach is developed from the investigation in this chapter, in which we illustrated that the profitability of a trading strategy can be affected by different predictive characteristics. Thus, we develop a methodology for optimising certain design factors of a forecasting model which may influence trading performance. Specifically, we consider an optimisation criterion, consisting of two predictive characteristics, predictive accuracy and prediction smoothness. We develop a technique for controlling the trade-off between these two characteristics using a *meta* parameter (Towers, 2000). This has the effect of optimising the forecasting model conditioned on a given trading strategy. Other design factors are considered including the forecast horizon and the forecast object.

In chapter 7 we combine the two separate modelling stages to perform a joint optimisation over both the forecasting model and the decision model in order to "globally" optimise trading performance. This process involves repeatedly alternating between the two modelling stages until the two models are optimal with respect to each other and so jointly maximise trading performance. This approach is developed from the investigation in this chapter, which discussed the potential "forecasting bottleneck" that results from the two stage modelling design. In chapter 8 we evaluate our joint optimisation methodology in the context of statistical arbitrage trading of predictive models. We conduct extensive empirical evaluations of realistic statistical mispricings identified within the UK equity market (FTSE 100).

5 Conditional optimisation of a trading strategy

In chapter 4 we investigated the factors that may effect the profitability of a trading strategy which exploits some predictable component of asset price dynamics. In addition, we proposed a methodology to optimise performance based on two modelling stages, one for each of the separate tasks of decision-making and forecasting. In this chapter we develop the first stage of our methodology for the task of decision-making. This involves the conditional optimisation of a trading strategy, given predicted asset returns from a forecasting model, in the presence of general trading costs, which may include market impact. It has been shown, in a number of studies (e.g. Bertismas and Lo, 1998), that market impact can have an important effect on the total cost of trading.

In section 5.1 we provide an overview of how we apply decision-modelling techniques to optimise a trading strategy for efficiently exploiting predicted asset returns. We highlight the progress from simple heuristic trading rules through to advanced decision models, such as parameterised trading rules and reinforcement learning. In section 5.2 we develop a parameterised class of trading rules for predicted return signals and investigate the optimisation of trading strategies for synthetic forecasts with controllable characteristics. We apply smoothing techniques to the parameterised trading rule to optimise trading in the presence of fixed transaction costs. We simulate the effects of different levels of transaction cost and forecast model characteristics. Performance improvements are explained in terms of a novel form of the bias-variance trade-off, which balances the performance gains from exploiting the forecasting model against minimising the transaction costs associated with trading. In section 5.3 we develop a reinforcement learning system to optimise a trading strategy in the presence of general trading costs, which may include market impact. Specifically, a Q-learning algorithm is designed to exploit the properties of the financial domain. In particular, the standard RL algorithm is modified to take account of the partial independence between asset prices and trading decisions to learn from a region of possible states in each time period. This special quality is due to the inherent difference between the state of the market and decision-making in financial markets. This is extended to parameterised reinforcement learning, using a neural network, in order to generalise to a continuous state-action space. Simulation experiments are conducted to test performance of our RL algorithm against the optimal myopic trading strategy and also standard RL implementations. In section 5.4 we summarise the advantages and disadvantages of each method.

5.1 Overview

In this section we discuss the application of the decision modelling techniques, described in chapter 3, to the problem of optimising a trading strategy given some predicted returns. This is the first stage of our methodology and is concerned with the construction of a decision model to optimise a trading strategy given predicted returns from a forecasting model. The trading strategy is represented by a stationary policy which is a mapping from the state information s, and the predicted return, $\hat{y}_{t+1}$ to some trading position, a, and takes the general form:

$$a_t = f(s_t, \hat{y}_{t+1}) \tag{5.1}$$

where f is some function of the predicted return and the state information, which may include variables describing past actions, previous predicted returns and other arbitrary variables, as described in equation (4.1).

For practical dynamic trading systems, where trading decisions are interdependent, the trading policy cannot be determined analytically except under the most idealised conditions. Thus, optimisation of the trading policy requires the development of a sequential decision model to approximate the optimal trading strategy in order to maximise expected performance. In this chapter we develop models using decision modelling techniques, namely, parameterised decision rules and reinforcement learning. These two learning methods were first discussed in chapter 3 where they are described and compared with related decision modelling techniques.

To construct a decision model for trading requires that some underlying assumptions be made regarding both the state and the nature of the trading environment. In the simplest case, where the trading environment yields negligible trading costs, the state, as described in equation (5.1), is redundant and the mapping consists solely of the predicted return. Under these trading conditions, the policy is just a one-dimensional mapping from the predicted return to the trading position. We can assume, for an unbiased forecasting model, that a positive predicted return should indicate a positive trading position and, conversely, a negative predicted return should give a negative or an under weight trading position. Based on this assumption, we can impose this information as *a priori* structure on the functional form of the policy. This intuition is implicit in the definition of the two trading rules, which were discussed in chapter 4, where the trading position is either linearly proportional to the value of the predicted return or simply a fixed quantity depending on the sign of the predicted return.

We can use this a priori information to develop a structured decision modelling technique based on a class of parameterised decision rules, as described in chapter 3. In section 5.2, we build upon the underlying assumptions of two simple trading rules to develop a class of parameterised trading rules which allows optimisation of the function which describes the mapping from predicted returns to trading positions. A parsimonious modelling framework is constructed to efficiently approximate the optimal trading policy using only two decision rule parameters. These control the magnitude and sensitivity of the trading position with respect to the predicted return. This approach, based on a class of parameterised decision rules, is analogous to imposing a "correct" trading bias given *a priori* knowledge with minimum model variance imposed by the parameterisation of the decision model.

In trading environments with significant trading costs, however, the state should consist of other variables in addition to the predicted return. In these circumstances parameterised trading rules based solely on the predicted returns will not adequately take into account the additional factors and so potentially lead to sub-optimal decision making. However, before constructing a more general sequential decision model, we extend the parameterised trading rule approach to consider trading with significant, fixed transaction costs. These conditions are common for many practitioners actively trading relatively small funds in well-established financial markets with high levels of market liquidity. In this case, a parameterised trading rule can be adapted to take into account the effect of transaction costs on the optimal trading strategy. This is achieved by applying time series smoothing techniques to the parameterised trading rule, which has the effect of smoothing the trading position through time. The level of smoothing is determined by a parameter, which operates in conjunction with the two trading rule parameters. Conceptually, this approach performs a trade-off between bias and variance factors thereby reducing the costs of changing the trading position at the expense of not fully exploiting the predictive information. We demonstrate, using simulation, how this approach can improve trading performance in the presence of fixed transaction costs.

In practical trading environments, however, we often cannot simply assume that trading costs are fixed in that they are derived from a stable bid-ask spread. For example, costs arising from market impact depend both on the state of the market and the state of the portfolio. These effects cannot be adequately modelled using structured decision modelling techniques, such as parameterised decision rules, so we develop a sequential decision model which can optimise a more general form of trading policy. The conventional modelling approach to solve the majority of sequential decision tasks under uncertainty is stochastic dynamic programming, as discussed in section 3.2. However, in the case of a trading system for predicted returns, there is not a complete model of the underlying dynamics of the system. This is evident by the fact that the state does not contain the predicted returns or other state variables at each future time step (i.e. time steps beyond the current decision epoch). The lack of state and/or state transition probabilities hinders the utilisation of dynamic programming methods to optimise a trading strategy. To solve this issue we resort to a method, widely used within the machine learning community and successfully applied to a number of complex sequential decision tasks, which is commonly referred to as reinforcement learning. This approach has the advantage over dynamic programming methods of relaxing the assumption of a complete model of the system and instead learning the value function from trading experience.

In section (5.3), we implement a reinforcement learning algorithm for the task of optimising a trading strategy for predicted returns. In this learning method no *a priori* structure is applied to the policy function, described in equation (5.1). Instead, a value function stores the salient information related to trading experience from which the trading policy is derived. At convergence, the value function can be used to derive an approximation to the optimal trading strategy. A standard reinforcement learning algorithm, Q-learning (Watkins, 1989), is modified in order to improve the speed of learning, by learning from an entire region of action-states rather than a single action. We refer to this new approach as multi-action learning. Furthermore, we develop a parameterised RL, using a neural network, to take account of continuous action-state spaces typically found in trading systems. In principle, function approximation techniques, such as neural networks, reduce model variance by imposing a smoothing bias to the learning process and so improve value function approximation in high dimensional or continuous spaces.

It is worth noting that reinforcement learning has a number of potential disadvantages or risks compared to the simpler but less rigorous approach of applying parameterised trading rules. For instance, the "correct" trading bias imposed by the parameterised decision rule approach is not included in the RL model. Furthermore, the greater modelling flexibility required for the RL model induces higher model variance, which may result in an inferior approximation to the optimal trading strategy. In section 5.4 we summarise the advantages and disadvantages of each modelling technique in terms of three attributes: flexible, conditional and adaptive and discuss which methods are most suitable for particular trading situations.

5.2 Parameterised trading rules and path dependency

In this section we develop a parameterised class of trading rules that approximates the optimal policy for trading predicted returns from a forecasting model. The approach is based on the assumption that only the necessary state information is the predicted return. The goal is to constrain the mapping from state to trading position by using a family of decision rules which are determined through a small set of rule parameters. Optimisation of the decision rule parameters then provides a means of approximating the optimal trading policy and so maximising expected trading performance.

In this case, the assumption that the trading position is only influenced by the predicted return allows us to impose some *a priori* knowledge about the possible functional form of the relationship. Firstly, we assume that positive predicted returns will lead to positive trading positions, and, conversely, negative predictions will lead to negative trading positions. Intuitively, we suppose that it does not make sense to sell an asset (i.e. move to a negative weighting or trading position) if the asset return is expected to outperform. Secondly, we assume that the trading position is a monotonically increasing function of the predicted return (i.e. higher predictions lead to larger trading positions). We further assume that the parameterised trading rule encompasses the two fixed trading rules, discussed in chapter 4, in which the trading position is either linearly proportional to the predicted return or simply a fixed magnitude depending on the sign of the predicted return. Finally, for simplicity, we assume that there are no limits to the maximum possible trading position.

Given these assumptions, we devise a parameterised class of trading rules that describes the functional form of the trading policy, in which the trading position a_t for a predicted return, $\Delta\hat{y}_{t+1}$ is given by

$$a_t = m\left|\Delta\hat{y}_{t+1}\right|^k sign(\Delta\hat{y}_{t+1}) \tag{5.2}$$

where k and m are the two decision rule parameters with the constraints, $k \geq 0$ and $m > 0$.[4]

In equation 5.2, the decision rule parameters, k and m control the shape and magnitude of the function defining the trading policy respectively. Examples from this parameterised decision rule are shown in figure 5.1.

[4] For $\boxtimes_t = 0$ and $k = 0$: assume $0^0 = 0$ on the basis of L'Hopital's theorem.

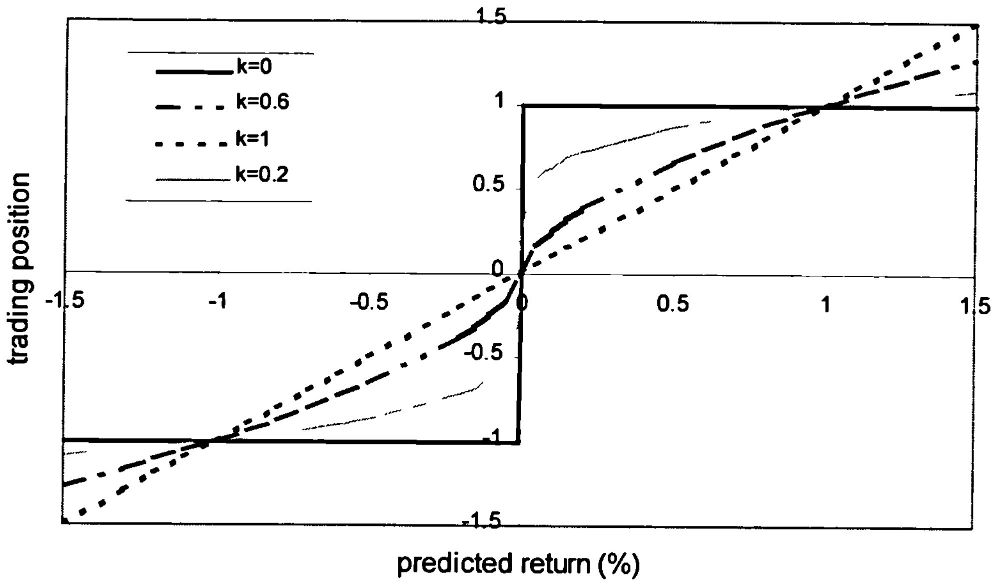


Figure 5.1 shows four examples of trading rules derived from the parameterised decision rule by varying the value of k with m fixed at unity.

When the parameter k equals one, the trading rule defines the trading position to be linearly proportional to the predicted return. When the parameter k equals zero, the trading rule defines a step function where the trading position is a fixed amount depending on the sign of the predicted return. Other positive values of parameters k and m give a wide range of non-linear decision functions while still imposing the constraints that the sign of the trading position is equal to the sign of the predicted return and that a higher return gives rise to at least as large a trading position.

This modelling technique uses only two decision rule parameters to describe an infinite set of possible trading policies. Optimisation of the decision rule parameters is accomplished by maximising some trading performance criterion over a sample data set to provide an efficient approximation to the optimal trading policy. Conceptually, the use of *a priori* knowledge to restrict the trading policy is analogous to imposing a "correct" trading bias while the two parameters ensure a parsimonious model with minimum model variance.

Given this parameterised trading rule, we wish to compute the expected trading returns distribution for forecasts with a certain degree of prediction accuracy. However, even for this relatively simple trading system, the expected mean and variance of the return distribution cannot be solved analytically so we need to resort to simulation studies to investigate performance attributes. This is achieved by using the simulated trading environment described in chapter 4. For illustrative purposes, we consider three examples of the two decision rule parameters and approximate the expected investment return and variance for forecasts with varying levels of prediction accuracy.

5.2.1 Simulation experiments for a class of parameterised trading rules

In this section we use simulation experiments to investigate the effect of different trading rule parameters using forecasting models on investment performance with different levels of prediction accuracy, as defined in equations (4.13) and (4.15). The predicted and actual asset returns are simulated for 4000 time periods using a data generating process with controllable characteristics for prediction accuracy. For these experiments any trading costs are assumed to have a negligible effect on the trading performance and so are ignored. The performance of different trading rules is compared by setting the rule parameter m to be a normalisation factor. The parameter, m is then equivalent to an average leverage factor with no upper limit. Figure 5.2 shows the first two moments of the expected trading performance distribution for three values of k.

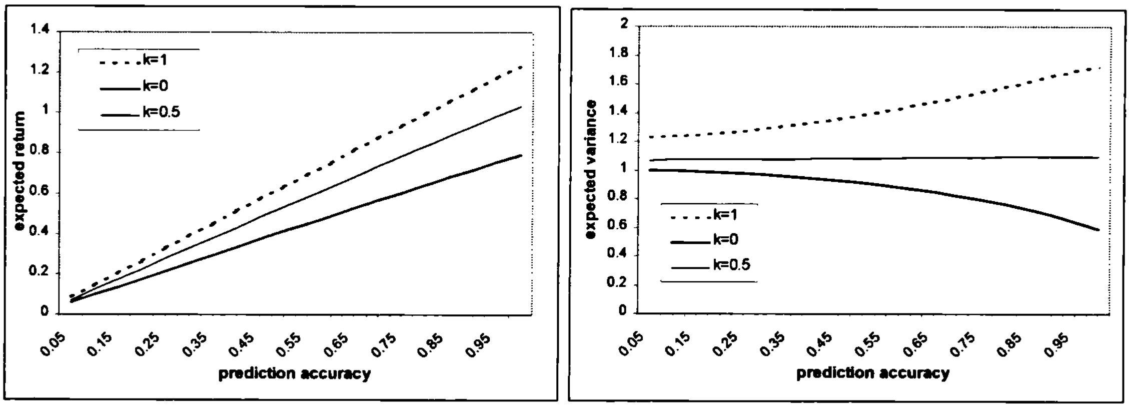


Figure 5.2 shows the mean and variance of the expected investment returns for three values of the decision parameter, k, with varying prediction accuracy.

Figure 5.2 indicates that the investment return distribution is sensitive to both the prediction accuracy of the forecasting model and the choice of the decision rule parameter, k. The left hand graph shows that increasing the prediction accuracy of the forecasting model increases expected trading return. However, the difference between the three lines clearly demonstrates the sensitivity of expected return to the choice of the trading rule. The expected return for the sign rule ($k = 0$) increases at a slower rate than for the other two rules ($k = 0.5$ or 1). This is because the sign rule only exploits the sign of the predicted return while rules with non-zero values of k exploit, to some degree, both the sign and the magnitude of the predicted return. This difference grows for higher levels of prediction accuracy as the information content of the predicted returns increases.

The right hand graph shows how the expected trading variance is also dependent on the selected trading rule. Interestingly, the sign rule ($k=0$) has a lower investment return variance while the other two trading rules increase variance for higher prediction accuracy. The sign rule reduces the variance of the trading returns with higher prediction accuracy as there is a lower probability of negative trading returns. The proportional rule ($k=1$) increases variance with higher prediction accuracy as the correlation increases between the trading position and the actual asset return, which, when multiplied together, produces higher variance.

From this analysis it is not clear which rule provides the better trading strategy. The proportional rule improves trading return but at the expense of higher variance compared to the sign rule. To investigate the optimisation of the trading rule, we set the trading performance criterion to be the ratio of the mean trading return to the standard deviation of the returns. This is a scaled measure of risk adjusted return based on the Sharpe Ratio. This metric is able to measure the risk-return trade-off and also has the advantage of treating the trading rule parameter, m, as a common multiplying factor of both expected return and standard deviation so that the trading performance is only influenced by the trading rule parameter, k. The trading rule is optimised to maximise the ratio (mean return divided by standard deviation) for different levels of prediction accuracy to investigate how the optimal value of the parameter, k varies with prediction accuracy, as shown in figure 5.3.

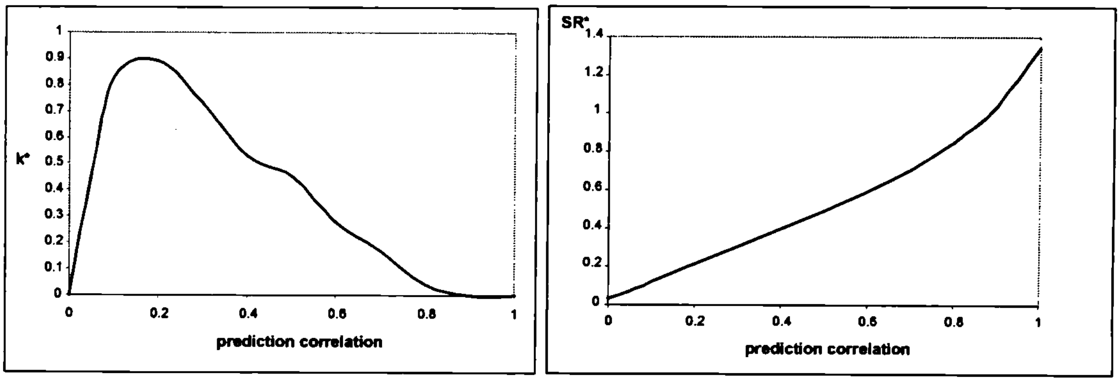


Figure. 5.3 shows the optimal value of the trading rule parameter k^* (left), and the optimal performance ratio SR* (right) for different levels of prediction accuracy.

The left-hand side graph of Figure 5.3 shows how the optimal trading rule parameter, k, varies with the prediction accuracy of the model. These experiments clearly demonstrate the sensitivity of optimal trading rule parameter, k to the prediction accuracy of the forecasting model. The analysis shows that, for this risk adjusted performance measure, a naïve trading rule (i.e. $k=0$), is only optimal when there is either zero or total predictability (i.e. prediction correlation = 0 or 1). The optimal value of k reaches 0.9 for a prediction accuracy of approximately 0.15. The right-hand side graph shows how this optimal performance ratio increases at approximately the same rate as prediction accuracy. It is worth noting that empirical research for forecasting models of asset returns shows that out-of-sample prediction correlation is low, typically between 5% and 20% (Lo and MacKinlay, 1998; Burgess 1999). This would indicate that the optimal range of k typically would be between 0.45 and 0.9.

These simulation experiments indicate that optimisation of the trading rule parameter k for a forecasting model with a consistent level of prediction accuracy may lead to improved trading performance. We conclude that the class of parameterised trading rules which we have developed is an effective method of optimising trading performance when transaction costs are not significant. In the next subsection we consider trading with significant transaction costs with a fixed rate.

5.2.2 "Path dependent" trading rules

In the previous analysis we assumed that trading costs are negligible and so trading is solely a function of the predicted return. In environments with significant transaction costs, however, a penalty is incurred for trading (i.e. when the trading position is changed). In this case performance is, in general, dependent on the sequence of trading positions or the "path" of the trading position as well as the predicted return. The smoother the trading position the smaller the impact of transaction costs on trading performance.

In this section we extend the parameterised trading rule, described in equation (5.2), to encompass "path dependent" trading rules which optimise trading performance in the presence of fixed transaction costs. The purpose of these rules is to allow the trading position to be smoothed out during trading by means of an additional trading rule parameter. We devise three different trading rules, each with a different form of smoothing and test the performance against the original trading rule, described in equation (5.2). The purpose of the path dependent trading rules is to smooth the trading position by including an additional parameter to reflect the most recent previous trading positions.

The first path dependent trading rule $a_t{*}^{1}$, is based on simple exponential smoothing of the trading position, a_t, given in equation (5.2), and is defined as

$$a_t{*}^{1} = (1-\theta)\, a_t + \theta\, a_{t-1}{*}^{1} \qquad (5.3)$$

where θ is a decay rate parameter with constraint, $0 \leq \theta \leq 1$. For example, if θ equals zero, no smoothing is applied to the trading position but if θ equals 1, the trading position reamins constant at the initial starting position.

The second rule, $a_t{*}^2$, is based on a simple moving average of the trading position, a_t, and is defined as

$$a_t^{*2} = \frac{1}{h}\sum_{j=0}^{h-1} a_{t-j} \tag{5.4}$$

where h is the rolling window parameter controlling the number of past observations with constraint $h \geq 1$. For example, if $h=1$, no smoothing is applied to the trading position and increasing h produces a higher degree of smoothing.

The third rule, $a_t{*}^3$, is a "noise tolerant" approach which attempts to decrease overall trading costs by reducing the number of relatively "small" changes in the trading position, and operates by only trading if the proposed change is significantly large. This rule is defined as

$$\begin{aligned} a_t{*}^3 &= a_t \quad && if\, |a_t - a_{t-1}{*}^3| > \lambda \\ &= a_{t-1}{*}^3 && otherwise \end{aligned} \tag{5.5}$$

where λ is the tolerance parameter with constraint $\lambda \geq 0$. For example, if $\lambda = 0$, no smoothing is applied to the trading position and increasing λ increases smoothness.

The effect of these conditional trading rules can be illustrated by using the synthetic trading system to generate predicted returns and the associated trading positions, as shown in figure 5.4.

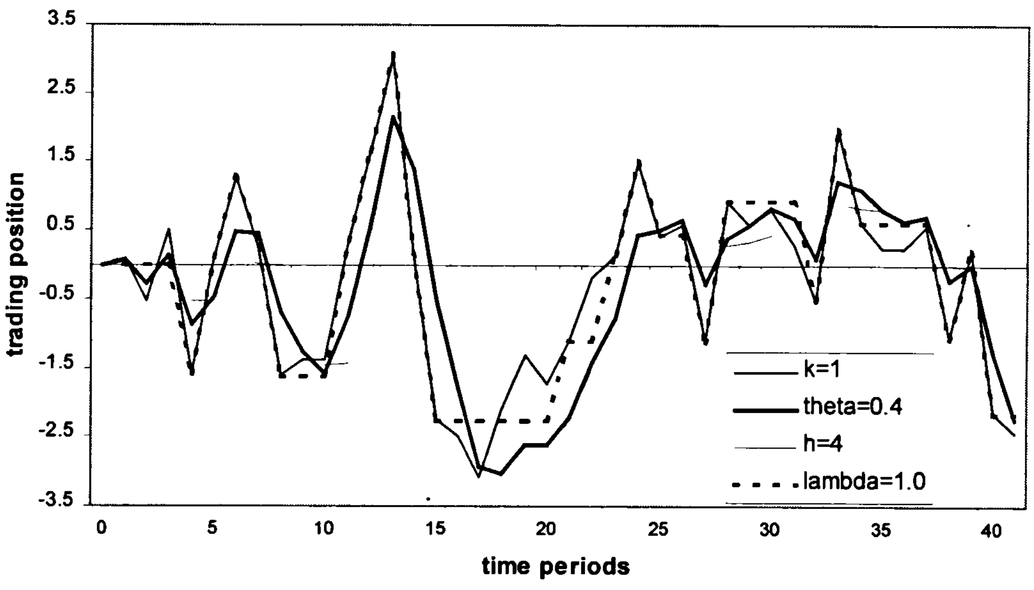


Figure 5.4 shows the trading positions of three examples of the path dependent trading rules

Figure 5.4 shows the simulated trading position for examples of the three path dependent trading rules over a sample of 40 time periods. The predicted returns are generated for a forecasting model with a prediction accuracy of 0.25 and a prediction autocorrelation of 0.5 using the simulation environment described in chapter 4. For simplicity, the k parameter, defined in equation (5.2), for all decision rules is set to one and m set such that the amplitude of the average trading position is normalised to one. For illustrative purposes, the exponential trading rule, defined in equation (5.3), is set with parameter, $\theta = 0.4$, the moving average trading rule, defined in equation (5.4), is set with parameter $h = 4$, and the heuristic trading rule, defined in equation (5.5), is set with parameter $\lambda = 1$. The graph illustrates the different smoothing effects of the three conditional trading rules. Next we assess the impact of the three "path dependent" trading rules for different levels of transaction costs and different characteristics of the predicted returns.

5.2.3 Simulation Experiments for "path dependent" trading rules

In this subsection we conduct simulation experiments to illustrate the performance effects of the "path dependent" trading rules compared to the initial class of parameterised trading rules, specified in equation (5.2). This synthetic trading system has parameters controlling the characteristics of predicted returns (i.e. predictive correlation and prediction autocorrelation), the parameterised trading rules and the level of transaction costs for a particular asset class. For investigative purposes, the parameters describing the transaction costs and parameterised trading rules are set to representative values in order to examine the effects of different predictive characteristics. This is achieved by setting the trading rule parameter, $k = 1$ and m equal to a normalising constant. The three remaining smoothing parameters of the path dependent trading rules are set to $\theta = 0.4$, $h = 4$, and $\lambda = 1$ respectively. This also enables us to compare trading profits between the three path dependent trading rules. For comparative purposes, three scenarios are investigated based on the transaction cost and price variability associated with trading different asset classes, which were specified in section 4.4.1, with transaction cost ratio, $T = 0.1$, 05 and 1.0 respectively.

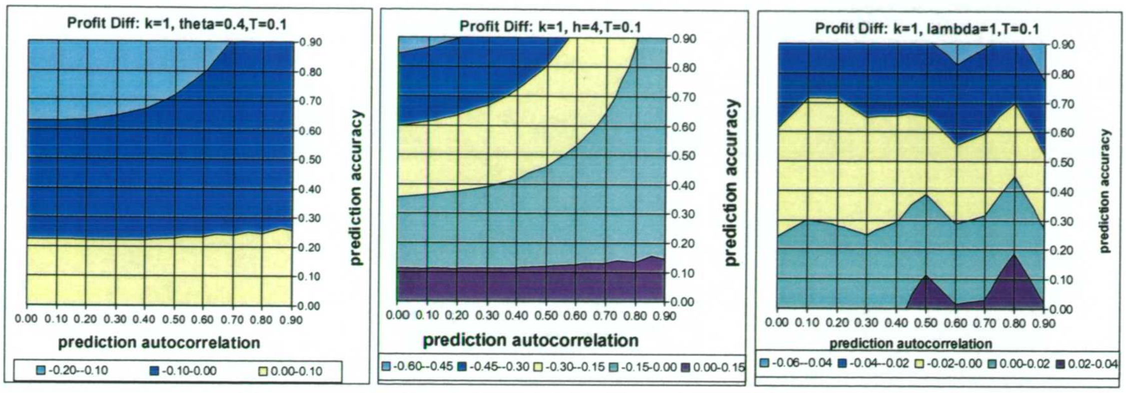


Figure 5.5 depicts the contour profiles for changes in average profit for the three "path dependent" trading rules for a transaction cost ratio of $T = 0.1$.

Figure 5.5 shows the profit difference for the three path dependent rules, which were described in equations (5.3), (5.4) and (5.5), compared to the parameterised trading rule *(k=1)*, given in equation (5.2*)*, for trading with a transaction cost ratio of $T = 0.1$. The first plot is constructed using the exponential moving average rule, the second plot uses the simple moving average rule and the third uses the adaptive step rule. The contours on all three graphs indicate the presence of both positive and negative regions. The negative regions are found for high levels of predictive correlation (i.e. greater than 0.4) while positive regions inhabit the lower levels of predictive correlation. The negative profit regions indicate that the path dependent trading rules can decrease profits compared to the original trading rule. This is due to smoothing having the dual effect of reducing overall transaction costs whilst inhabiting the exploitation of any predictive ability. Thus, in the case of high predictive correlation and low costs, smoothing actually has a detrimental effect on profits, as better exploitation outweighs the benefit of reducing overall costs.

In this scenario with low transaction costs, prediction autocorrelation has a marginal effect compared to predictive correlation, although the angled contours of plot 2 show that smoothing adversely affects performance for high predictive correlation and low prediction autocorrelation. This can be explained by the fact that smoothing has most impact on the trading position for predictions with low prediction autocorrelation, which thereby reduces the exploitation potential of high predictive correlation returns.

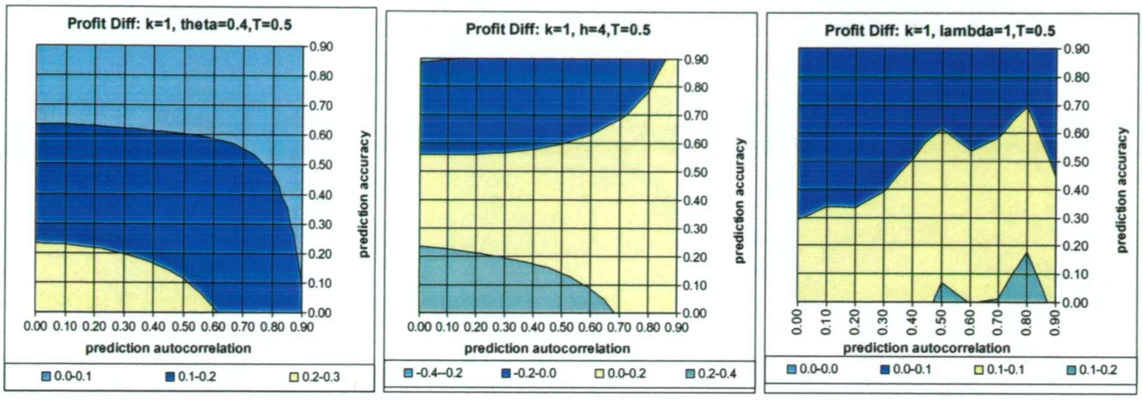


Figure 5.6. depicts the contour profiles for changes in average profit for the three path dependent trading rules for a transaction cost ratio, T=0.5.

Figure 5.6 shows the average profit difference for each of the path dependent trading rules compared to the original trading rule (k =1) with a higher transaction cost ratio of 0.5. For all three graphs, the path dependent rules have larger positive profit regions compared to the lower transaction costs shown in figure 5.5. This indicates that, in general, smoothing the trading position has most benefit for trading with higher transaction costs. In the first and third plots, for the exponential and step trading rules respectively, the positive contours indicate increased profits for all levels of predictive correlation and prediction autocorrelation. The second plot, for the simple smoothing model, has negative contours only for a small region with high levels of predictive correlation (i.e. above 0.55).

In plots one and two, the angled contours show that performance is also improved for low levels of prediction autocorrelation and predictive correlation. This is because costs saved through smoothing the trading position outweigh the profits which could otherwise have been generated by exploiting predictive ability. However, the negative region in the second plot indicates that smoothing can still result in worse performance for predictions with high levels of predictive correlation and low prediction autocorrelation. This is because, even at this higher level of transaction costs, smoothing the trading position produces less benefit to profits than exploiting the potential of predictions with high levels of predictive correlation. The jagged contours of the third plot show that the performance benefits of the adaptive step rule may be unstable. This instability is caused by the sensitivity of the trading position to the level of the tolerance parameter. Further analysis shows that all three of the path dependent trading rules reduce the size of the lower profit region compared to the class of parameterised trading rules, which was described graphically in figure 4.6.

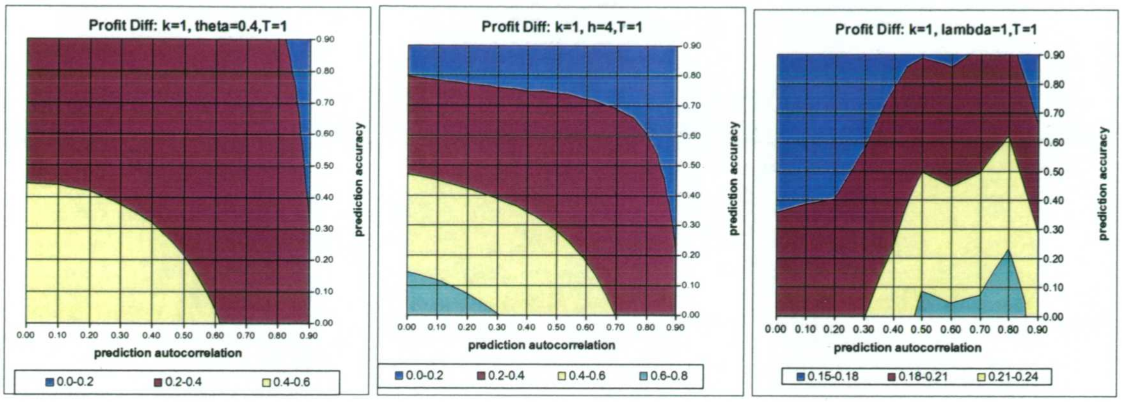


Figure 5.7 depicts the contour profiles for changes in average profit for the path dependent trading rules with a transaction cost of $T = 1.0$.

Figure 5.7 shows the average profit difference for the three path dependent trading rules compared to the original rule with a high transaction cost ratio of $T = 1.0$. For all three plots, there are no negative profit regions, indicating that smoothing improves performance for prediction from all characterised forecasting models. The curved contours for plots 1 and 2 are similar in shape and show increased profit for lower predictive correlation and lower prediction autocorrelation. This is because the smoothing of trading positions based on predictions with a low level of prediction autocorrelation involves higher turnover, which generates higher average transaction costs. Similarly, predictions with low levels predictive correlation have less predictive information to exploit. Thus, in this case, any smoothing of the trading position, based on a moving average, produces most profit improvement. In contrast, the contours in the third plot generated using the adaptive step rule, show increasing profits for higher prediction autocorrelation combined with low prediction accuracy. This can be explained by the fact that as this rule reduces turnover most effectively for high levels of prediction autocorrelation, which generates an opposite contour gradient.

These experiments illustrate the potential importance of the path dependent trading rules for improving the profitability of trading strategies with a fixed transaction cost rate. Even though the path dependent rules have just representative parameter values, they show that significant improvements in profitability can be achieved over the original trading rule (k = 1). This is particularly evident for higher transaction costs and predictions with low levels of predictive correlation and prediction autocorrelation.

We conducted further simulation experiments to investigate the effects of varying the smoothing parameter with fixed forecast characteristics and fixed transaction costs. A summary of these experiments is given in figure 5.8, where we focus on the path dependent trading rule with exponential smoothing. In these experiments, prediction autocorrelation is fixed to zero in order to focus on the effect the predictive correlation.

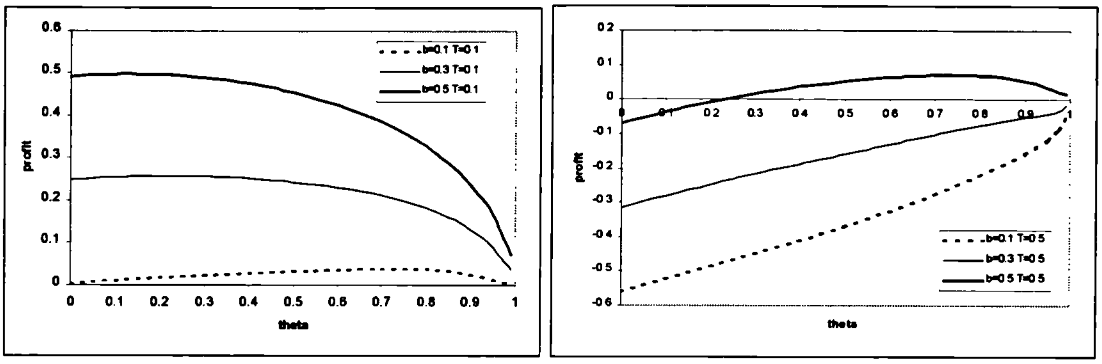


Figure 5.8 plots the profit profile of the first path dependent trading rule with varying smoothness controlled by a decay parameter, θ, for three levels of prediction accuracy, b, of 0.1, 0.3 0.5 and a transaction cost parameter of T=0.1 (left) and T= 0.5 (right).

Figure 5.8 demonstrates the profitability of the path dependent trading rule, given in equation (5.3), for two levels of transaction cost and sets of predictions based on three different levels of predictive correlation. The left-hand graph shows the profit profiles for trading three levels of predictive correlation (b = 0.1, 0.3 and 0.5), in an environment with a low transaction cost ratio of T=0.1. The right-hand graph shows the profit profiles for trading the three different levels of predictive correlation, in an environment with a higher transaction cost ratio of T =0.5. The graphs show that different levels of prediction correlation result in different trading profits. In all cases, higher correlation gives higher profits for the same trading conditions. They also show that the value of the smoothing parameter has a significant impact on trading performance.

If we consider the left-hand graph, in the case of high prediction accuracy, b =0.5, with no smoothing of the trading position, (θ=0) the average profit is equal to 0.5. In this case, increasing the smoothing only results in decreased profits, with large levels of smoothing (above 0.8) producing a dramatic reduction in trading performance. For a prediction accuracy of 0.3 a smoothing parameter of 0.2 can slightly increase profits although high levels of smoothing only result in significantly reduced profits. For low levels of prediction correlation (b =0.1), however, profitability is significantly improved by smoothing, with optimal performance achieved with θ equal to 0.8. This is explained by the fact that increasing smoothing reduces turnover, which has the effect of lowering overall transaction costs at the expense of not fully exploiting potential predictive ability. In the case of predictions with low predictive correlation (b = 0.1), there is little predictive ability to exploit so profits are boosted by introducing a high degree of positional smoothing.

If we consider the right-hand graph, then we can see how the increased transaction costs result in reduced profit levels compared to the equivalent experiments in the left-hand graph. However, not only do higher transaction costs result in lower profits but they also affect the value of smoothing the trading position. For all three levels of prediction accuracy, increasing the smoothing parameter results in increased profits, with optimal performance achieved at 0.8 for b =0.5, and total smoothing (θ=1) for b =0.1 and b=0.3, which indicates no trading at all (i.e. profits are not capable of overcoming transaction costs). Trading systems with negative expected profits should always result in total smoothing in order to minimise transaction costs. In the next subsection we provide an explanation of trading performance in terms of an analogy with a bias-variance trade-off, which is commonly found in statistics.

5.2.4 Explanation of Performance in terms of a Bias-Variance Trade-off

In this section we explain the performance of the path dependent trading rules in terms of an analogy with a bias-variance trade off.

In statistics, performance is typically described in terms of model accuracy which is commonly decomposed into two complementary quantities, known as bias and variance. Bias reflects systematic error and is typically measured using mean error. If all things are equal, a model is preferred if it has small bias. On the other hand, variance measures the dispersion of the forecast error and so again, if all other things are equal, a model is preferred if it has small variance.

The concepts of bias and variance are commonly used in statistical modelling to describe the relationship between model complexity and model error. A model with the best generalisation is achieved when the best compromise is reached between small bias and small variance. In model construction this trade-off between bias and variance is reached by controlling model complexity. Thus, simple models tend to have large bias and highly complex models have too much flexibility, which induces large variance. Techniques that control the complexity of a model provide a means of finding the optimal position on the bias-variance trade-off curve and so maximise expected performance, as illustrated conceptually in figure 5.9.

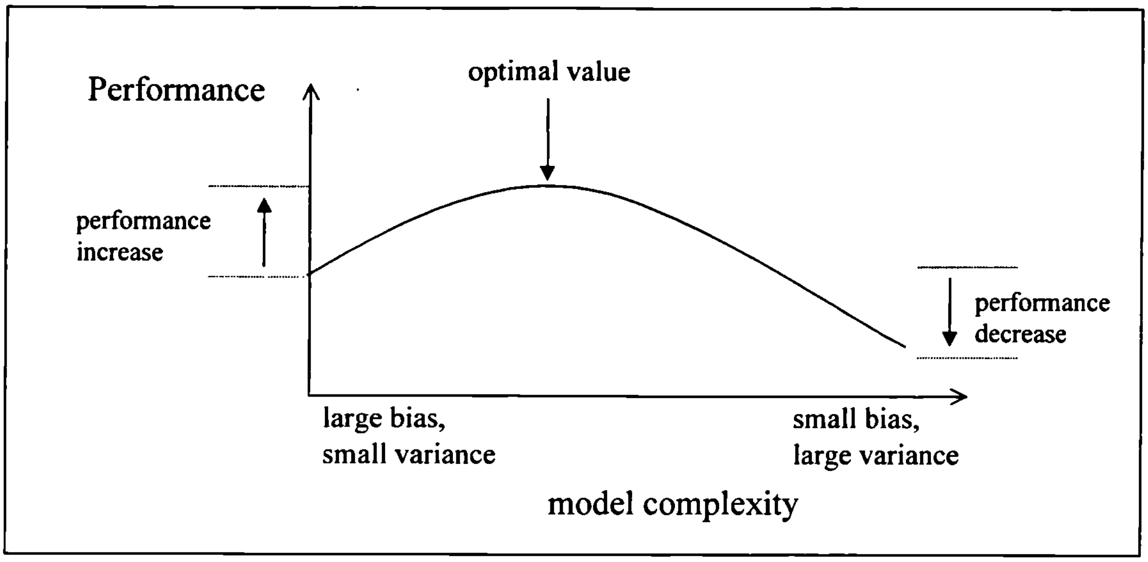


Figure 5.9 depicts the bias-variance trade-off commonly found in statistics.

For trading strategies, we can represent "bias" as the rigidity of the trading policy, which describes the smoothness of the trading position through time. Thus, simple trading models have large bias as they can only change the trading position slowly regardless of the predicted return. In contrast, "variance" represents the flexibility of the trading policy, which relates to the variability of the trading position through time. Thus complex models have large variance as they frequently modify the trading position. The optimal trading policy can then be found by a decision modelling technique that can control the trade-off between exploiting predictability (low bias) and minimising trading costs (low variance). This could also be viewed as a trade-off between opportunity costs and actual costs.

Under this bias-variance analogy, the class of parameterised trading rules, described in equation (5.2), is considered a "sensitive" model with small bias and large variance. This is because the trading policy only exploits predictability and makes no attempt to control costs by smoothing the trading position through time. The path dependent trading rules, described in equations (5.3), (5.4) and (5.5), however, allow for a trade-off between bias and variance by providing an additional parameter to smooth the trading position through time. In each case, a smoothing parameter provides a means of controlling model complexity in order to achieve a compromise between small bias and small variance.

In this section the simulation experiments have shown the importance of controlling bias and variance when trading in the presence of transaction costs. Results show that path dependent trading rules are most effective at improving performance in the presence of high transaction costs and low levels of prediction accuracy and prediction autocorrelation. This can be explained by considering two factors: the correlation between the trading position and the predicted return, and the stability in the trading position. A path dependent trading rule has the ability to reduce correlation but increase trading position stability. This has the effect of introducing more bias to the trading position but lowering variability. Under the conditions of low predictive correlation, low prediction autocorrelation and high transaction costs; controlling smoothing allows the trading policy to move towards to optimal value on the bias-variance curve and so significantly improve performance.

In this section we have developed "path dependent" parameterised trading rules and shown, using simulation experiments, that these models can be used as a valuable tool for optimising trading strategies for predicted returns in the presence of negligible or fixed transaction costs. However, we also need to consider trading in the presence of more general trading costs which requires the development of more advanced decision modelling techniques. In the next section, we apply reinforcement learning to the implementation and optimisation of trading strategies for predicted returns. This modelling approach relaxes some of the assumptions underpinning parameterised trading rules and allows trading to be optimised in more general conditions, which include transaction costs arising from market impact.

5.3 Optimising Trading Strategies using Reinforcement Learning

In the previous section, we optimised trading strategies under the assumption that the system state, which represents all relevant information, only consisted of the current predicted return. In this case, the trading policy was a direct mapping from the predicted return to the trading position, implemented through a class of parameterised decision rules. The advantage of this modelling approach is that *a priori* knowledge of the functional form of the trading policy can be efficiently implemented. Whilst the simplest form of these rules assumes that transaction costs are insignificant, we compensated for the effect of transaction costs by developing "path dependent" rules, which used smoothing techniques to incorporate previous trading positions.

In this section, we develop a more general decision modelling approach that is capable of optimising a trading strategy in the presence of more general sources of transaction costs. These commonly arise from frictional forces such as trading restrictions and market impact, as described in section 4.3. In this framework we <u>do not</u> assume that the state only consists of the current predicted return but rather it is extended to include the previous trading position and other variables that influence trading performance, as described in equation (4.4).

First consider the simplest case, where transaction costs are only influenced by the state of the portfolio (i.e. the previous trading position) and not the state of the market. In this case, the system state is simplified to consist of the predicted return, denoted by $\hat{y}_{t+1}$, and the previous trading position, a_{t-1}. In this case, the trading policy is a two dimensional mapping from the state information, $\hat{y}_{t+1}$ and a_{t-1}, to the trading position, a_t, which takes the form:

$$a_t = f(\hat{y}_{t+1}, a_{t-1}) \tag{5.6}$$

where f is an unknown function.

Even for this simple case, the functional form of the mapping from state to action is, in general, non-trivial and it is difficult to identify any suitable class of parameterised decision rules. For example, a negative trading position may be appropriate even for a positive predicted return if the previous trading position is negative because of transaction costs. Furthermore, the potential of more general sources of transaction costs, and consequently a more general state, makes the process of defining decision rules too complex and restrictive, except for idealised conditions. However, although this trading system is analytically intractable, we can use alternative decision modelling techniques based on powerful empirical methods to approximate the optimal trading strategy. This is achieved using a computational modelling technique known as reinforcement learning (see chapter 3 for details), in which the functional form of the mapping from state to action is learnt from a data sample. Reinforcement learning (RL) is closely related to dynamic programming but unlike DP it is capable of identifying the optimal policy directly from past actions without requiring complete knowledge of state-action dynamics. In addition, RL can operate by learning in real-time, which has the benefit of allowing the trading policy to be optimised in the presence of time varying system dynamics (e.g. degrading forecast model characteristics).

Reinforcement learning is based on the Markov property, so in order to develop a decision framework we need to specify all possible states and actions in each time period. If we consider the simplest case, then the state just consists of the predicted return and the previous action, and the action states are just the range of possible trading positions, as shown above. The classification of state and action pose a problem as, in general, predicted returns and trading positions are considered to both have a continuous state space (although often security prices are denominated in fixed increments). One simple method of overcoming this issue is to impose a restriction on the size and increment of the action space, which defines possible trading positions, and similarly restrict the predicted return to a discrete, finite number of states. For the trading position this implies that only discrete fractions of the fund (or the leverage limit for futures trading) may be allocated to any particular asset. For the predicted return this implies rounding the forecasts to the nearest allowable state. It should be noted that discretisation of the action and state variables is an approximation to the properties of the trading system and so may lead to sub-optimal trading performance. In the next subsection we addressed this issue with the development of parameterised reinforcement learning, which approximates the value function for continuous action-states using a neural network. In this case, the neural network acts as a general function approximation technique that overcomes the limitations that arise from discretisation.

For a trading system we need to specify the reward received from an action in a particular state. This is often critically dependent on the transaction costs generated from changing the trading position. In general, transaction costs are dependent on the state of the market, the assets traded, and the nature of the trader (i.e. hedge fund or market maker) as described in more detail in chapter 4. The estimation of transaction costs typically involves developing some transaction cost model that approximates the expected cost of trading. This enables the reward function to take into account the expected cost of trading and so be included in the optimisation process. Assuming that the state variables, actions and rewards can be specified for all possible states then we can design an RL algorithm to optimise a trading strategy for predicted returns from a forecasting model. Reinforcement learning can be implemented using an on-line Q-learning algorithm (discussed in chapter 3), as presented in figure 5.9.

$s_t = \{ s_t', \hat{y}_{t+1} \}$ where $\hat{y}_{t+1}$ = predicted return and s_t' = state of the portfolio/market
Initialise value function $Q(s, a)$
Initialise state of portfolio/market s_0' and generate first prediction $\hat{y}_1$
Repeat
    *** ***decision stage*** ***
    change trading position, a_t using current policy derived from $Q(s,a)$

    when $t=t+1$
    observe state of market, s_{t+1}' and actual asset return, y_{t+1}
    calculate trading reward, $r_t(a_t, y_{t+1}, s_t')$
    generate new prediction, $\hat{y}_{t+2}$
    *** ***learning stage*** ***

    compute temporal difference, $d_t = r_t + \gamma \max_{a(t+1)} Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)$
    update value function, $Q(s_t,a_t) = Q(s_t,a_t) + \alpha d_t$
Until t is terminal

Figure 5.9 details the classical on-line, one-step, tabular Q-learning algorithm to optimise a trading strategy for predicted returns from a forecasting model.

In this implementation the state, s, consists of two components on the basis of the state of market/portfolio, s' and the predicted return, $\hat{y}_{t+1}$ which is generated by the forecasting model. This highlights the difference between the different sources of the system state.

Before the trading strategy begins, the value function (which for discrete states is implemented as a look-up table) is initialised, the state of the market/portfolio observed, and the prediction for the asset return in the first time step generated from some forecasting model. The algorithm proceeds by entering a decision stage, which selects the trading decision on the basis of the policy derived from the value function for all possible state-action pairs. For RL applications, this is traditionally implemented using a greedy algorithm which selects the action proposed by the optimal policy. An alternative is a near greedy (ε-greedy) algorithm, which selects the optimal action most of the time, but occasionally (with probability ε) selects sub-optimal actions. The main advantage of an ε-greedy action selection rule is that, in the limit as the number of time periods increases, every action will be sampled a large number of times, thus ensuring value function convergence. This provides an effective means of balancing exploration and exploitation, as described in section 3.4. The main disadvantage of the ε-greedy approach is that when it explores it chooses equally from among all actions so is just as likely to choose the worst action as the best action. More complex implementations, such as softmax action selection rules, rank probabilities according to value estimates (for more details see chapter 2 of Sutton and Barto, 1998.)

At the next time step (when $t = t+1$), the state of the market is observed and actual asset return calculated. The reward associated with the trading decision is computed and the predicted return received from the forecasting model. The learning stage can now begin with the value function updated on the basis of the trading reward and an estimate of the investment value of following the best policy from the next time period. The learning rule uses the difference between the approximated value of the trading policy and the current value function. The learning parameter, α, controls the rate at which the value function is updated. The discount parameter, γ, specifies the present value of future rewards. If $\gamma = 0$, the policy is "myopic" and so only maximises immediate rewards. As γ approaches 1, the objective takes more account of future rewards and so the trading policy becomes more farsighted.

Although this RL algorithm has been designed for optimising trading strategies it has a number of practical limitations that reduce its application to trading predictability. The main problem, which is common in many RL applications, is that the learning process requires large quantities of data before approximating the optimal policy. However, for trading systems, we can address this drawback by modifying the learning stage to learn from actions that could have been taken as though they had been taken. This is a specific property of the financial domain where we can assume partial separation between actions and subsequent states of the system. In the next section we show how we can exploit this knowledge to enhance RL for trading systems.

5.3.1 Enhancing on-line RL using "multi-action" learning

The traditional approach of control systems is to assume that the agent-environment interacts in such a way that a selected action directly affects the subsequent state of the system. For our trading system, this assumes that trades from a single investor directly affect the state of the market and also the predicted return from the forecasting model. However, these assumptions are partly unnecessary for our trading system as we can assume independence between the state of the market and the selected trades. This assumption is on the basis that the market is fully elastic and so trading of an individual cannot permanently affect the market state. Thus actual returns (and also predicted returns) are not adversely affected by previous trading decisions. This enables the learning process to consider other actions (trading positions), in addition to the selected action, without affecting the state in the next time period. Thus an entire region of the value function is updated as opposed to only one state for the selected action. This is described graphically in figure 5.10.

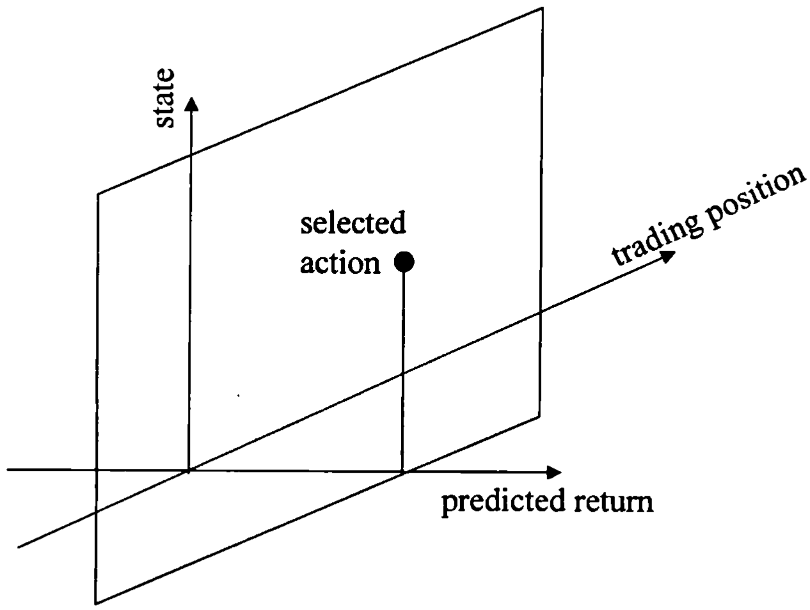


Figure 5.10 conceptually depicts the region (shaded area) of the value function in which learning can take place for a given predicted return compared to the selected action.

For trading systems, figure 5.10 shows conceptually that the learning stage in standard RL can be extended to cover a region within the state-action space. This enhancement to the learning stage of the standard RL algorithm is implemented using additional loops that try all possible actions and states. We refer to this as "multi-action" learning to indicate that learning takes place from all possible actions (multiple actions) in each time period. This modified Q-learning algorithm is presented in figure 5.10.

$s_t = \{ s_t', \hat{y}_{t+1} \}$ where $\hat{y}_{t+1}$ = predicted return and s_t' = state of the portfolio/market
Initialise value function $Q(s, a)$
Initialise state of portfolio/market s_0' and generate first prediction $\hat{y}_1$
Repeat
    *** ***decision stage*** ***
    change trading position, a_t using current policy derived from $Q(s,a)$

    when $t=t+1$
    observe state of market, s_{t+1}' and actual asset return, y_{t+1}
    calculate trading reward, $r_t(a_t, y_{t+1}, s_t')$
    generate new prediction, $\hat{y}_{t+2}$
    *** ***multi-action learning stage*** ***
    For (all possible states s_t')
        For (all possible actions, a_t)
            approximate "would have been" reward r_t
            compute temporal difference, $d_t = r_t + \gamma max_{a(t+1)} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$
            update value function, $Q(s_t, a_t) = Q(s_t, a_t) + \alpha d_t$
        Next
    Next
Until t is terminal

Figure 5.10: on-line tabular "multi-action" Q-learning with additional loops in the learning phase.

The modified algorithm proceeds in the same manner as the classical algorithm by selecting actions based on the current value function. In the learning phase, however, a feature specific to the financial markets is exploited, on the basis of independence between market asset returns and our own trading decisions. This separation enables the reward to be approximated for different actions and states of the market and so allow the learning stage to explore a region of the state-action space in each time period. This enables the value function to be updated for all actions which *could have been* taken as though they *had been taken*. This differs from the typical sequential learning in which an outcome (i.e. the next state) is dependent on the action taken, hence the value function is only updated for the selected action. In the standard case, this creates a trade-off between exploration (trying all actions) and exploitation (choosing the best action given the current value function). Thus, in addition to learning from a region of state-action space, our modified algorithm also allows the RL algorithm to avoid the normal trade-off between exploitation and exploration and so increase learning efficiency. The increase in efficiency of "multi-action" learning, over the sequential algorithm, is dependent on the relative size of the region of the state-action space that can be explored. It is considered that such speed up is essential as predictable market anomalies may have a limited life span.

We propose that this modification makes RL feasible for practical applications in investment finance where value function convergence is slow due to low predictability and where there may be limited amounts of valid empirical data. In the next section we develop simulation experiments to test the effectiveness of our algorithm. We build upon the synthetic trading system developed in chapter 4 to simulate predicted returns from a forecasting model with two controllable characteristics. The experiments examine the speed of multi-action learning compared to a standard RL implementation for a simple transaction cost model.

5.3.2 Simulation Experiments

In this section, we describe simulation experiments which are designed to demonstrate the learning effectiveness of "multi-action" learning compared to standard Q-learning. A data generating process is constructed to simulate both actual asset returns and the output of a forecasting model with two controllable predictive characteristics, as described previously in equations (4.13) and (4.15). For these simulations, the trading position is discretised between -1 to $+1$ with a parameter to control the number of positional states, as discussed for a bin-smoother in section 4.3. In this representation, possible trading positions can be considered to signify some degree of fractional asset allocation with negative trading positions representing the ability to short sell. The predicted return series is generated from a standard normal distribution, as described in equation (4.13), and so we arbitrarily constrain the state space of the forecast from -2 to $+2$ using rounding and use a parameter to control the number of states within this range.

For these simulation experiments, we simplify the real case by assuming that transaction costs are only influenced by the state of the portfolio and not market conditions. This means that the cost of trading is some pre-determined function of the change in the trading position. Thus the predicted return and the previous trading position are sufficient to represent the current state of the system, as described by equation (5.6). The effect of market impact is assumed to act as a "liquidity threshold" and modelled using a simple trading restriction, as discussed in chapter 4. In this model we assume that within some specified range, trading incurs no cost whilst changes in the trading position outside this range are considered to have a significant impact on costs. This effect makes trading undesirable outside of the trading restriction and so acts as a constraint on trading. This approach is motivated by the assumption that practical trading conditions restrict trading opportunities and can be considered an exaggerated case of a non-linear market impact model. For the purposes of these simulation experiments we ignore other trading costs so the investment return at each time step is defined as

$$R_t = a_t y_{t+1} \qquad where\ |a_t - a_{t-1}| < \lambda \tag{5.7}$$

where a_t is the trading position (i.e. the percentage of the fund invested in asset y), λ is the tolerance parameter which restricts the change in the current trading position, y_t is the actual asset return and r_t is the percentage trading profit.

We generated a training data set consisting of predicted and actual returns for 2000 observations with a predictive correlation of 0.2. In order to test the performance after each time period (i.e. simulating real-time use of such a model) we generated an additional test sample consisting of 4000 observations with the same predictive characteristics. For these experiments, the number of possible states for the trading position and the predicted return state variables were set to 10 so that each state represented a 20% change in the trading position and a predicted return of 2% respectively. This is considered an oversimplification of a typical trading system but adequate for testing learning efficiency.

The value function was initialised with random numbers close to zero and the initial trading position set to zero to compare restults for standard RL and multi-action RL. For the standard RL algorithm we assumed that action selection is made on the basis of a greedy policy. The learning rate for both algorithms was set to 0.05 and the discount parameter to 0.9. Figure 5.11 shows the sample learning curves for both the standard and multi-action learning algorithms in the case where there is no trading restriction.

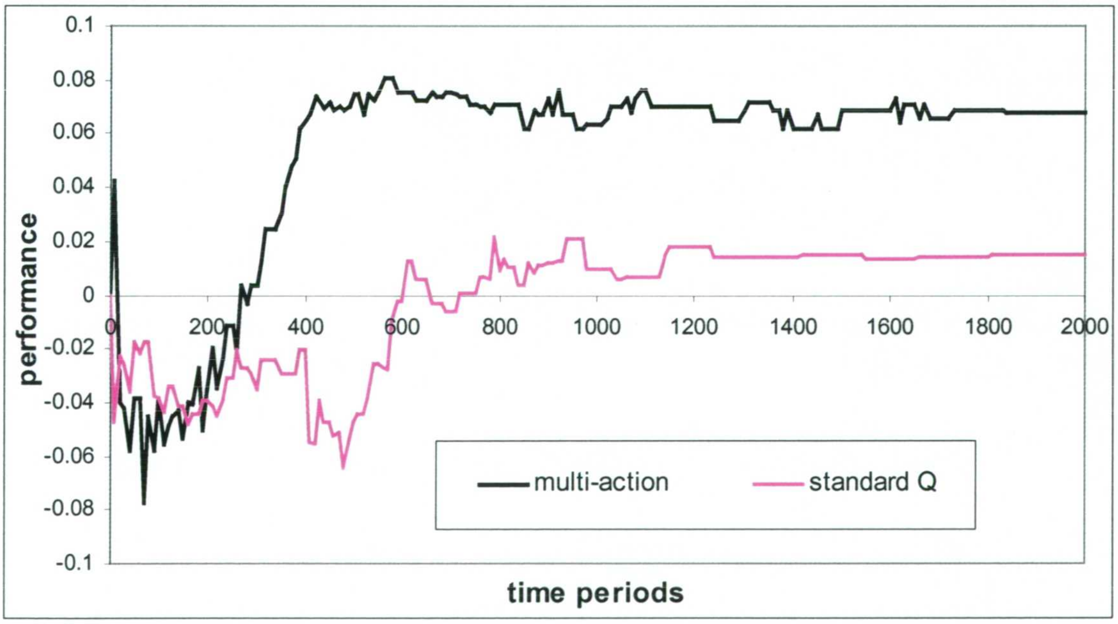


Figure 5.11 shows the sample learning curves for test performance (i.e. average profit) taken for the two algorithms, standard Q-learning and multi-action learning.

The graph shows that initially performance is poor for both algorithms, however, after 250 time periods the multi-action learning rule rapidly improves performance until converging after approximately 400 time periods. The standard Q-learning rule only starts to improve performance after 500 time periods until converging after roughly 1200 time periods. At convergence there is a considerable performance differential between the two algorithms. This can be explained by the lack of any exploration in the standard Q-learning algorithm thereby limiting the ability of the rule to learn better trading policies. On the other hand the multi-action learning action is able to fully explore all possible actions while exploiting the best trading policy at each time step. It is interesting to note the jaggedness of the sample curve. This is partially due to the coarsely defined trading position and predicted return grid, which leads to a poor approximation to the optimal trading policy. This further motivates the neural network approach for value function approximation which we describe later. We can observe the effect of encoding the predicted return and the trading position by plotting the value function across all possible values. Figure 5.12 shows the value function for the multi-action update rule at convergence (i.e. after 2000 observations). Note that in this plot the fourth dimension of the value function, the previous trading position, is set to zero.

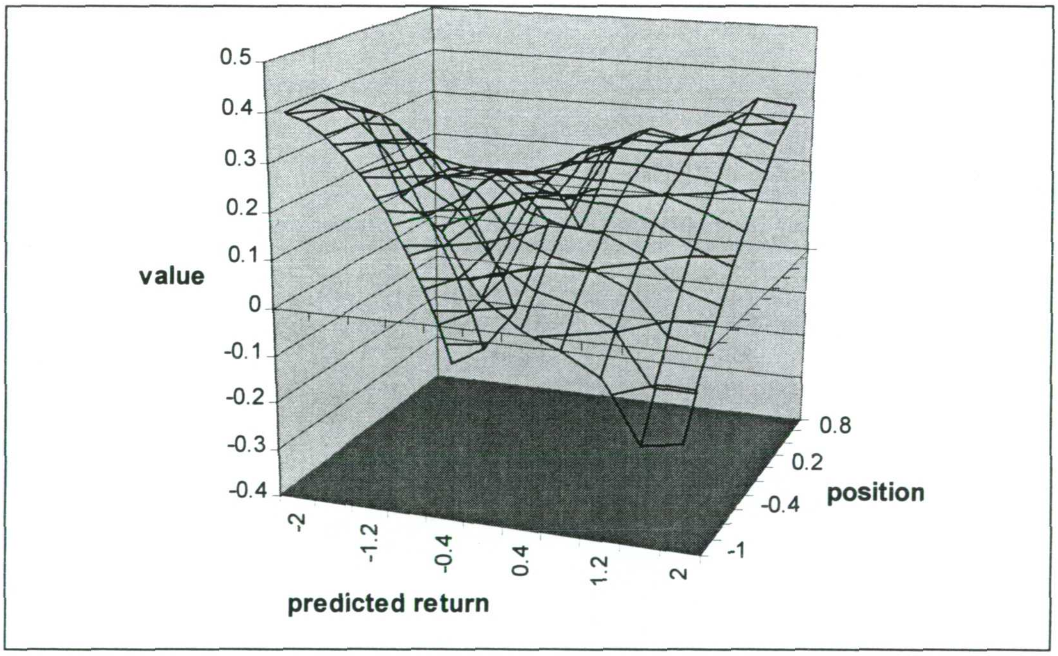


Figure 5.12 shows the shape of the value function for the multi-action update rule at convergence.

This graph shows that although the value function has learnt some of the underlying structure jagged parts in the state space still exist. It is proposed that performance may be improved by parameterisation of the value function, using a neural network. This issue is addressed further in the next subsection. The performance advantage of multi-action learning is primarily due to the region of state-action space that can be updated in each time period. We can test the effect of changing the size of the update region by modifying the trading restriction parameter. This is because the trading restriction constrains the size of the state-action space. For this trading system the effect of the trading restriction parameter for a given trading limit is measured in terms of a percentage trading restriction which is defined as:

$$\%\,\text{Trading Restriction} = 1 - \frac{\lambda}{\text{trading limit}} \tag{5.8}$$

where, in this case, the trading limit is set to 2 for fractional asset allocation $(-1, +1)$.

A summary of the results from modifying the trading restriction is presented in table 5.1

Trading Restriction	RL Method	Performance (at n time steps) *10									
		100	200	300	400	500	600	700	1000	1500	2000
20%	Standard Q	-0.08	0.077	0.044	0.037	0.100	0.072	0.155	0.155	0.192	0.189
	Multi-action	0.618	0.708	0.698	0.708	0.708	0.716	0.688	0.716	0716	0.716
40%	Standard Q	0.052	0.084	0.090	0.085	0.101	0.111	0.099	0.133	0.150	0.150
	multi-action	0.354	0.548	0.542	0.532	0.536	0.518	0.532	0.538	0.536	0.536
60%	Standard Q	-0.108	-0.043	-0.04	0.03	0.017	0.07	0.061	0.08	0.076	0.085
	multi-action	0.144	0.348	0.348	0.348	0.348	0.348	0.340	0.346	0.360	0.360
80%	Standard Q	-0.115	-0.087	-0.08	-0.078	-0.076	0.06	0.062	0.063	0.065	0.065
	multi-action	0.014	0.060	0.062	0.031	0.110	0.121	0.144	0.146	0.160	0.160

Table 5.1 shows the performance, in terms of average profit (*10), after n time periods of "multi-action" and standard Q-learning for simulations with different % trading restrictions.

Table 5.1 shows that, at convergence, the larger the percentage trading restriction the lower average profit for both multi-action and sequential learning. This is because the restriction limits the ability of the trading policy to exploit predictability and so reduces profitability. The results also show that the multi-action algorithm learns more efficiently than standard Q-learning across the entire range of trading restrictions. In general, the trading restriction affects the convergence rate as it governs the size of the region in action-state space that is explored in each time step. The higher the percentage trading restriction, the smaller the update region and consequently the smaller the improvement over standard Q-learning. It is also interesting to note that the rewards from the standard update rule are consistently lower than the multi-action version. This can be explained as the traditional approach is unable to sufficiently explore the internal state space and learn the optimal value function using a greedy algorithm. It is thought that multi-action learning will explore considerably better than an ε-greedy rule. In the next subsection we develop a parameterised value function using a neural network to approximate the value function across the space of possible state-action pairs.

5.3.3 Parameterised Reinforcement Learning

In the previous subsection we developed an enhanced reinforcement learning algorithm for optimising a trading strategy in the presence of general trading costs. In this model, the learning process relied on some *ad hoc* quantisation of the trading position and the predicted return space. This enabled the current estimate of the value function (or Q-function) to be stored in a lookup table, which partitioned the state and action spaces into separate regions with an associated entry for each. However this approach relies on how well the value function is represented by quantisation of the state-action space. A coarsely defined trading position (or predicted return) grid may lead to a poor approximation of the optimal trading policy. In contrast, too fine a trading position (or predicted return) grid requires large quantities of training data to learn the value function for each cell in the look up table. In typical trading systems it is most appropriate to model the state-action space as a continuous state space problem where trading positions and predicted returns can take any bounded real value.

The intractable nature of a continuous state space can be addressed by value function approximation. This involves modelling the value function in order to approximate the value of a policy across the entire state-action space. The reinforcement learning algorithm can be applied to a function approximator in order to learn the input-output mapping to the required accuracy. The aim is to learn the function in order to generalise information to other states in an analogous manner to forecasting or pattern recognition tasks. The RL update rule operates by updating the parameters of the neural network rather than a look-up table (for more details see chapter 3).

In stochastic environments, where rewards from action-states are subject to noise, the learning process introduces the possibility of *over-fitting*. This occurs when the function approximator learns the noise component of the training set thus reducing its ability to generalise to new data. The choice of a suitable parameterisation requires some knowledge of the shape of the function to be approximated. If we consider a trading strategy to maximise profits with no transaction costs then the optimal trading policy and the optimal value function can be simply computed as shown in figure 5.13.

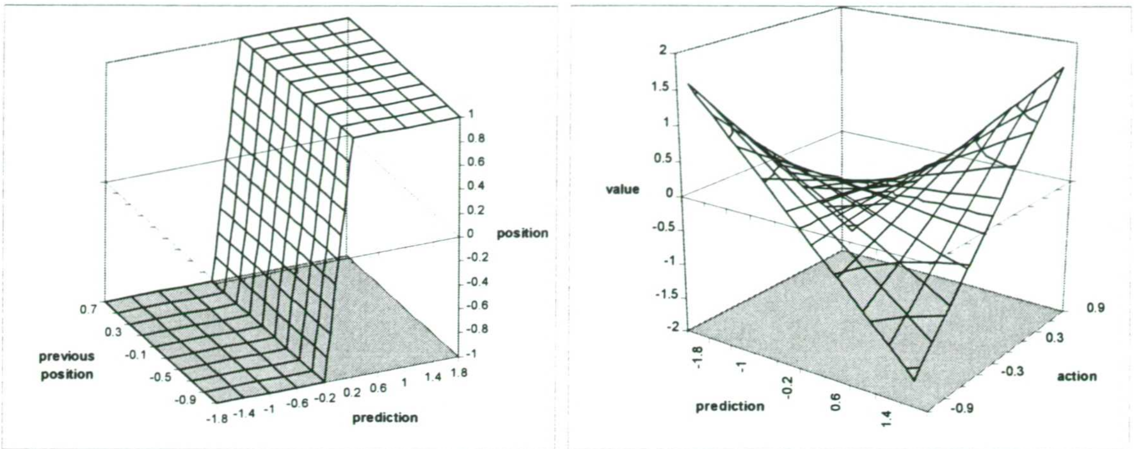


Figure 5.13 shows the optimal myopic trading policy and the optimal myopic value function (expected return function) for maximising profitability with no transaction costs.

The left-hand graph of Figure 5.13 shows that for asset allocation with no trading restriction the optimal trading policy is a step function of the sign of the prediction return. The optimal trading position flips between the two extreme positions depending on the sign of the predicted return. In this trading environment, with no costs, the previous trading position has no effect. However, even though the optimal policy is relatively straightforward (equivalent to the parameterised trading rule with k =0) the associated value function (expected reward function) is a more complex function consisting of two overlapping quadratic functions. Due to the potential complexity of the value function in more general cases we decide to parameterise the value function using a neural network, which is considered to be a general-purpose function approximation technique. In the next subsection we test the performance of connectionist reinforcement learning against the optimal myopic policy for a trading system with a trading restriction.

5.3.4 Simulation Experiments

In this section we describe simulation experiments designed to test the performance of RL using the multi-action update rule and the value function parameterised with a neural network. The RL performance is tested for trading in the presence of a trading restriction and compared against the optimal myopic policy.

A stationary synthetic trading environment was set up with actual and predicted returns generated, as described in chapter 4. For these experiments there was no need to quantise the predicted return variable as parameterised RL allows for a continuous state space. Similarly, the trading position was allowed to take any position between –1 and +1. The RL learning stage is implemented using the multi-action update rule, as described in section 5.3.1. For a continuous state-action space, the additional loops in the learning phase cannot update all possible states so stratified sampling is applied in order to randomly sample trading positions evenly across the range of possible values. The number of samples is based on the trade-off between computational work and model accuracy. In principle, a higher number of samples will lead to higher computational work but greater accuracy. For a stationary environment we can implement the RL algorithm in batch mode, which means that neural network weights are only updated after all the data samples have been processed. For batch update we can simply generate random trading positions for the state-action space which will ensure that training will fully explore the state-action space given a large number of training iterations.

Trading costs are simulated by a trading restriction model, as described in equation (5.7). For this trading environment the myopic trading policy is not a simple step function that switches between the two states (-1, +1) given the sign of the predicted return. The myopic trading policy is dependent on the existing trading position, the trading restriction parameter λ, and the predicted return. The optimal trading policy maximally increases (decreases) the trading position for a positive (negative) forecast within the trading limits, which can be defined as

$$\begin{aligned} a_t &= \min(a_{t-1} + \lambda, 1) && \text{if } \hat{y}_{t+1} > 0 \\ &= \max(a_{t-1} - \lambda, -1) && \text{if } \hat{y}_{t+1} < 0 \end{aligned} \tag{5.9}$$

where $\hat{y}_{t+1}$ is the predicted return.

The optimal myopic policy fully exploits the information of the predicted return to maximise profits in the current time period without considering the consequences of changing the position for subsequent time periods. This may be sub-optimal for trading that is subject to some restriction and justifies the development of a trading policy using RL.

The neural network is specified with one hidden layer and 6 hidden units. A training set is generated consisting of predicted and actual returns for 1000 observations with a predictive correlation of 0.2. For the RL method, the neural network weights are initialised close to zero and the learning rate set to 0.1 and the discount parameter to 0.9. After each time period the temporal difference is used to compute the error derivative with respect to each weight. The weight changes are stored and the weights updated after an iteration of the data set (i.e. after all the training patterns have been presented to the network). This iterative process is repeated until the in-sample performance of the RL policy converges. An example of a training curve is shown in figure 5.14.

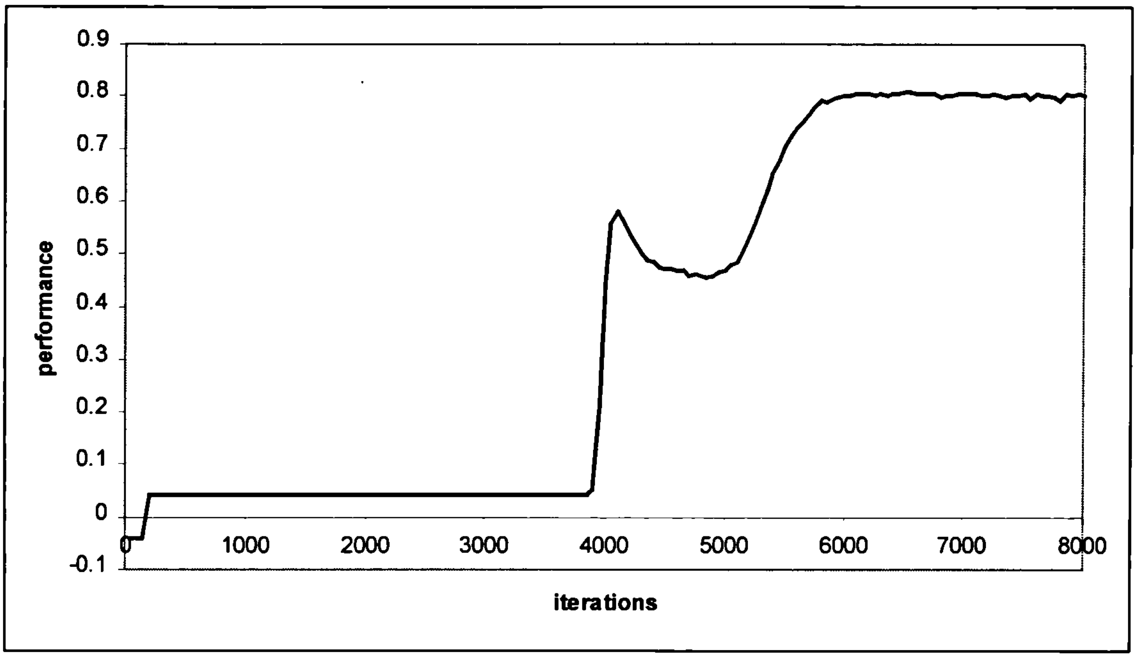


Figure 5.14 shows the training curve for optimising trading with no trading cost restriction, where the performance is measured in terms of average profit.

The training curve graph shows the profile of the learning process for the neural network. The graph can be interpreted as showing three stages of learning: the first increase in performance (from –0.04 to 0.04) relates to the value function approximator learning the correct sign of the value of each trading position given a predicted return. The second stage (from 0.04 to 0.58) relates to learning the linear components of the relationships describing the magnitude of the value of a trading position given a predicted return. The third stage (from 0.58 to 0.81) relates to learning the two quadratic functions that make up the true value function. The performance of the RL trading policy is 0.8112 at convergence, which compares to 0.7967 for the optimal myopic trading policy, as given in equation (5.7). In the case where there is no trading restriction we expect the optimal myopic policy to give optimal out-of-sample performance. To test the generalisation properties of the neural network we generated 50 test data sets of 1000 observations with the same predictive characteristic as the in-sample data in order to computed the out-of-sample performance of both the RL policy and the myopically optimal policy. The results from these experiments are shown in table 5.2.

	Performance Metrics (50 test sets of 1000 observations)		
	Average	**Standard Deviation**	**Standard Error**
Multi-action RL policy	0.79671	0.03015	0.00426
myopically-optimal policy	0.79781	0.02998	0.00424
Difference	0.00110	0.00640	0.00090

Table 5.2 shows the performance metrics for 50 test sets of 1000 observations.

Results from the test data sets show that the expected trading performance of the RL policy (0.7967) is only slightly lower than the optimal trading policy (0.7978) with no statistically significant difference. This indicates that the value function approximated by a neural network has a trading policy with very good generalisation properties. The output of the neural network with respect to the input variables is shown in figure 5.15.

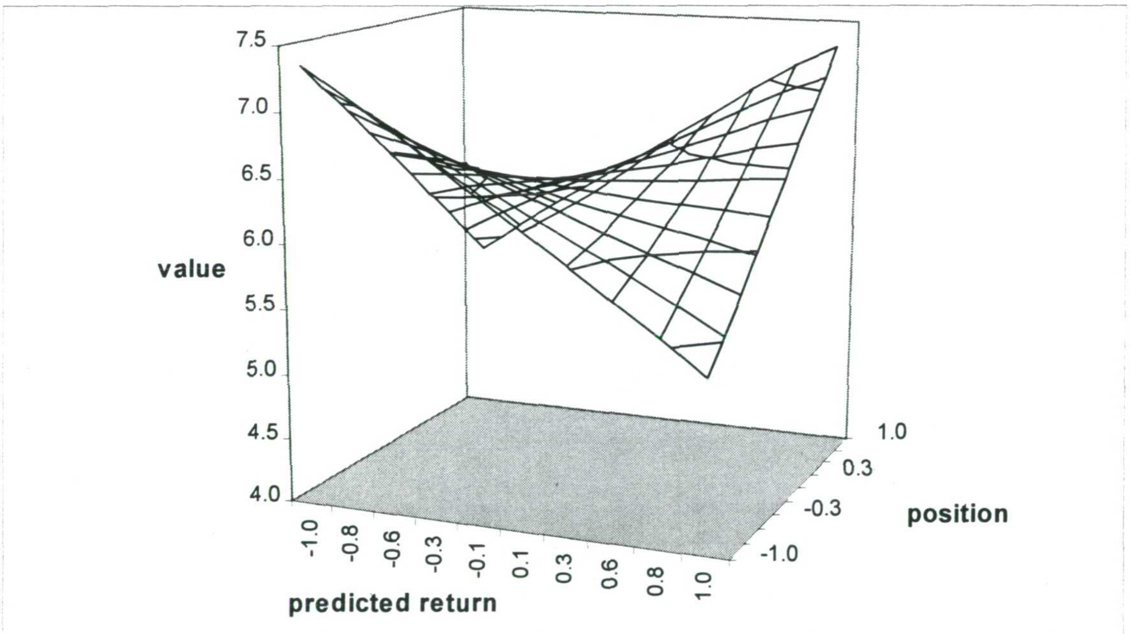


Figure 5.15 shows the value function approximated by a neural network.

A comparison of figures 5.15 and 5.13 shows that the neural network accurately learns the shape of the optimal myopic value function. The difference between the two value functions is the relative magnitude of the output value. This is due to RL algorithm, which learns both the short term and longer-term value of each trading position given a predicted return. This is achieved by the non-zero discount parameter γ in the RL update rule. A comparison of figures 5.15 and 5.12 indicates that the neural network gives a better value function approximation that discretising the state-action space and using a look up table. The neural network smoothes between neighbouring states to provide a more accurate approximation to the shape of the optimal value function, which leads to an improved trading strategy.

We completed further experiments to test the effect of a trading restriction on trading performance and see whether the RL trading policy can outperform the optimal myopic policy, as summarised in table 5.3.

Average Profit	**Trading Restriction**				
	0% ($\lambda = 2$)	**20% ($\lambda = 1.6$)**	**40% ($\lambda = 1.2$)**	**60% ($\lambda = 0.8$)**	**80% ($\lambda = 0.4$)**
Multi-action RL	0.16421	0.13027	0.12011	0.09209	0.0579
myopically optimal	0.16424	0.12945	0.11818	0.08922	0.05479
Difference	-0.00003	0.00082	0.00201	0.00287	0.00311
% difference	-	0.629%	1.607%	3.117%	5.371%
Std error (difference)	0.00065	0.00061	0.00077	0.00066	0.00075

Table 5.3 shows the average profit over 100 data sets each consisting of 1000 observations for RL and the myopically-optimal trading policy for different % trading restrictions.

The results show that, in the presence of a trading restriction, performance is significantly improved by using the multi-period RL policy compared to the myopically optimal policy. The larger the percentage trading restriction, the greater the path dependency, and so, as expected, the more significant the impact of the multi-period policy. In terms of investment performance, the RL policy with a trading restriction results in a small but significant increase in average reward. For example, in the case of a trading restriction of 60% the multi-action RL policy outperforms the trading performance of the myopically optimal policy by 3%. To highlight the difference between the myopic and RL trading policies we plot the trading policies, in terms of the change in the trading position against the predicted return, as shown in figure 5.16.

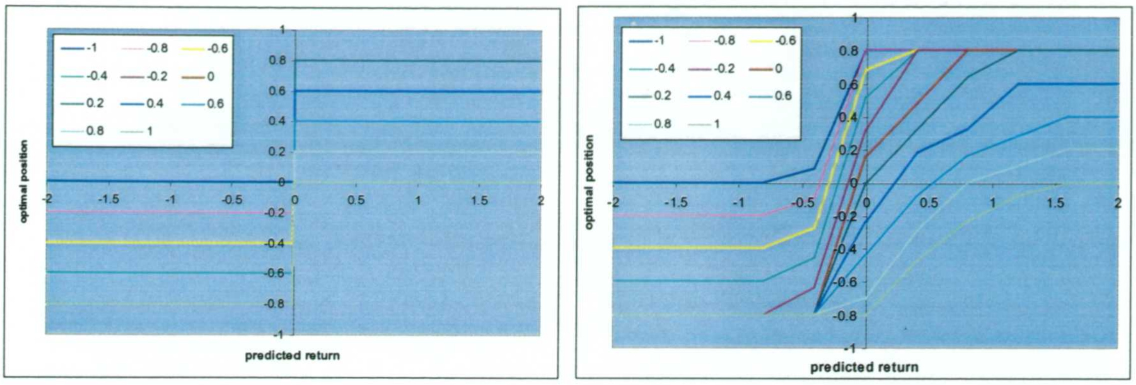


Figure 5.16 shows the optimal myopic policy (left) and multi-period RL trading policy (right) for a 60% trading restriction.

Figure 5.16 shows the myopic and RL trading policies in terms of the relative change in the trading position against predicted return for a 60% trading restriction. The left-hand graph shows how the optimal myopic policy shifts the trading position depending only on the sign of the predicted return. It takes no account of the future consequences of altering the trading position. In terms of bias-variance, this policy maximally exploits current predictability without considering the future cost of trading (i.e. small bias but large variance). The right-hand graph shows how the optimal position for the multi-period RL policy depends on both the previous trading position and the magnitude of the predicted return. The RL policy takes into account the future consequences of changing the trading position. Using RL to approximate the optimal trading policy can be considered equivalent to finding the optimal position on the bias-variance trade-off curve in order to maximise performance (see figure 5.9). Furthermore, we can illustrate the difference between the myopic and RL trading policy for 40 simulated time periods, as shown in figure 5.17.

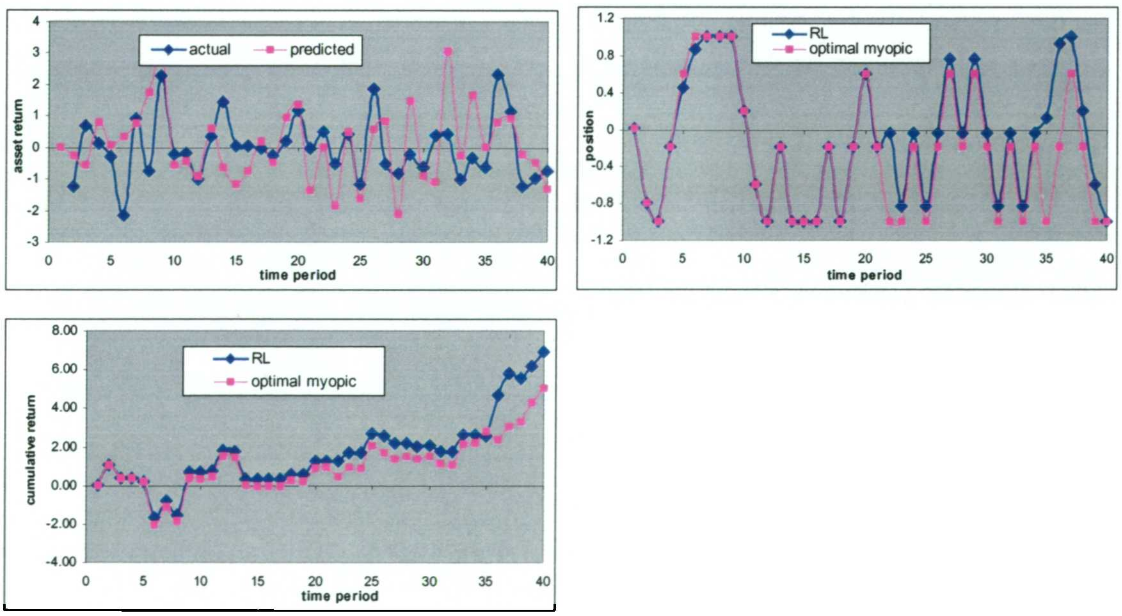


Figure 5.17 shows the synthetic asset returns (top panel), trading signals (top left) and cumulative sums of returns (bottom left).

Figure 5.17 compares the two trading systems over 40 simulation time periods. The top left and bottom left panels show how the two trading policies can lead to different trading policies and consequently different trading performance.

In this section we have demonstrated the potential for optimising trading strategies using RL. We have developed a novel learning algorithm, which we have shown speeds up the on-line learning process, by taking advantage of the partial independence between asset returns and trading decisions. In addition, we have parameterised the value function, using a neural network, to improve accuracy and generalisation. Experiments have been devised which highlight the merits of this approach by using a trading restriction to compare learning speed with standard Q-learning and performance with the myopically-optimal trading policy. These results illustrate the benefits of optimising a sequential decision policy using RL. Clearly, the RL algorithm with the selected parameters is not necessarily the only or optimal solution. We believe that these results set the scene for more specific experiments in which the model may be further improved and calibrated by examining other RL learning algorithms, neural network architectures, or network training parameters as well as considering other forecast model characteristics, trading performance criterion and transaction cost models. However, these simulations are computationally intensive and require many hours to complete each experiment. Thus, it is not possible to extensively validate these promising results within the scope of this thesis. In the next section we summarise results and compare our developed decision modelling techniques for optimising trading strategies of predicted returns.

5.4 Summary

In this chapter we have developed four decision modelling techniques for optimising a trading strategy: parameterised trading rules, path dependent trading rules, standard Q-learning and multi-action Q-learning. Each method has been developed given some underlying assumptions about the state and nature of the trading environment. Specifically, we have developed models based upon realistic but simplified assumptions of expected trading costs. We have shown that under a specific transaction cost model the developed techniques can outperform both heuristic trading rules and the myopically-optimal trading policy.

We can compare the relative merits of each modelling technique by considering the ability of each technique to control the trade-off between exploiting predictability and minimising expected trading costs. We base this comparison on three model attributes which are denoted by: flexible, adaptive and conditional. "Flexible" indicates that the model has the ability to optimise the functional form of the trading policy given different predictive characteristics. "Adaptive" indicates the ability of the model to respond to a dynamically changing trading environment. "Conditional" indicates the ability of the model to accommodate additional state variables, which influence trading performance (i.e. act as conditioning factors). Table 5.4 presents an overview of each methodology for the three attributes.

	Flexible	**Adaptive**	**Conditional**
Heuristic Trading Rule	No	No	No
Parameterised Rule	Yes, but constrained	No	No
Path dependent rule	Yes, but constrained	No	Only previous position
Standard RL	Yes, but may overfit	Yes, but may be too slow to learn	Yes
Multi-action RL	Yes, but may overfit	Yes	Yes

Table 5.4 shows the different modelling methodologies in terms of the attributes that they provide for modelling the trading policy.

Table 5.4 indicates the advantages of each decision modelling technique, which typically depend on the nature of the trading environment. For example, models with these attributes offer limited performance improvements for low volume, low frequency trading in well-established markets with small transaction costs. If the investment context, however, involves high frequency trading with significant transaction costs, then models with "flexible" and "conditional" attributes, such as path dependent trading rules, may offer considerable performance benefits. Furthermore, if trading takes place in a dynamic trading environment, then models with "adaptive" attributes that can adjust quickly to changing market conditions, such as multi-action RL, may offer further performance enhancements.

6 Conditional Optimisation of a Forecasting Model

In chapter 5 we developed the first stage of our methodology, which involved developing a decision model to conditionally optimise a trading strategy given predictions derived from a statistical forecasting model. We developed two methods, one using a class of parameterised trading rules and the other using a reinforcement learning algorithm, for performing the trading task in the presence of different trading conditions. In this chapter we develop the second stage of our methodology which involves optimising a forecasting model conditionally on a given trading strategy. We approach this problem by considering an optimisation procedure to control the "model design factors" that are specific to a modelling strategy. Examples of these factors include forecast horizon, optimisation criterion and forecast object. For our trading system, we consider that these design factors can influence the economic value of the forecasting model in the trading environment, as highlighted in section 4.8. In this chapter we consider the control of typical design factors using meta parameters, which provide a means of conditionally optimising the forecasting model given a particular trading strategy. In chapter 7 we incorporate this second stage into a methodology to perform a joint optimisation over both forecasting and decision stages, thus allowing us to "globally" maximise the trading performance of the combined system.

In section 6.1 we provide an overview of the process of conditionally optimising a forecasting model given a trading strategy and preview procedures to complete this task. In section 6.2 we discuss the general relationship between the economic value of a trading strategy and the forecast model. In section 6.3 we illustrate, using a synthetic example, how we can control the optimisation criterion of a forecasting model with two key characteristics, namely predictive correlation and prediction autocorrelation, given a trading strategy. In section 6.4 we discuss model design factors and suggest procedures for optimising these factors based on *meta* parameters. In section 6.6 we use simulation experiments to investigate the effect of the forecast horizon for a particular trading strategy. Our results demonstrate that for a trading system, with all other factors constant, the optimal forecast horizon depends on the degree of predictive accuracy of the forecasting model and the level of transaction costs.

6.1 Overview

In general, forecasts are made and have economic value because they provide predictive information to improve decision-making. Consequently, the effective design, use and evaluation of a forecasting model will ultimately require information of the associated decision-making task. In most practical situations this process is achieved by a qualitative assessment of the task, which is usually adequate for most applications. However, in the case of trading strategies for predictive models the choice of these design factors may not be obvious and so an *ad hoc* approach may not be adequate. This is because trading systems typically exhibit a number of properties that are uncommon in other applications. For example, the highly stochastic nature of financial markets limits any detectable predictive ability to relatively low levels and the complexity of financial systems means that forecasting models tend to weak *a priori* assumptions. In addition, the profitability of trading systems for forecasting models tends to be high sensitive to forecast accuracy. The combination of these factors means that naïve implementation of the forecasting model may indirectly lead to sub-optimal trading performance.

Thus, for trading systems, a more systematic approach seems appropriate in situations where the design of a forecasting model can be optimised with respect to a particular trading strategy. In this chapter we develop a methodology to solve this problem using a *multi-objective* optimisation procedure that is not part of the traditional approach to constructing forecasting models. This is implemented by taking a "higher level" approach to forecast model construction, which focuses on controlling the design of the model rather than the "micro-level" optimisation of the model parameters. This allows us to focus on the generic properties of a model rather than specific methodological issues by assuming that a generic forecasting model can be described by a set of model design factors. We propose that, in general, financial forecasting models can be characterised by a number of such attributes that together determine the economic value of the model. Examples of such "design factors" include the object of the forecast (e.g. the specific time series), the forecast type (i.e. point, interval or density forecast), the forecast horizon, the model optimisation criteria, and the candidate input variable set. From our perspective, construction of the forecasting model can then be abstracted away from the specific forecasting method so that predictions can be simulated from a set of model characteristics. This procedure allows us to focus on the modelling strategy rather than the model building process itself and so develop a more generic methodology for constructing a conditional forecasting model given a trading strategy. This approach is an extension of the traditional modelling techniques applied financial forecasting which are typically used to construct models purely on the basis of maximising predictive accuracy (e.g. Refenes, 1995).

This component of our methodology is referred to as stage 2 as it is assumed that an initial forecasting model has been built and stage 1 of our methodology is completed, which involves conditionally optimising the trading strategy given predicted returns from the forecasting model.

Figure 6.1 shows conceptually the process of re-optimising the forecasting model conditional on the trading strategy. The feedback loop from the trading strategy indicates the process of re-optimising the forecasting model after the trading strategy has been optimised.

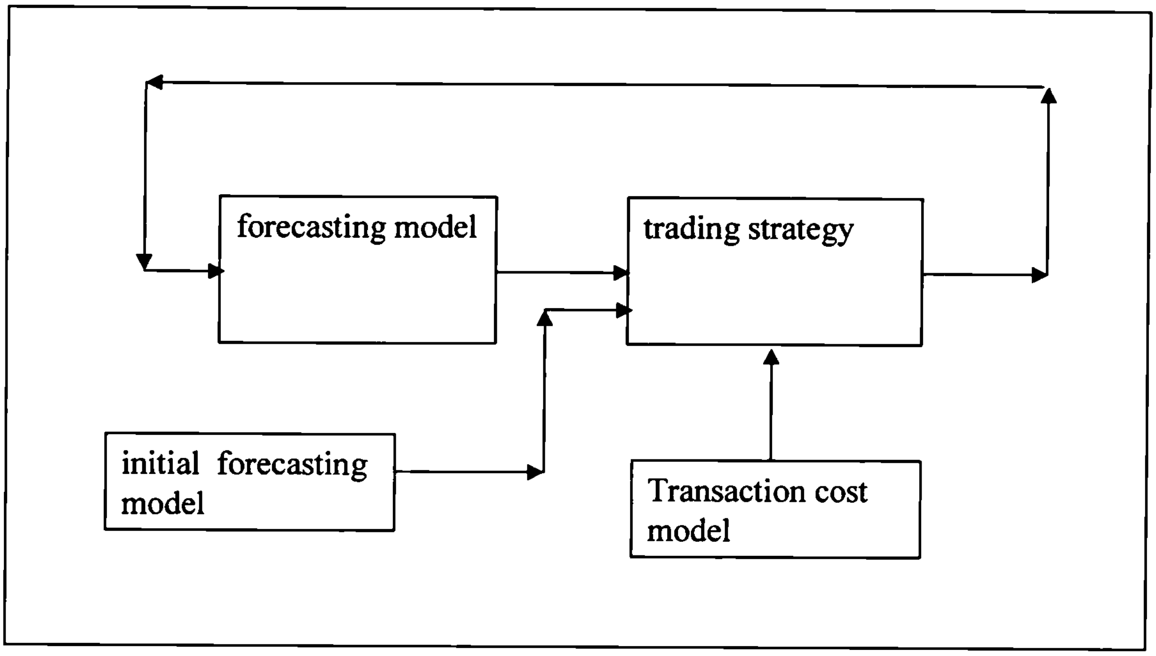


Figure 6.1 depicts the process of conditionally optimising a forecasting model for a given trading strategy.

In this scheme, the construction of the initial forecasting model arises from some conventional approach to forecast model construction. The first stage of our methodology, as described in chapter 5, constructs a model to optimise the trading strategy given the predicted returns from this initial forecasting model and some model of the expected trading costs that will be incurred from trading. After optimisation of the trading strategy, the forecasting model may need to be re-optimised to take account of the subsequent trading strategy and it is this process which is referred to as the second stage of the methodology. A further development of the methodology is that these two stages may be iterated in turn to perform the joint optimisation of the trading strategy and the forecasting model, which is discussed in chapter 7.

Before examining this topic in detail we summarise the content of the remaining sections of this chapter. In section 6.2 we focus on the relationship between a forecast model and the economic value of trading predictability. We argue that if a forecasting model is defined by a single characteristic related to prediction accuracy then we can directly estimate the optimal trading strategy. This explains the lack of research into multi-criteria optimisation methods for forecasting models, as at least an implicit relationship is assumed between the quality of the forecasting model and the performance of any optimal trading policy. In the context of trading systems, this traditional approach assumes that improvements in forecast accuracy will always provide the best predictive information. This implies that forecasting methodologies do not need any direct reference to the associated decision making. In the conventional approach, the forecasting model requires no information feedback from the decision-making application.

In section 6.3 we demonstrate how conditional optimisation may be performed on an artificial decision task that is designed to be similar to a trading system. We simulate the output from a generic forecasting model with two characteristics that are assumed to be a function of the model parameters. This assumption is based on the evidence in section 4.3 that trading performance can be influenced by the two predictive characteristics of predictive accuracy and prediction smoothness. We develop a procedure to optimise these two characteristics, using an additional control parameter, which controls their relative importance during construction of the forecasting model.

In section 6.4 we consider more general design factors that may influence trading performance. We discuss the nature of design factors and argue that they cannot be simply optimised independent of the trading strategy on the basis of forecast accuracy. We discuss three factors in detail (i.e. forecast object, forecast horizon and optimisation criterion) and argue that each factor involves a number of decisions or inferences that influence the forecast model. In this section we suggest procedures to optimise these design factors for a given trading strategy using a meta parameter approach. These factors are considered to equally apply to the wide range of modelling techniques, which were described in chapter 3.

In section 6.5 we develop further simulation experiments to investigate the effect of the forecast horizon on trading performance in the presence of transaction costs. Our results demonstrate that, for a trading system with all other factors fixed, the optimal forecast horizon is dependent on the prediction accuracy of the forecasting model and the level of transaction costs. Further simulation experiments demonstrate how the forecast horizon can be optimised using the Trisection search method.

6.2 Optimising a Forecasting Model for Trading

In the previous section we argued that the construction of a financial forecasting model typically involves not only the selection of the forecasting method but also control of "design factors" that can have a significant bearing on the economic value of the model. For example, the selection or construction of the target time series is considered to be a major factor in developing a forecasting model with significant predictive power (Lo and MacKinlay, 1997;

Burgess, 1999). In addition, other factors such as the time horizon or the model optimisation criteria have also been identified as factors that can significantly effect the usefulness of a forecasting model (Diebold, 1998).

It is from the perspective of design factors that we consider the development of a forecasting model for trading. From this view, the purpose of the forecasting methodology is to construct a model that produces predictions with characteristics which maximise the performance of the optimal trading strategy. Thus, the general relationship between the forecasting model and economic value of the trading strategy can be described in terms of predictive characteristics.

Let us assume that our trading system has a forecasting model that produces predictions with a number of characteristics, denoted by C_1, C_2,...C_n. In general, the relationship between the economic value of the forecasting model and the predictive characteristics takes the form:

$$E[U \mid \pi^*] = f(C_1, C_2, ... C_n) \tag{6.1}$$

where 'f' is some unknown function of the vector of model characteristics, C, U is the investment utility and π^* is the optimal trading strategy.

This equation indicates that there is a direct link between the characteristics and the economic value of the forecasting model. This indicates that any optimisation of these predictive characteristics should be done with respect to the optimal trading strategy. Thus, if the true optimal trading strategy is known, it should be possible to quantify the definition of the "best" forecasting model by computing the associated trading performance. Unfortunately the true optimal policy is, in general, unknown and can only be approximated in the context of a particular forecasting model. This implies that optimisation of the forecasting model ought be completed conditionally upon the optimal trading strategy. The process of estimating the optimal forecasting model and trading strategy, which together gives maximal performance, can only be achieved by some form of global optimisation over both the two modelling stages.

If we assume that only one characteristic exists for the optimisation criterion then the traditional approach to forecasting model construction fits within the scope of equation (6.1). The expected value of the optimal trading policy is then some *monotonically increasing* function of predictive accuracy. If we could define the function f then for a certain level of predictive ability we could compute the expected value of the optimal trading strategy. It would also provide the additional benefit of knowing the value of any subsequent improvement in predictability of the forecasting model. This may be useful in deciding what level of resources to dedicate to improving a particular forecasting model.

Furthermore, with just one characteristic, the lower the forecast error the higher the economic value of the forecasting model. The forecasting model can then be optimised before optimisation of the trading strategy under the assumption that minimising forecast error will provide the best information to the trading strategy. In this way, the two sequential modelling stages automatically implement a joint optimisation over forecasting and trading without explicitly optimising the forecasting model conditionally on the trading strategy. However, if multiple characteristics of the forecasting model affect trading performance then the forecasting model cannot be optimised in isolation from the trading strategy. The optimisation of the forecasting model needs to reflect the impact of each characteristic to trading performance, and this cannot be computed without optimisation of the trading strategy.

We propose that this process be implemented using "higher level" design parameters to control the modelling strategy and some information feedback from the trading strategy to re-optimise the forecasting model. The control of these higher level parameters needs to reflect the value of the forecasting model with respect to the trading strategy and this can only be done after the trading strategy has been optimised. In this chapter we develop a method of controlling the characteristics of a forecasting model using special parameters, referred to as *meta* parameters, that allow the forecasting model to be optimised with respect to the trading strategy.

In the next section we develop a synthetic experiment for a decision system which is influenced by two predictive characteristics. We demonstrate how this system can be optimised using a meta parameter, which links control of the model design with the "micro level" optimisation of the forecasting model parameters.

6.3 Illustration: Optimising Two Predictive Characteristics

In the section we illustrate the dilemma facing the optimisation of a forecasting model for trading and show how we may optimise a model with two characteristics conditionally on decision-making. This is considered to be analogous to a trading system for a forecasting model with two predictive characteristics which both influence trading performance, as described in section 4.4.

We investigate this trading system by devising a forecasting model with two different characteristics that both influence decision performance. The characteristics were assumed to have values directly related to the underlying model parameters. Thus, simulating the two characteristics emulates a generic forecasting model whose predictions are described by two characteristics that are determined by the value of the model parameters. In the context of a trading system, these predictive characteristics could typically represent predictive correlation (accuracy) and prediction autocorrelation (smoothness), as described in section 4.4.

More specifically, assume that a forecasting model consists of two parameters, θ_1 and θ_2, and has two important predictive characteristics, C_1 and C_2, which take the form:

$$C_1 = g_1(\theta_1, \theta_2) \quad \text{and} \quad C_2 = g_2(\theta_1, \theta_2) \tag{6.2}$$

where 'g_1' and 'g_2' are some functions of the two model parameters, θ_1 and θ_2.

This specification provides a means of simulating a forecasting model without actually assembling data and producing a specific model or even simulating the predictions from a forecasting model. The only requirement is to simulate the two functions in equation (6.2) which describe the two model characteristics within the 2-D model parameter space. To do this we devised a function describing characteristics that gradually deteriorate around some optimal value in each direction of parameter space, which takes the form:

$$g_i = \max\left[0, 1 - \left(\frac{\theta_1 - x_i}{\sigma_{x_i}}\right)^2 + \left(\frac{\theta_2 - y_i}{\sigma_{y_i}}\right)^2\right] \tag{6.3}$$

where x and y are the co-ordinates of the centre of the distribution and parameters, σ_x and σ_y control the spread of the distribution in each direction.

In equation (6.3) the function 'g' is restricted within the range zero to one, taking the value one at the central co-ordinates, x and y. The inclusion of the two spread parameters, σ_x and σ_y allows some flexibility in the shape of the distribution. To investigate the potential effects of the two model characteristics on decision-making we select function parameters, as given in table 6.1, and shown graphically in figure 6.2.

	X	Y	σ_x	σ_y
Characteristic 1	0.2	0.3	0.5	0.3
Characteristic 2	0.6	0.5	0.5	0.5

Table 6.1 shows the function parameters for equation 6.3 to simulate two model characteristics.

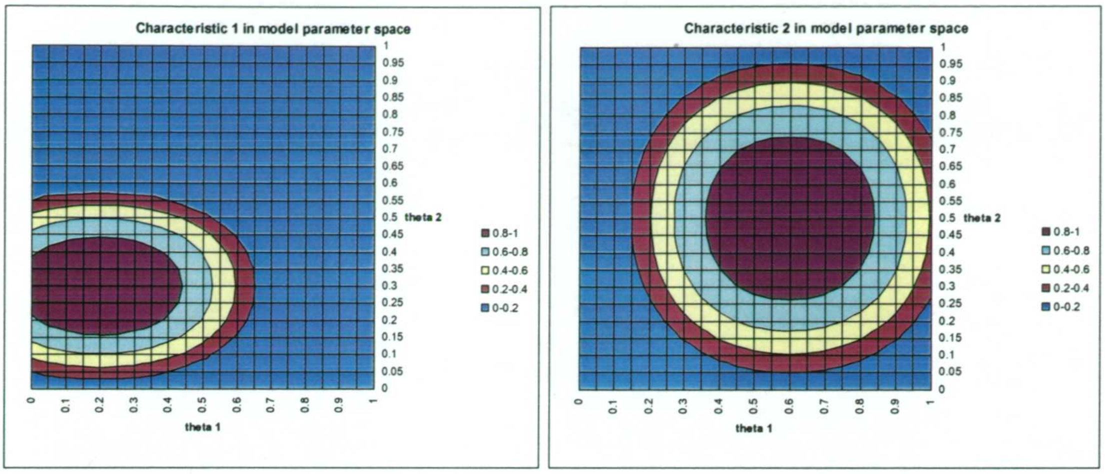


Figure 6.2 shows two model characteristics as a function of the model parameters, θ_1 and θ_2.

The key feature of the two distributions is that they overlap and that the parameters, θ_1 and θ_2, that maximise the characteristics C_1 and C_2 are different. If the two distributions do not overlap then the best forecasting model just depends on which characteristic has most value to the associated decision making. Generating overlapping distributions allows the possibility of a trade-off between the two characteristics. In addition, as the characteristics have different distribution centres in parameter space, the model parameters cannot be simply modified to maximise both characteristics simultaneously. Thus, optimisation requires some associated value of two characteristics to decision-making, in order to determine the optimal model parameters.

To optimise the forecasting model we ideally need to know the relationship between the parameters of the forecasting model and decision-making. This relationship, however, can only be considered once the forecasting model has been optimised so at this stage we cannot optimise the parameters for the best "application specific" forecasting model. However, we can formalise the specification of the optimal model parameters, θ^*, as the parameters that maximise some weighted combination of the two model characteristics, C_1 and C_2, which are directly related to θ_1 and θ_2, by the functions 'g_1'and 'g_2'. This expression describing the optimal parameters is defined as

$$\theta^* = \arg\max_{\theta} \left[\alpha C_1(\theta_1,\theta_2) + (1-\alpha) C_2(\theta_1,\theta_2) \mid \alpha \right] \qquad (6.4)$$

where α, the control parameter, is some given value between 0 and 1.

In equation (6.4) the control parameter, α, to optimise the linear combination of the two model characteristics cannot itself be optimised without some information about the value of the characteristics to the associated decision making. This illustrates the need for some information feedback from decision-making to optimise the forecasting model conditionally upon the decision policy. This process can be interpreted as a local linearisation of some decision performance criterion, U, with respect to model characteristics, so that $\alpha = \partial U / \partial C_1$ and $1 - \alpha = \partial U / \partial C_2$.

In order to associate some value with the model characteristics we need some knowledge of the relationship between decision making and forecasting, as described in equation (6.1). In general this cannot be defined without approximating the optimal decision policy and accurately specifying the model characteristics that influence decision-making. In this artificial decision system, however, the decision modelling process can be simplified by assuming that the performance criterion of the optimal decision policy is simply defined as the multiplication of the two model characteristics and so takes the form:

$$E\left[U \mid \pi^*\right] = f(C_1, C_2) = C_1 * C_2 \tag{6.5}$$

In the context of a trading strategy, we can justify this model of the performance criterion, given in equation (6.5), by saying that expected trading profit is described by the prediction accuracy multiplied by smoothness of forecasts. In this case, a forecasting model with positive prediction accuracy but zero smoothness would generate zero profits as these are wiped out by transaction costs. Likewise, a model with no prediction accuracy but high smoothness would also produce zero profits. It can be argued that, in some trading conditions, profitability requires a forecasting model with both prediction accuracy and prediction smoothness. It should be noted that although this relationship is plausible, it is unlikely, in most trading environments, that this relationship can be reduced to the simple multiplication of these two factors.

We have now defined a simple synthetic decision problem that is dependent on two model characteristics, which are defined with respect to the model parameters, as described in figure 6.2 and the expected performance in equation (6.5). For this example, we can show how utility changes with respect to the control parameter, by selecting a range of values between zero and one and optimising the forecast model parameters, using equation (6.4) as shown below:

α	θ_1*	θ_2*	C_1	C_2	Utility
0	0.6	0.5	0.000	1.000	0.000
0.1	0.55	0.45	0.134	0.994	0.133
0.2	0.5	0.4	0.457	0.965	0.441
0.3	0.45	0.4	0.565	0.923	0.521
0.4	0.45	0.35	0.693	0.900	0.624
0.5	0.4	0.35	0.778	0.838	0.651
0.6	0.35	0.35	0.840	0.753	0.633
0.7	0.35	0.35	0.840	0.753	0.633
0.8	0.3	0.3	0.952	0.604	0.575
0.9	0.25	0.3	0.974	0.473	0.461
1	0.2	0.3	1.000	0.317	0.317

Table 6.2 shows the optimal model parameters, θ_1* and θ_2*, the associated characteristics, C_1 and C_2, and the utility for the range of values of the control parameter, α.

Table (6.2) shows the values of the optimised forecasting model parameters, characteristics and utility for a range of values of the control parameter, α. Results show that when α equals 0 the optimal forecasting model has parameters (0.6,0.5) which maximise C_2 but has a utility of zero. For the other extreme value of the control parameter (i.e. α equal to 1) the optimal forecasting model has parameters (0.2,0.3), which maximises C_1 with decision utility of 0.317. This is due to the non-zero value of C_2. If the optimisation process just considered these two characteristics individually, then it would make sense to just select the model that maximise C_1, which represent prediction accuracy. This is typical of the traditional approach to forecast model construction which minimises prediction error. However, by inspecting the values of utility it is clear that a higher value of utility can be achieved through selecting a value of α less than one, as shown in figure 6.3.

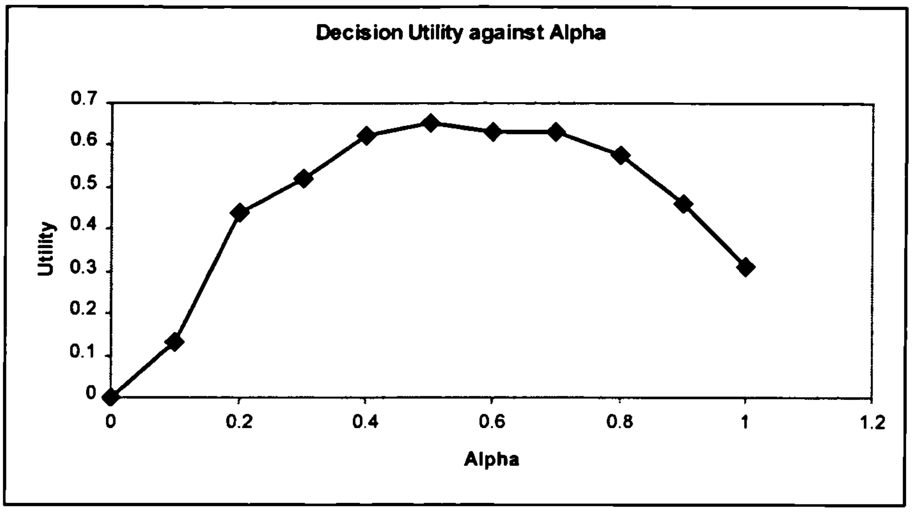


Figure 6.3 shows the decision utility against the control parameter, α.

Figure 6.3 shows that, in this case, the optimal decision performance of 0.65 is achieved when α equals 0.5. The optimal forecasting model, in terms of decision performance, is thus an equal trade off between the two model characteristics. Most traditional approaches to constructing forecasting models would only optimise one forecast model characteristic and so, in this example, optimisation of the first model characteristic (α=1) would result in a performance of 0.32, approximately 50% below of optimum value of 0.65. Our results above indicate that conditional optimisation can significantly improve performance when decision making is influenced by two model characteristics.

This illustration shows how the optimal parameter for a forecasting model may be conditional on decision making. This has been shown to occur when a forecasting model contains two or more characteristics that influence decision-making. From this illustration, we can infer that systems developed for trading predictability may require a methodology to optimise over both forecasting and decision-making in order to globally optimise some utility function. The illustration shows that the optimisation of a forecasting model may require a trade-off between model characteristics. We have shown that this may be controlled using an additional parameter, α. The control of the trade-off between the two model characteristics requires some feedback from the trading strategy to the construction of the forecasting model. This is considered a more systematic approach to model construction than taking an *ad hoc* approach based upon some predefined optimisation criteria.

In this illustration we used a grid-based search procedure to approximate the optimal value of the control parameter. In more practical problems, which require construction of specific forecasting and decision models, more efficient search procedures may be more appropriate. A simple gradient-based directed search algorithm could be implemented to find the optimal control parameter. This requires estimation of the derivative of utility with respect to the single control parameter, α, which takes the form:

$$\alpha_{n+1} = \alpha_n + \rho \frac{dU}{d\alpha} \tag{6.6}$$

where ρ is the step size.

In equation (6.6) the derivative can be estimated by perturbing α and re-estimating the forecasting and decision models. Other approaches may be employed that use modern optimisation techniques based upon heuristic methods such as Genetic Algorithms (Holland, 1975), Simulated Annealing (Metropolis et al., 1953) do not rely on calculating derivatives and may be appropriate to find approximate solutions in optimisation problems with large, non-linear control parameter spaces. In chapter 7 we discuss these issues when we combine the two stages of the methodology to perform a joint optimisation over a forecasting model and a decision model.

In the next section we apply the methodology developed in this section, for conditionally optimising a forecasting model, to a more realistic example. In this next experiment we construct a forecasting model for an identified mean-reverting relative price series. The purpose is to consider whether the conditional optimisation of the forecasting model, for a given trading strategy, improves trading performance in the presence of transaction costs. It should be noted that the purpose is not to evaluate the profitability of trading predictability in financial markets, which is considered later in chapter 8. We implement the procedure using the gradient-based method, described in equation (6.6), to optimise of the parameter, α, which controls the trade-off between predictive correlation and prediction autocorrelation.

6.3.1 Example: Exploiting a mean-reverting time series

In this sub-section we consider a more realistic trading system consisting of a forecasting model for a relative price series which exhibits some mean-reverting behaviour. We exploit this form of predictable behaviour by constructing an exponential smoothing model to forecast the next value of the time series. We assume that the predicted returns have two predictive characteristics, measured in terms of predictive correlation and prediction autocorrelation, which both can influence trading performance. We control the optimisation criterion of the forecasting model using the meta parameter α, which is described in equation 6.4. This enables us to conditionally optimise the forecasting model for a predefined trading strategy.

For this example, we constructed a relative price series which was defined as the ratio of the two equity indices, DAX (Germany) and IBEX (Spain) for daily closing prices over a 500 day period. It was found that this relative price series exhibited some potentially predictable behaviour in the form of mean reversion of the time series (see figure 6.4). Some practitioners involved in "pairs" trading use this simple form of relative value analysis. In chapter 8 we describe a more rigorous statistical approach to a form of relative value trading, which we refer to as statistical arbitrage.

Given the mean reverting time series, we developed a prediction of the relative price series by constructing a forecasting model based on simple exponential smoothing, which is defined as

$$\hat{y}_{t+1}(\theta) = \theta y_t + (1-\theta)\hat{y}_t \tag{6.7}$$

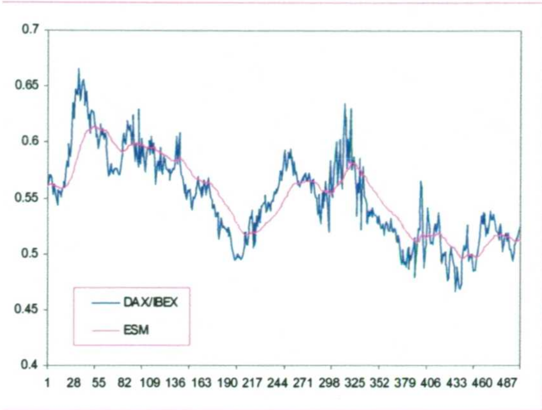
where y is the relative price series, $\hat{y}_t$ is the forecast value and θ is the model parameter.

For this forecasting model we assume that the two predictive characteristics of predictive accuracy and prediction smoothness both influence trading performance. In this case, we want the optimise the parameter, θ, with respect to the two predictive characteristics. We therefore, design the optimisation criterion for the forecasting model using a meta parameter to determine the relative importance of the two characteristics. More specifically, the function, F describing the optimisation criterion is defined as

$$F = \alpha\beta + (1-\alpha)\rho \tag{6.8}$$

where β is the measure of predictive accuracy and ρ the measure of prediction smoothness.

In this way α acts to control the characteristics of the forecasting model by defining the desired trade-off between the two characteristics. We can measure the two characteristics for predictive accuracy and prediction smoothness using predictive correlation and prediction autocorrelation respectively. Figure 6.4 shows the relative price series, an example of the exponential smoothing model (ESM) and also the two forecast model characteristics as a function θ.



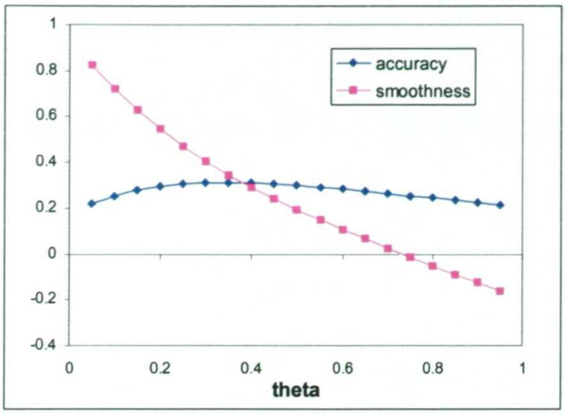


Figure 6.4: the left hand graph shows the actual and predicted asset price, and the right hand graph shows the forecast characteristics as a function of the model parameter, θ.

The right hand graph shows that two characteristics are both effected by value of θ. The maximum predictive correlation of 0.31 is achieved when $\theta = 0.4$ and the maximum prediction autocorrelation when $\theta = 0$. However this still does not determine the value of θ for the optimal forecasting model. To do this, we need some determine the trading strategy for the predictions and develop a procedure to optimise α.

For this trading system, we define the trading strategy on the basis of the "linearly proportional" trading rule, as defined in equation (5.2), with parameter, $k = 1$. This trading strategy exploits the mean-reverting behaviour by taking a trading position based on the size of the difference between the predicted price series the value of the previous price. For these experiments we defined trading performance in terms of Sharpe Ratio, as defined in section 4.6.

Given the forecasting model and the trading strategy, the next step is to optimise the meta parameter, α. This can be achieved using the gradient based search algorithm defined in equation (6.6). This optimisation process is specified with step length parameter, $\rho = 0.1$ and an initial forecasting model with $\alpha = 1$, which represents a model optimisation criterion that only maximises predictive correlation and is considered to represent the traditional optimisation criterion for forecasting models. For the initial experiment, the transaction cost parameter was set to a rate of 0.5%. At each iteration, α is updated, using equation (6.6), with perturbation analysis used to estimate the derivative of the Sharpe Ratio with respect to α. A summary of the results from the optimisation process is shown graphically in figure 6.5.

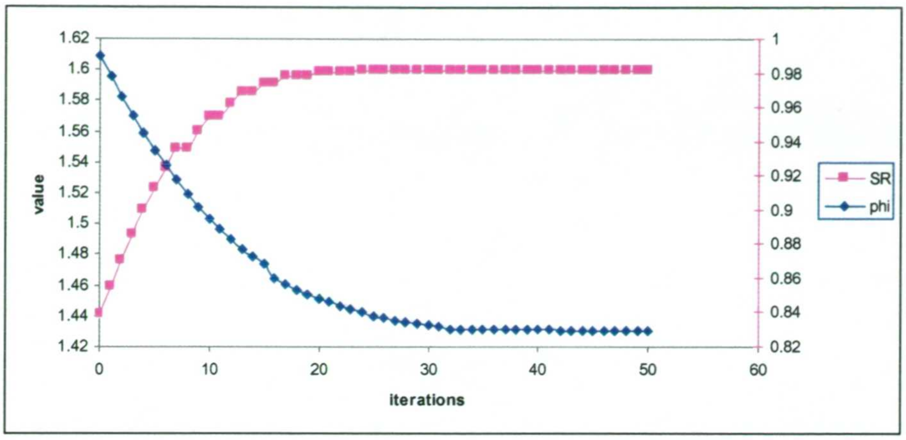


Figure 6.5 shows the value of the Sharpe Ratio and the meta parameter α, over 50 iterations of the update rule, which is described in equations (6.4) and (6.6).

Figure 6.5 shows that the Sharpe Ratio converges to a maximum after 25 iterations and the value of the meta parameter falls to 0.83 during the optimisation process. This shows that the predictive autocorrelation is a significant model characteristic and that re-optimisation of the forecasting model improves performance by 11%. A further set of experiments were completed to show the influence of the level of the assumed transaction costs. The transaction cost parameter was varied and the optimisation procedure repeated with a summary of the results given in table 2. The table reports the optimal values of the meta parameter, $\alpha*$, forecasting model parameter, $\theta*$, predictive correlation $\beta*$, prediction autocorrelation, $p*$, and the initial and optimal Sharpe Ratio.

cost(%)	α*	θ*	β*	ρ*	SR(α*)	SR(0)	ΔSR
0.0	1.00	0.400	0.310	0.300	3.3577	3.3577	0.0000
0.1	0.971	0.320	0.308	0.374	2.9834	2.9833	0.0001
0.2	0.918	0.270	0.305	0.438	2.6362	2.5959	0.0403
0.3	0.918	0.270	0.305	0.438	2.2715	2.2126	0.0589
0.4	0.866	0.210	0.294	0.524	1.9354	1.8272	0.1083
0.5	0.840	0.180	0.286	0.572	1.5995	1.4404	0.1591
0.6	0.621	0.010	0.181	0.915	1.4865	1.0528	0.4337
0.7	0.584	0.010	0.181	0.915	1.2959	0.6652	0.6307
0.8	0.732	0.050	0.218	0.821	0.8748	0.2783	0.5965
0.9	0.623	0.010	0.181	0.915	0.9142	-0.1072	1.0215
1.0	0.633	0.010	0.181	0.915	0.7233	-0.4908	1.2142

Table 6.3 shows the optimal parameters, model characteristics and performance measures for different levels of transaction cost.

These results show that trading system performance, SR(α*), reduces with higher transaction costs, as expected. However, the optimisation of α* increases performance gains (ΔSR) for transaction costs above 0.1%. These gains are made as a result of decreasing the optimal meta parameter, α*, which boosts predictive autocorrelation, ρ*, at the expense of predictive correlation, β*. This change in trade-off alters the optimised forecast parameter, θ.

The performance improvements can be explained in terms of the trading position and the two predictive characteristics. Increasing prediction smoothness has the effect of improving stability of the trading position through time thereby reducing the overall transaction costs. Under conditions of high transaction costs it is preferable to *trade off* predictive correlation for greater predictive smoothness. This explanation is analogous to the bias/variance trade-off commonly found in statistics where, in this case, "bias" refers to trading position stability induced by predictive smoothness, and "variance" refers to the exploitation of predictive correlation through varying the trading position. This analogy is used in section 5.2.4 to explain the performance improvements of path dependent trading rules. Under this same analogy, the meta parameter α is used to find the optimal position on the bias-variance curve and so maximise trading system performance. In the next section we discuss more general design factors for forecasting models which may influence decision making, especially focusing on the context of developing trading strategies for financial forecasting models.

6.4 The Design and Control of Model Development

In general, the building a statistical forecasting model does not involve simply following some automatic modelling procedure or applying some mechanised methodology but also the selection of some general design factors that affect model development. In the context of trading, we have shown, through illustration and simplified examples, that the naïve implementation of a forecasting model, for instance, to solely maximise prediction accuracy, may lead to sub-optimal trading performance. This indicates the importance of considering the optimisation of a forecasting model in the light of the wider system design. In this section we discuss typical design factors that control the development of a forecasting model and show how these may influence the value of the model in a decision environment. We suggest that the features of the decision environment in which the predictions are applied have implications to the design, use and evaluation of a forecasting model and that control of these model design factors can improve the value of the forecasting model for trading.

In general, the construction of a forecasting model involves a sequence of stages, which can be broken down as follows:

- identify intended purpose of the forecast
- determine time period and frequency of the forecast
- select the appropriate forecasting technique
- collect the appropriate historical data
- optimise model parameters and produce forecasts
- evaluate the reliability and suitability of the forecasts
- monitor the accuracy of the forecasts

The first stage is generally to assess the purpose of the forecast, which provides context to the subsequent stages and pinpoints the use of and reason for the forecasting model. In many applications the development of the forecasting model is due to the increasing complexity of a system within which a decision maker must operate, together with changing demands and expectations that drive the need to establish some view as to future values of key variables. The purpose of the forecasting model is therefore to reduce the uncertainty, by generating information that may help assess the consequences of decisions and evaluate the consequences of alternatives. In the context of trading, clearly the purpose of the forecasting model is to maximise predictive information in order to assist trading decisions and so improve an investment's risk adjusted return. Once the key variables have been identified the time period for decision making and the frequency of the forecast need to be selected. For most applications this is typically pre-defined by the nature of the decision-making task, however, for trading greater flexibility is often possible. Next, the forecasting technique is selected depending on the nature and the expected dynamics of the identified target variables. A range of forecasting techniques that can be applied to predict asset returns are described in chapter 3.

Once the modelling methodology has been selected, the appropriate historical data needs to be collected. This stage is particularly problematic for methods that allow the inclusion of exogenous variables. In investment finance, the prediction of asset returns can include a vast number of candidate variables due to the complexity of the financial system and the lack of understanding of the underlying system dynamics. Automatic variable selection methods cannot reliably deal with very large numbers of variables, so typically, variable selection requires the assistance of a market expert who can guide the variable selection process.

At this point, the forecasting technique can be applied to the selected historical data in order to produce the forecasts. The forecasts can then be evaluated for reliability and suitability within the decision environment for assisting the selection of alternative decisions. The consequence of the forecasts can then be assessed in terms of their value to decision making. In trading, the decision environment typically includes a trading strategy, which may involve optimising a decision model. This creates another layer of complexity that influences the suitability of the forecasts. Finally, most forecasting models require monitoring in order to detect any model breakdown or structural problems within the model and may include re-optimising model parameters given more recent historical data or redevelopment of the model.

During this modelling process, a number of design factors govern the construction of the forecasting model. In the context of optimising trading strategies for forecasting models, we consider these design factors to be "higher level" parameters that not only control model construction but also influence the value of forecasts within the decision environment. In the remainder of this section, we discuss some of these factors, including the object of the forecast (e.g. the specific time series), forecast type (i.e. point, interval or density forecast), forecast horizon and model optimisation criterion. Each of the design factors involves a number of decisions or inferences that influence the selection and optimisation of the forecast model and can be controlled during model construction.

6.4.1 Forecast object and type

The forecast object and type describe the choice of the predicted variable (i.e. target time series) and the information content of each individual prediction respectively. The object is usually determined by the variables that influence the consequences of alternative decisions. The forecast type may be a point, a prediction interval or a density forecast. A point forecast is a single number which typically represents the expected mean or a percentile of the next value. An interval forecast is a range of values in which we expect the realised value to fall and is normally referred to as a confidence interval. A density forecast is an entire expected probability distribution. Each type of forecast has implications to the choice of modelling methodology and the information available to assist decision making. The choice of forecast type is normally dependent on the amount of available data and the role of the forecasts within the decision environment.

In the context of trading predictability in financial markets, we have assumed that the prediction of the asset is a point forecast of the expected return. However, we have not assumed that the forecast object is a particular asset, in fact, we have considered a generic predicted return, which may apply to a wide range of financial instruments. Clearly, in practice, the forecast object should be directly related to a selected portfolio of assets in order to provide predictions of future asset returns. In investment finance, typically the choice of forecast object is designated by the style of a particular investment fund or trading desk. This determines the assets that can be traded for any trading strategy and is often dependent on issues related to portfolio diversification or asset correlation. However, flexibility often exists in how the forecast object is formed from the candidate assets. For example, if a short history of data exists for a particular asset then it may be appropriate to model a pseudo asset return that closely matches the nature of the instrument of interest but has a much more extensive source of historical data. This may enable the construction of a forecasting model with much higher predictability, which overcomes any potential mismatch between instrument behaviour. From another perspective it may be appropriate to consider predicting the asset return of fixed portfolios rather an individual assets.

It is interesting to note that in recent year, some of the most convincing evidence for predictability in asset returns has been derived from a forecast object that is constructed from a weighted combination of assets. The design of the forecast object then involves developing a method of constructing combinations of assets whose time series contains some predictability. This has advantages over financial forecasting models for the returns of single assets, which are likely to be more market efficient, and so offer less potential predictive behaviour. This approach to identifying predictability in portfolios of assets is considered in chapter 8 where we investigate the application of statistical arbitrage trading. In this application, combinations of assets are constructed and tested to identify significant statistical mispricings between a group of assets. For this process multivariate techniques can be used, such as principal component analysis or cointegration. For example, in the case of cointegration, a statistical mispricing M_t between of set of asset price series, P takes the form:

$$M_t = P_{T,t} - \sum_j \beta_j P_{C(j),t} \tag{6.9}$$

where T is the target asset, β is the cointegrating vector of constituent assets, $C(j)$.

For this process, control parameters can be used to organise the selection of the target time series, T and the set of constituent variables $C(j)$.

6.4.2 Optimisation Criterion

The construction of a forecasting model usually involves optimising some model parameters that define the functional form of the relationship between the explanatory variables and the forecast object, given some modelling methodology. This process is typically specified by some optimisation criterion.

The optimisation criterion (or loss function) is usually defined as some function of the prediction error. Loss functions are normally continuous functions that increase as the absolute value of the prediction error increases. One class that is commonly used is the symmetric loss functions, such as the squared-error loss. Other examples exist, such as the absolute loss and the "direction-of-change" loss function, which have primarily been developed on the basis of robust statistics. In addition, asymmetric loss functions have been developed for which negative forecast errors are viewed differently to positive forecast errors. The application of this type of loss function is usually based on some analysis of the decision environment (Diebold, 1998).

In the context of financial forecasting, a number of researchers have suggested modifying the loss function from the conventional least squares to a more robust metrics that reflect a trading application. For example, studies have suggested that, in the context of trading predictability, the sign of the forecast is a more useful measure of predictive ability (e.g. Refenes, 1995, Timmerman, 1996).

In this thesis we propose an extension of standard loss functions in which we consider the optimisation criterion as a general function of some predefined predictive characteristics. For example, we have showed in this chapter how a loss (gain) function may be specified for two characteristics, namely predictive correlation and prediction autocorrelation. The trade-off is controlled depending on the implementation of the trading strategy and the costs associated with trading. The loss function therefore reflects the predictive characteristics that maximise the economic value of the forecasting model with respect to a given trading strategy. For a general set of forecast characteristics, C (which may include percentage signs correct, predictive correlation, prediction autocorrelation, etc.) the optimal model parameters are defined as

$$\theta^{*} = \arg\max_{\theta}\left[\sum_{j} \alpha_{j} C_{j}(\theta)\right] \tag{6.10}$$

where α are the meta parameters specify the functional form of the loss function.

6.4.3 Forecast horizon

Forecasting models are generally constructed by dividing the historical data into a number of equal length time intervals that define the forecast object at discrete points in time. Each forecast then denotes the future value of the time series at subsequent future time intervals, denoted as the forecast horizon. The choice of forecast horizon can have significant implications to the accuracy of the forecasting model. This is because forecasting models are approximations to some particular underlying time series dynamics and so the accuracy of short term forecasting may not be related to the accuracy of longer term forecasting. In addition, the forecast horizon may have significant effect the economic value of an associated trading strategy. For example, short-term forecasts may provide more trading opportunities even if the forecasts are less accurate. In this case, more trading opportunities may result in a higher performance than trading longer-term forecasts even from a more accurate forecasting model. In contrast, shorter-term forecasts may not provide significant trading opportunities in situations with high transaction costs. In this case, higher trading performance may be achieved by using less accurate longer-term forecasts. The effect of forecast horizon is particularly pertinent to trading predictability as prediction accuracy can vary over different time horizons and trading can involve significant transaction costs.

A variety of techniques have been developed in order to optimise across different forecast horizon for a given data set. Some time series modelling techniques can be used to generate multi-step-ahead predictions. One of the disadvantages of multi-step forecasting is that models are generally calibrated for single step prediction. Alternatives include using overlapping samples constructed from high frequency data to create separate models for longer term forecasting. This approach requires some modifications to allow for the properties of constructing from overlapping historical samples (for more details see Taylor, 1998).

It is interesting to note that the evidence for predictability in financial markets has been detected over a wide range of time scales from very short (almost "tick by tick" that can be in the order of seconds), to much longer monthly or in extreme cases annual data. The wide range of potential time scales and forecast objects and the complexity of constructing forecasting models means that all possible time scales cannot be tested in practice. Consequently, particular time scales are investigated based on the style of the trading strategy rather than maximising predictability across all time horizons. In general we can assume that the degree of predictability of a forecasting model varies with the time horizon. This raises the issue to how to control the forecast horizon, τ in order to optimise some trading performance criteria U, which may be defined as

$$\tau^* = \arg\max_{\tau}\left[U(\tau)\right] \tag{6.11}$$

where τ denotes the range of time intervals.

There are a variety of numerical search methods for finding a bounded maximum depending on the characteristics of the function. These methods fall into two categories, gradient-based methods and heuristic methods. In this case, quantisation of the time horizon means that gradient based methods are inefficient. For this integer problem we use the Trisection method (defined in the next section) which can simply be implemented without computing a derivative of the function. For more details of optimisation techniques, see section 7.4. In the next section we show, using simulation experiments, how the forecast horizon influences trading performance.

6.5 Simulation Experiment: Optimising the forecast horizon

In this section we develop a synthetic trading system to investigate the effect of changing the forecast horizon on trading performance. The synthetic trading system was designed for trading predicted returns from a forecasting model, similar to section 4.4. The system consists of a data generating process that simulates the output from a forecasting model of single risky asset over different time intervals. The framework allows control of the time interval and prediction accuracy characteristics to enable a generic forecasting model to be emulated without constructing a specific model. This permits synthetic asset returns and predicted returns to be generated over any time horizon. The process of constructing the returns is described in equation (4.14) and equation (4.16). To construct the trading environment we assumed no market impact from trading so the costs were set to a fixed rate per unit change in the trading position, as described in equation (4.19), which emulates the typical transaction cost spread for trading. The trading strategy is set using a parameterised trading rule, as described in equation (5.2), with shape parameter k set to 1 and trading performance, in terms of average profit per unit time interval.

For each simulation run, 100,000 simulated asset returns are generated with associated predicted returns generated at a range of time intervals from 1 to 50 each with a controllable level of predictability. Trading positions were computed over each time horizon and trading returns recorded at each unit time interval in order to compare results across time horizons. Simulation experiments were repeated for different levels of transaction costs and forecasts with different levels of prediction correlation. An example of the relationship between forecast horizon and trading performance is given in figure 6.6.

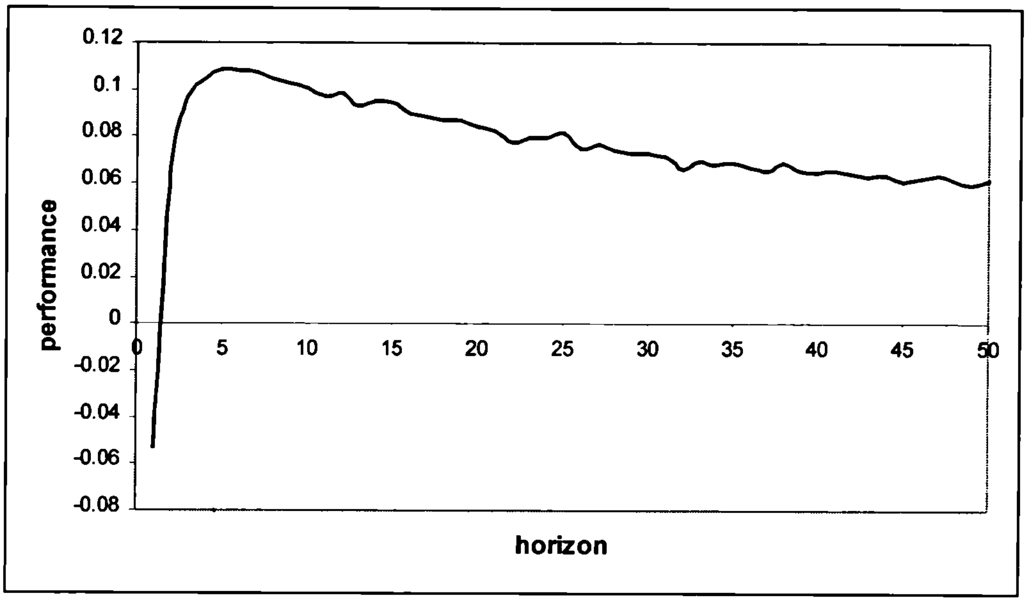


Figure 6.6 shows the relationship between trading performance, in terms of average profit, and forecast horizon for predictions with a predictive correlation of 0.5 and transaction costs of T=0.5.

These results indicate how profit varies with forecast horizon for a particular trading system. In this example, the optimal trading performance was recorded for a forecast horizon of 5 time intervals. The graph shows how increasing the forecast horizon from 1 to 5 time intervals significantly improves performance. For time intervals above 5, performance gradually deteriorates with a longer forecast horizon. This example demonstrates that the construction of a forecasting model may benefit from optimisation of the forecast horizon.

In this example, the optimal forecast horizon was identified by performing an exhaustive search across all possible time intervals. However, there are more computationally efficient optimisation methods that can be employed to find the maximum of a function. We review these methods later in section 7.4 in the context of our joint optimisation methodology. For example, in the case above of finding the optimal forecast horizon we may use a Trisection search method, which is effective for finding the optimal solution in well-defined problems without computing the derivative of the function.

The Trisection search method is defined as follows: If the maximum point is contained within the interval $[a,b]$, then define the sub-interval $[c,d]=[(a+1/3*h),(a+2/3*h)]$ where $h=b-a$. If $f(c)>=f(d)$ then the right sub-interval is dropped and the brackets adjusted for the next iteration. To apply this technique to optimise the forecasting horizon requires the rounding of time intervals and the construction of up to four forecasting models in each iteration. Table 6.4 shows the results from applying the trisection method to the trading system described in figure 6.4.

Iteration	Horizon				Performance			
	a	c	D	b	f(a)	f(c)	f(d)	f(b)
1	1	17	34	50	-0.0529	0.0884	0.0682	0.0612
2	1	12	23	34	-0.0529	0.0983	0.0798	0.0682
3	1	8	16	23	-0.0529	0.1048	0.0895	0.0798
4	1	6	11	16	-0.0529	0.1077	0.0970	0.0895
5	1	4	8	11	-0.0529	0.1047	0.1048	0.0970
6	4	6	9	11	0.1047	0.1077	0.1027	0.0970
7	4	6	7	9	0.1047	0.1077	0.1071	0.1027
8	4	5	6	7	0.1047	**0.1085**	0.1077	0.1071

Table 6.4: trisection search for finding the optimal forecast horizon for a trading system with prediction accuracy 0.5 and transaction costs of 0.5%.

These results show that the optimal horizon is found after 8 iterations which involves computing forecasting models over 16 different horizons compared to 50 different horizons for an exhaustive search. Further experiments were conducted across different levels of predictive correlation and transaction costs on the basis of three trading scenarios as previously discussed in section 4.4.1. The three scenarios represent trading different classes of assets depending on their associated price variability and typical transaction costs.

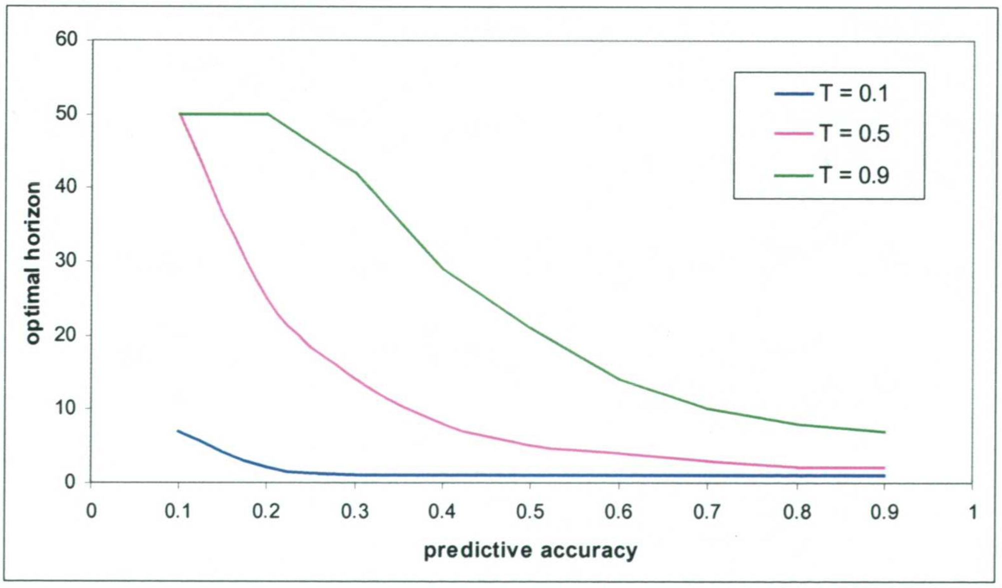


Figure 6.7 shows the optimal forecast horizon as a function of predictive accuracy (measured in terms of predictive correlation) for three different levels of transaction cost, represented by T =0.1, 0.5 and 0.9.

These results demonstrate how the optimal forecast horizon decreases with increasing prediction accuracy and also lower transaction costs. From our analysis, the optimal forecast horizon varies almost exponentially with predictive correlation. This behaviour can be explained in term of a trade-off between exploiting the prediction accuracy of the forecasting model and "higher" overall transaction costs associated with more frequency trading. Shorter forecast horizons allow more changes in the trading position, which generate higher overall costs but also provide a means of exploiting the predictability of the forecasting model.

6.6 Summary

In this chapter we have motivated the construction of forecasting models for trading based on control of model design factors which have been shown may indirectly effect the performance of the trading strategy. We have illustrated, using synthetic experiments, how the optimisation criterion can be controlled for two predictive characteristics using a meta parameter. A simple gradient ascent algorithm is implemented to search for the value of the optimal meta parameter. We discuss more general design factors that may influence trading performance and suggest techniques for controlling these modelling decisions. In particular, we discuss strategies for controlling the forecast object, optimisation criterion and forecast horizon. Simulation experiments are conducted to demonstrate techniques for optimising the forecast horizon. Our results show that for a trading system, with all other factors fixed, the optimal forecast horizon is influenced by the predictive accuracy of the forecasting model and the level of transaction costs.

7 Joint optimisation of a trading strategy and a forecasting model

In chapter 5 we developed a methodology to optimise a trading strategy given a forecasting model and in chapter 6 we developed a procedure to optimise a forecasting model conditionally on a trading strategy. These chapters illustrated that trading systems can be influenced by design factors that affect the construction of the forecasting model. In this chapter we combine the optimisation procedures of chapter 5 and 6 to develop a methodology to jointly optimise over the combined trading system to globally optimise trading performance. The proposed method repeatedly iterates between the two modelling stages for the tasks of forecasting and decision-making in order to maximise trading performance.

In section 7.1 we provide an overview to the process of performing a joint optimisation over a trading strategy and a forecasting model. In section 7.2 we discuss the joint optimisation of two interdependent tasks by considering iterative dynamic programming methods which repeatedly alternate between two processes to achieve joint optimality. In section 7.3 we illustrate, using synthetic examples, how the iterative methodology is able to control model design in order to optimise a two stage trading system. In section 7.4 we discuss the application of heuristic optimisation techniques which require no derivative. In section 7.5 we use simulation experiments to consider how our methodology can be applied to the optimisation of the forecast horizon using a simple heuristic technique, the Trisection search method.

7.1 Overview

We have shown in previous chapters that in order to efficiently trade predictability it is often preferable to develop two modelling stages: first a forecasting model to capture the predictability, then secondly a decision model to optimise the trading strategy and so exploit the predictive information. We have argued that although trading and forecasting are two interdependent processes it is often preferable to divide the system into two separate tasks. Given techniques for independently modelling these two tasks, a procedure is needed to control the optimisation of the two models in order to reach the goal of joint optimality. In this chapter we investigate the development of a joint optimisation methodology on the basis of the two modelling stages developed in chapters 5 and 6 and illustrated in figure 7.1

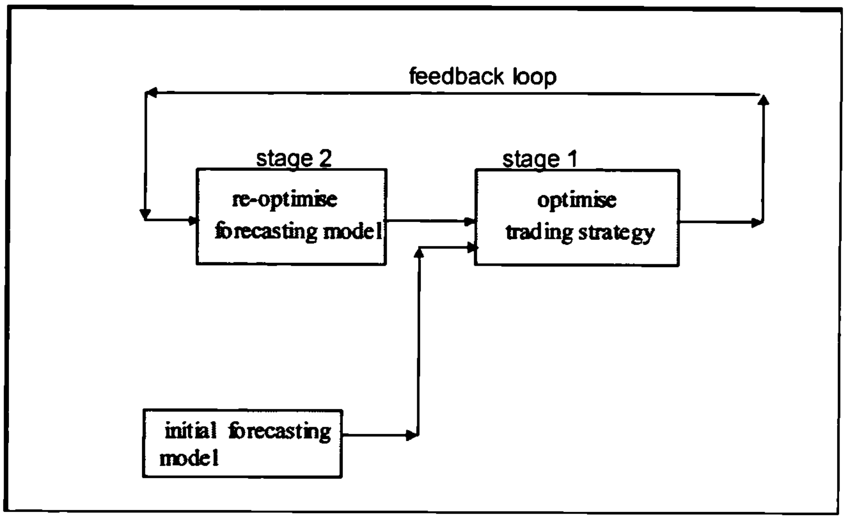


Figure 7.1 depicts the process of jointly optimising over both forecasting and trading tasks.

In chapter 5 we described stage 1 of the joint optimisation process which involved developing methods to optimise a trading strategy given some predictive information under different trading conditions. In the case of stable transaction costs, we developed a method using parameterised trading rules and, for trading under more general conditions (e.g. including market impact effects), we developed a method using reinforcement learning. For the purposes of developing a joint optimisation methodology we can consider stage 1 as a single decision model that can be used to approximate the optimal trading strategy given the expected cost of trading.

In chapter 6 we developed a method for optimising stage 2, which involved re-optimising the forecasting model with respect to the current trading strategy. We considered an iterative method of determining the optimal forecast model characteristics using a meta parameter approach which controlled certain model design factors. We showed using synthetic experiments how the iterative procedure can be used to control the optimisation criteria (i.e. the trade-off between predictive correlation and prediction autocorrelation) or more general features such as the forecast horizon. In each iteration a design factor is controlled using a meta parameter and the forecasting model re-optimised. Given these two modelling stages we can implement the joint optimisation process in two ways.

For one method, we could optimise the trading strategy and then re-optimise the forecasting model until the optimal meta parameters are found for the current trading strategy. This "course grained" approach may require the forecasting model to be reconstructed many times in each iteration in order to find the optimal meta parameters. Although this method should result in the goal of joint optimality it is generally considered a highly inefficient joint optimisation

optimisation procedure. This is because, at each iteration, the conditional optimisation of the forecasting model is computed with respect to a sub-optimal trading strategy.

Alternatively we could interweave the optimisation of each modelling task until convergence. In this "fine grained" approach each iteration only optimises each modelling stage once. This can be considered as gradually improving the forecasting model rather than estimating the optimal forecasting model given the current trading strategy. In this approach the models are alternately improved with the forecasting model not completing its task of finding the optimal forecasting model before re-optimising the trading strategy. Although this method is expected to require more iterations we consider it to be a more efficient optimisation technique.

Before examining the joint optimisation process in more detail, we summarise the rest of this chapter. In section 7.2 we discuss the joint optimisation of two interdependent modelling tasks by considering iterative dynamic programming that jointly optimises the two tasks of value function evaluation and the policy improvement. In general, these methods proceed by alternatively optimising each stage which is analogous to our preferred joint optimisation approach. This class of DP methods works through a combination of competition and co-operation, which in the long run leads to the goal of joint optimality: the optimal value function and the optimal policy.

In section 7.3 we develop an example of artificial trading system that is optimised using our joint optimisation methodology. The results show that the joint optimisation process controls the optimisation criteria of the forecasting model in order to improve expected trading performance. Results show that our joint optimisation approach significantly improves performance in the presence of transaction costs. In particular, performance improvements increase almost exponentially with transaction costs. We also validate our joint optimisation technique by completing an exhaustive search across parameter space, which confirms that our procedure converges to the joint-optimal solution.

In section 7.4 we briefly discuss employing other optimisation methods to optimise forecast model design factors by comparing derivative-based and heuristic techniques. In section 7.5 we use simulation experiments to jointly optimise the trading strategy and the forecast horizon using a simple heuristic based technique, the Trisection search method. Results show that expected trading system performance can be critically dependent on the choice of forecast horizon.

7.2 Optimisation of two interdependent tasks

In this section we describe how the optimisation of two interdependent tasks occurs in reinforcement learning methods and how this approach forms the basis for our preferred methodology to jointly optimise a trading model and a forecasting model.

The aim of DP and RL algorithms, which we reviewed as section 3.4, is to approximate the optimal value function from which the optimal policy can be approximated. However, these two processes are dependent on each other, as one process optimises the value function with respect to the current policy (policy evaluation) and the other optimises the policy with respect to the current value function (policy improvement). This problem is solved by interleaving the two processes without necessarily completing the optimisation of a single process. This type of method has been shown to guarantee convergence to the optimal value function and optimal policy, as long as each process continues to update all possible states (Sutton and Barto, 1998). If the method is implemented correctly, then the process produces no further changes in the policy and value function when the optimal solution is reached.

It is also interesting to note that the two sub-goals of policy evaluation and policy improvement often work against each other while still acting to optimise the modelling task. The process of updating the policy with respect the current value function typically makes the current value function incorrect as it was approximated from the *previous* policy. Likewise, updating the value function with respect to the current policy typically causes the policy to be incorrect as it was approximated from the previous value function. However, it has been found that in the long run the two processes do co-operate in achieving the overall goal of finding a joint optimal solution, as discussed by Sutton and Barto, (1998), and illustrated in figure 7.2.

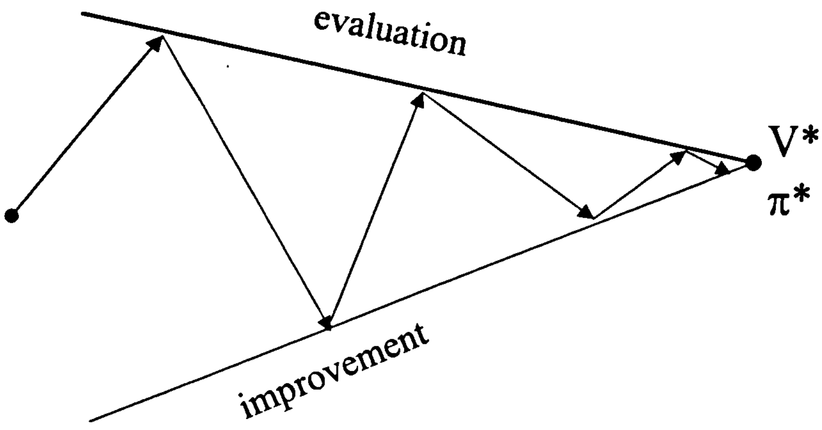


Figure 7.2 depicts the process of finding the joint optimal solution for the optimal value function, V^* and the optimal policy, π^*.

Figure 7.2 depicts conceptually the two sub-goals as two, non-orthogonal lines in two-dimensional space. The lines represent the "locally" optimal solutions of the two sub-goals, which are non-orthogonal, as the two processes interact with each other. If the two processes were independent then they would be represented by two orthogonal lines. In this case, the two processes could be optimised separately. For the non-orthogonal example, the optimisation process proceeds by first optimising one of the sub-goals which has the effect of moving the other process away from its optimal solution, although drawing closer to the joint optimal solution. The arrows indicate how the sub-goals solutions begin to converge as the two processes approach the overall goal of optimality, even though neither is attempting to achieve it directly. For DP problems, convergence has been shown to occur in finite time (Bertesekas, 1987).

Although our trading system is an instance of a more general formulation we can apply the same principles used for RL methods to construct a joint optimisation procedure. In our trading system we have shown previously that we have two interacting processes for the tasks of forecasting and decision-making. In this case, it is reasonable to assume that the solutions to the two modelling stages will interact in a similar fashion to the two processes of evaluation and improvement, which are shown conceptually in figure 7.2. Thus, in the context of the trading system, we except that the conditional optimisation of the forecasting model given a trading strategy will result in the strategy being effectively sub-optimal. Similarly, the conditional optimisation of the trading strategy given a forecasting model will result in the forecasting model being sub-optimal. In essence, each modelling process only optimises its own sub-goal, neither of which achieves optimality but only together do they have the potential of achieving joint optimality. If the process of joint optimisation for our trading system follows the same underlying principles as RL methods then we would except the process to be convergent. In this case, it seems rational to apply the same iterative approach to our trading system. In the next section we use simulation experiments to investigate the effectiveness of our joint optimisation approach for a simulated trading system.

7.3 Simulation Experiments

In this section we investigate the effectiveness of the two-stage optimisation process in reaching the goal of joint optimality. An artificial trading system is devised consisting of a forecasting model and a trading strategy as described in section 4.4. Simulation experiments are developed on the basis of experiments conducted in chapters 5 and 6. The trading strategy is implemented using a parameterised "path dependent" trading rule, as described in equation (5.3) with a decay rate parameter, θ, which controls the smoothing of the trading position through time. The trading environment is assumed to have stable transaction costs and a parameter is used to control the ratio of transaction costs to asset class volatility, as described in equation (4.18). The predicted returns are generated using the simulated trading system described in section 4.4.

The methodology for the optimisation of the trading system involves two modelling stages, first optimising the trading strategy and secondly re-optimising the forecasting model. These two modelling stages are repeated alternately until convergence. Setting the forecasting model optimisation criterion to maximise prediction accuracy initialises this process. A meta parameter is used to control the optimisation criterion to trade-off the two characteristics, predictive correlation and prediction autocorrelation, as described in equation (6.4) and updated using a gradient based method, as described in equation (6.6). The trading performance metric is taken as the ratio of expected return to standard deviation of returns as defined in previous experiments.

7.3.1 Example: Joint optimisation of an artificial trading system

An artificial trading system is constructed by building on the framework for simulating the predicted returns from a forecasting model with two parameters, which was developed in section 6.3. The original system is extended by implementing the trading strategy using a path dependent trading rule and generating predicted returns from the forecasting model using a controllable data generating process. The artificial system is described as follows:

Consider a forecasting model with two model characteristics, namely prediction correlation and prediction autocorrelation, which are directly related to two underlying model parameters, as devised in section 6.3. Suppose that the relationship between two characteristics, prediction correlation, β and prediction autocorrelation, φ, and the two model parameters, θ_1 and θ_2 is defined as follows:

$$\beta = \max\left(0, 0.3 - \left(\frac{\theta_1 - x_1}{\sigma(x_1)}\right)^2 - \left(\frac{\theta_2 - y_1}{\sigma(y_1)}\right)^2\right) \tag{7.1}$$

$$\varphi = \max\left(0, 0.9 - \left(\frac{\theta_1 - x_2}{\sigma(x_2)}\right)^2 - \left(\frac{\theta_2 - y_2}{\sigma(y_2)}\right)^2\right) \tag{7.2}$$

where x and y and $\sigma(x)$ and $\sigma(y)$ are function parameters, as defined in equation (6.3).

To investigate the effects of the predictive correlation and prediction autocorrelation in a trading system we select function parameters, as shown in table 7.1 and displayed graphically in figure 7.4.

	X	Y	σ_x	σ_y
Characteristic 1 (predictive correlation)	0.2	0.3	1	1
Characteristic 2 (prediction autocorrelation)	0.45	0.6	0.5	0.4

Table 7.1 shows the function parameters for equation 7.1 and 7.2 to simulate two model characteristics.

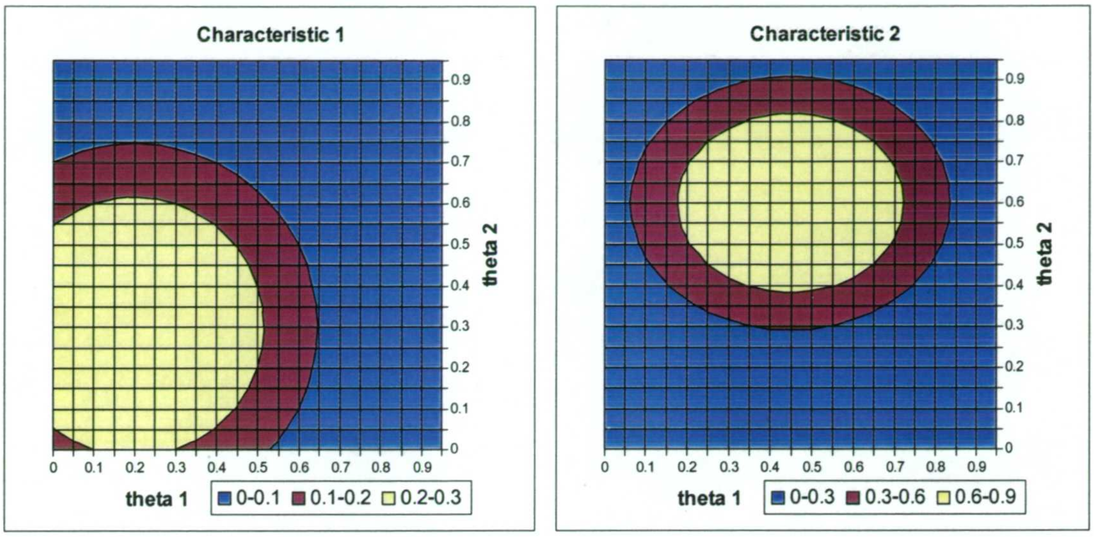


Figure 7.4 shows the two characteristics, predictive accuracy and prediction autocorrelation as a function of the two model parameters, θ_1 and θ_2.

For this example, the forecasting model has a maximum predictive correlation (characteristic 1) when the parameters x and y are at 0.2 and 0.3 respectively. This is considered equivalent to optimising a forecasting model to minimise prediction error and for our methodology forms the initial forecasting model. This is done in order to show that the optimal forecasting model can be "improved" in terms of its economic value to the trading strategy.

Given values for the two model parameters, θ_1 and θ_2, the two model characteristics can be used as inputs into a controlled data generating process to create predicted and actual return time series, as described in equations (4.13) and (4.15). The trading position is then determined by a "path dependent" parameterised trading rule, given in equation (5.3) with parameters k fixed to one and m acting as a normalisation factor. The smoothing parameter, θ, is modified during the first stage of the optimisation process to optimise the trading strategy. The forecast model parameters, θ_1 and θ_2, can be optimised given an optimisation criterion, as described in equation (6.4). This criterion is controlled using a meta parameter which is updated using equation (6.6) with step rate set to 0.1. On the basis of this artificial trading system, simulated returns were generated for 4000 observations and the transaction cost ratio parameter set to a range of values between 0 to 0.5. Results for the transaction cost parameter set to 0.2 are summarised in figure 7.2 with more detailed output available in Appendix A.

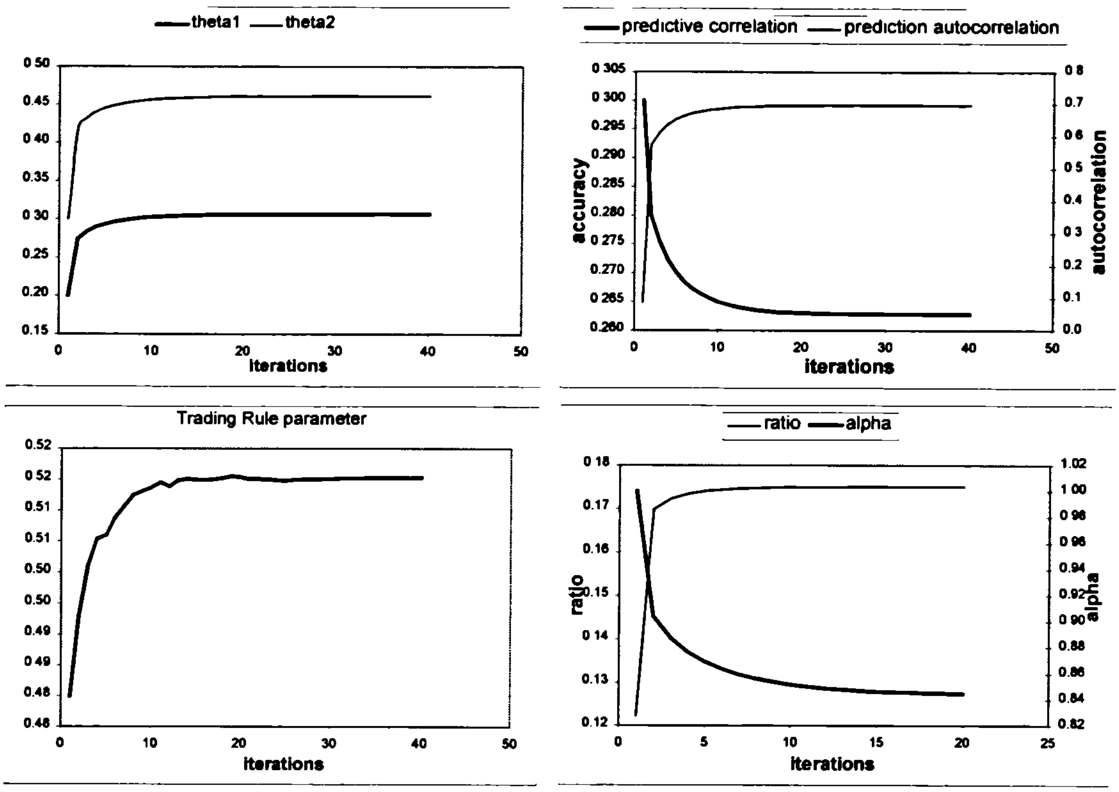


Figure 7.5 Diagrams show model parameters (top left panel), forecast characteristics (top right panel), smoothing parameter for the trading rule (bottom left panel) and meta parameter and performance ratio (bottom right panel).

Given the initial forecasting model with parameters (0.2,0.3) and predictive correlation 0.3, stage 1 is completed by optimising the trading rule parameter ($\theta = 0.480$) which gives a trading performance of 0.122. This is equivalent to building a forecasting model to minimise forecast error and then approximating the optimal trading strategy given the predicted returns. The joint optimisation process is then started by updating the meta parameter α, which controls the optimisation of the forecasting model. Subsequent iterations show how the meta parameter is gradually reduced, which has the indirect effect of changing the forecasting model parameters from (0.2,0.3) to (0.3,0.45), in order to boost autocorrelation at the expense of reducing predictive correlation (top right panel). Results show that trading performance converges after 17 iterations with an improvement in performance from 0.1223 to 0.1751, an increase of 43%.

For this simple trading system (i.e. three model parameters), we can validate our joint optimisation procedure by searching for the maximum performance across all parameter values. Using an exhaustive search procedure we found that the optimal trading rule and forecasting parameters were a close approximation (to within.4 significant figures) to the parameters found using the joint optimisation procedure. This indicates that the joint optimisation process has converged to the optimal solution. Further experiments were conducted to optimise the trading system for a range of transaction costs as shown in table 7.2. Results compared the two methods to optimise the trading system: the single optimisation approach which just optimises the two models once and the joint optimisation method which repeated optimises the meta parameter and the trading rule.

Tcosts	Method	Iter.	Forecasting Model				Characteristics		PTR	Perform
T			*α*	*Cost Fun.*	*θ₁*	*θ₂*	*β*	*φ*	*θ*	*Ratio*
0	Single	1	1.000	0.3000	0.2000	0.3000	0.3000	0.0875	0.0077	0.3015
	Joint	1	1.000	0.3000	0.2000	0.3000	0.3000	0.0875	0.0077	0.3015
0.05	Single	1	1.0000	0.3000	0.2000	0.3000	0.3000	0.0875	0.1127	0.2526
	Joint	7	0.9723	0.2972	0.2256	0.3454	0.2973	0.2933	0.1132	0.2558
0.1	Single	1	1.0000	0.3000	0.2000	0.3000	0.3000	0.0875	0.2212	0.2060
	Joint	9	0.9246	0.3029	0.2615	0.4013	0.2860	0.5110	0.2680	0.2209
0.15	Single	1	1.0000	0.3000	0.2000	0.3000	0.3000	0.0875	0.3384	0.1622
	Joint	12	0.8809	0.3158	0.2877	0.4373	0.2734	0.62935	0.4129	0.1952
0.2	Single	1	1.0000	0.3000	0.2000	0.3000	0.3000	0.0875	0.4799	0.1223
	Joint	17	0.8455	0.3298	0.3056	0.4550	0.2633	0.69401	0.5150	0.1751
0.25	Single	1	1.0000	0.3000	0.2000	0.3000	0.3000	0.0875	0.6007	0.0870
	Joint	23	0.8171	0.3427	0.3181	0.4749	0.2554	0.7327	0.5822	0.1583
0.3	Single	1	1.0000	0.3000	0.2000	0.3000	0.3000	0.0875	0.7134	0.0565
	Joint	16	0.8004	0.3509	0.3248	0.4827	0.2510	0.7514	0.6237	0.1435
0.35	Single	1	1.0000	0.3000	0.2000	0.3000	0.3000	0.0875	0.7867	0.0303
	Joint	24	0.7770	0.3629	0.3336	0.4926	0.2450	0.7737	0.6606	0.1303
0.4	Single	1	1.0000	0.3000	0.2000	0.3000	0.3000	0.0875	0.8508	0.0079
	Joint	20	0.7624	0.3708	0.3387	0.4982	0.2415	0.7857	0.6889	0.1182
0.45	Single	1	1.0000	0.3000	0.2000	0.3000	0.3000	0 0875	0.9900	-0.0053
	Joint	36	0.7375	0.3846	0.3468	0.5070	0.2356	0.8033	0.7166	0.1072
0.5	Single	1	1.0000	0.3000	0.2000	0.3000	0.3000	0.0875	0.9900	-0.0107
	Joint	32	0.7217	0.3937	0.3517	0.5120	0.2320	0.8130	0.7396	0.0970

Table 7.2 is a summary of simulation results over a range of transaction costs from 0 to 0.5 showing optimisation method, iterations, meta parameter, α, forecasting model cost function and parameters, trading rule parameter (PTR) and trading performance ratio, which represents a scaled version of the Sharpe Ratio.

Results show that, for no transaction costs, joint optimisation has no advantage over the single optimisation of the forecasting model and the trading strategy, with equal trading performance of 0.3015. Increasing trading costs has the effect of gradually reducing trading performance as expected, with the trading rule parameter increasing in order to boost the smoothness of the trading position through time. For all non-zero transaction costs the joint optimisation approach outperforms the single optimisation. This is achieved by gradually reducing the meta parameter, thereby optimising the trade-off between predictive accuracy and prediction smoothness for the optimal trading strategy. The percentage performance difference between joint and single optimisation is shown in figure 7.6. Again we tested convergence by searching across parameter space and found that model parameters were within 4 significant figures of the optimal values.

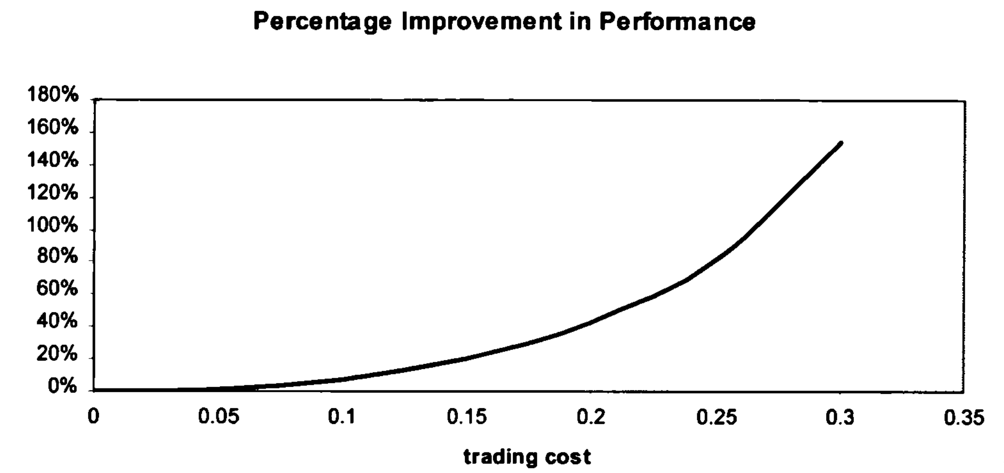


Figure 7.6 show the percentage difference between joint and single optimisation over a range of the transaction cost parameter.

Results show that joint optimisation can provide a significant increase in performance for trading systems with non-zero transaction costs. In this particular experiment, the percentage performance improvement increases at an exponential rate with increasing transaction costs. This shows how the joint optimisation methodology could be expected to improve real-world trading systems, which suffer from performance degradation due to the cost of trading.

7.4 Employing Advanced Optimisation Techniques

In the previous subsection we employed a simple gradient ascent method to control the search for the optimal meta parameter in order to converge to the joint optimal solution. This approach requires specification of the learning rate and is a suitable method for trading-off two characteristics of the optimisation criteria. In more general circumstances, however, for trading systems that require the specification of a number of meta parameters, other optimisation methods may prove more efficient and reliable. In this section we concisely review the application of other optimisation techniques which may be used to search for the optimal meta parameters.

Any optimisation problem can be considered to involve maximising (or minimising) an objective function of a number of variables, which is considered to have certain properties (e.g. linear, quadratic, non-linear, sparse, etc.) and possibly subject to constraints that restrict the values of the variables. The diversity of potential optimisation problems means that a single, all-purpose algorithm cannot be used to find the optimal solution and so a wide range of iterative methods have been developed to solve global optimisation problems. In general, methods fall into two categories: derivative-based and heuristic methods.

The former group contains methods that fit polynomial functions using stochastic gradient decent techniques such as Newton-type, Quasi-Newton and conjugate gradient methods. In contrast the latter group is based on methods that require no derivative and include downhill simplex algorithm (Nelder and Mead, 1965), Levenberg-Marquardt algorithm, grid search techniques and methods that learn from experience including Genetic Algorithms (Holland, 1975), Simulated Annealing (Metropolis, 1953) and Taboo Search (Glover, 1982). These heuristic methods are able to identify possible local minima (maxima) and non-linearity effects, and are particularly useful when the objective function has sharp contours, a large number of local minima, or the function is ill-conditioned (Craw,1999). Modern learning methods, such as G.As, are often cited as useful heuristics for global optimisation problems and have proved useful in financial applications (Glover et al., 1995; Burgess, 1999). These methods have been shown to be useful when the search space is very large, little is known about the problem structure or the cost function is noisy with significant variance. They are capable of searching an entire solution space and can have a higher probability of finding good, near global optimal solutions than traditional search and optimisation procedures. However they are a poor choice for problems readily solved by standard optimisation tools (Frost, 1997).

For the joint optimisation methodology it may be appropriate to consider heuristic methods in circumstances where the meta parameter space is ill-conditioned or where there is no obvious initial forecasting model. For example, in the case of optimising the forecast horizon there may be no obvious initial forecasting horizon and the integer step size leads to an ill-conditioned function which cannot be solved using gradient based methods. In the next section we implement a simple heuristic method to optimise a population of models, using the Trisection search method. This is considered a simple heuristic method but illustrates the application of other more complex methods which would be necessary for optimising multiple design factors.

7.5 Simulation Experiment: Optimising the Forecast Horizon

In this section we investigate the effectiveness of the two-stage optimisation process in optimising the forecast horizon. A synthetic trading system is devised consisting of a forecasting model and a trading strategy, as described in section 7.3. The trading environment is assumed to have stable transaction costs and a parameter is used to control the ratio of transaction costs to asset class volatility, as described in equation (4.18). The predicted returns are generated using the simulated trading system described in sections 4.4 and 6.5 which permits the returns to be generated over multiple time steps. For the purposes of this simulation we constrain the range of possible forecast horizons between 1 and 50 integer time steps. In this case, quantisation of the time horizon means that gradient based methods are inefficient and so we resort to heuristic methods. For this problem, which involves controlling a single design factor we implement the Trisection method which can simply implemented without computing a derivative. Fore more advanced problems other population based methods could be employed as described in the previous section.

In these experiment we assume that the prediction accuracy of the predicted returns is constant over all time horizons which enables us to focus on attempting to optimise the forecast horizon for an optimal trading strategy given some level of transaction costs. In the first experiment we simulated predicted and actual returns with a controllable prediction correlation of 0.3 and transaction cost parameter set to 0.5. A summary of results is given in table 7.3.

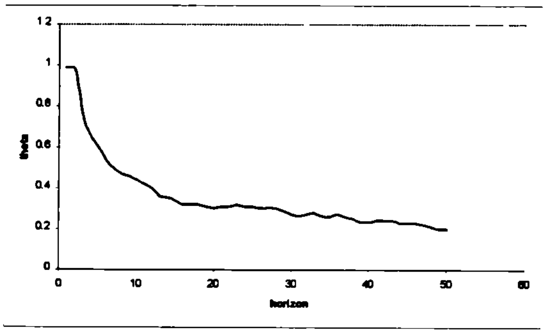
Iter.	Horizon				Trading rule parameter, θ				Performance ratio			
	a	c	d	b	a	c	D	b	a	c	d	b
1	1	17	33	50	0.990	0.350	0.305	0.221	0.001	0.045	0.040	0.036
2	1	11	22	33	0.990	0.420	0.315	0.305	0.001	0.046	0.043	0.040
3	11	18	25	33	0.420	0.320	0.310	0.315	0.046	0.041	0.040	0.040
4	11	15	20	25	0.420	0.333	0.313	0.310	0.046	0.043	0.042	0.040
5	11	14	17	20	0.420	0.341	0.315	0.313	0.046	0.044	0.043	0.042
6	11	13	15	17	0.420	0.358	0.333	0.315	0.046	0.046	0.044	0.043
7	11	12	13	15	0.420	0.389	0.358	0.333	0.046	0.047	0.046	0.044

Table 7.3: application of trisection search for finding the optimal forecast horizon for a trading system with prediction accuracy 0.3 and transaction costs of $T = 0.5$.

In the first iteration four forecasting models and their associated trading strategies are optimised for time horizons 1,17,33 and 50. The performance for time horizon 17 is greater than horizon 33 so the optimisation method assumes that the optimal horizon is not between horizon 33 and

50. The upper limit for the forecast horizon is then decreased to 33 and the second iteration proceeds. For this experiment the optimal horizon converges to 12 time periods after 7 iterations. This procedure for optimising forecast horizon has improved trading performance for the naïve forecast horizon (i.e. 1 time step) of 0.001 to 0.047. These results indicate the potential importance of optimising the forecast horizon in the presence of significant transaction costs. This optimisation approach requires the construction of 14 forecasting models against an exhaustive search procedure which requires 50 models.

We can check that joint optimality is reached by conducting an exhaustive search across the entire space of possible time horizons and trading rule parameter values. Figure 7.7 shows the performance ratio and optimal θ for the entire range of forecast horizon.



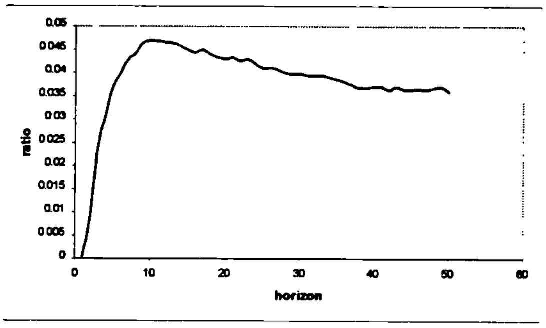


Figure 7.7 shows the optimal θ and the associated performance ratio over all forecast horizons for predicted returns with predictive correlation 0.3 and transaction costs of 0.5.

These results show that the trisection search accurately identifies the optimal solution. Further experiments are conducted for different levels of transaction cost between 0.1 and 0.7, to investigate the effect the level of transaction costs, as shown in table 7.4.

Costs	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Optimal Horizon	1	1	1	4	8	12	17	30
Optimal trading rule	0.008	0.191	0.221	0.174	0.205	0.389	0.427	0.486
Performance ratio	0.301	0.205	0.114	0.071	0.062	0.047	0.042	0.034

Table 7.4 shows the optimal forecast horizon, optimal trading rule parameter and performance ratio for increasing levels of transaction cost.

The results demonstrate that the optimal forecast horizon increases with higher transaction costs. In addition, the trading rule parameter increases and performance degrades with higher transaction costs. These results can be explained using the same analogy of the bias-variance trade-off described earlier in section 5.2.4. In this context, bias and variance are effected by the forecast horizon and the smoothness of the optimal trading rule. Thus "simple" models (large bias) have a long forecast horizon or smooth trading rule and so infrequently change the trading position. In contrast, "complex" models (high variance) have a short forecast horizon or flexible trading policy and so frequently change trading position. The optimal trading system, is then a trade-off between exploiting predictability (low bias) and minimising trading costs (low variance). This is accomplished by controlling the forecast horizon given the optimal trading rule. For example, a short forecast horizon may require a high degree of smoothing to control trading costs which results in a poor trading performance. In contrast, a long forecast horizon may not exploit available predictability and result in less frequent trading which leads to sub-optimal performance. Thus controlling the forecast horizon allows the trading system to optimise the bias-variance characteristics in order to improve trading performance. This analysis demonstrates the bias-variance point made earlier but from another angle.

7.6 Summary

In this chapter we have discussed the integration of the two modelling stages in order to joint optimise trading performance. We propose a joint optimisation methodology that interleaves the two modelling stages which is based on techniques used for DP and RL methods. Simulation experiments have been devised to test the convergence of the methodology for a forecasting model with two characteristics which both influence trading performance. A meta parameter construct is used to control the optimisation criteria of the forecasting model and our results demonstrate the effectiveness of the methodology. Both modelling components of the trading system are alternately optimised until a stable trading system is reached. A global search procedure is implemented to test convergence to the joint optimal solution. Other design factors are discussed and a method developed to optimise the forecast horizon. Further simulation experiments are conducted to test the control of the forecast horizon for a given level of predictability. Simulation results demonstrate that the methodology can be successfully applied to synthetic trading systems in the presence of significant transaction costs.

8 Application: Statistical Arbitrage Trading

In this chapter we apply our joint optimisation methodology to the development of statistical arbitrage trading strategies for predictive models. We build upon the methodology developed by Burgess (1999) for modelling the dynamics of statistical arbitrage which identifies and predicts statistical mispricings from amongst groups of assets. We start by discussing the development of our two-stage trading system for forecasting and exploiting a statistical mispricing. We then conduct extensive empirical evaluations using our joint optimisation methodology for a realistic set of 50 statistical mispricings, which were identified within the UK equity market (FTSE 100). Trading systems are optimised across all statistical mispricings for different levels of transaction costs. The results demonstrate that our parameterised trading rules and joint optimisation procedure can significantly improve trading performance.

In section 8.1 we introduce the concept of statistical arbitrage by drawing on the underlying principles of arbitrage trading and statistical analysis of financial assets. In section 8.2 we outline the methodology developed by Burgess (1999) for constructing and detecting statistical relationships from within a group of assets. In section 8.3 we apply our methodology to the optimisation of a trading system comprised of a forecasting model for some statistical mispricing and a trading strategy. In section 8.4 we consider optimising realistic statistical arbitrage trading strategies in the presence of fixed transaction costs.

8.1 Overview

From a conceptual perspective, statistical arbitrage can be considered an extension of traditional "riskless" arbitrage trading, which is defined as a financial transaction that is able to make almost immediate profit at (or close to) zero risk. Arbitrage opportunities occur when pricing anomalies appear in markets, which generate price differences between assets with equivalent cash flows. If the difference in price between two equivalent assets deviates from zero then it is considered a *mispricing*. These pricing discrepancies may then be exploited through a portfolio of equally matched short and long positions with the short-sale of the overpriced asset (or portfolio of assets) and the simultaneous purchase of the under-priced asset. Typically, mispricings tend to be small compared to the size of "market quoted" transaction costs and so traditional arbitrage trading is primarily conducted by large financial institutions who have ability to deal in large volumes of transactions at lower transaction costs.

Arbitrage trading involves identifying assets with future cashflows that can be replicated by some combination (or portfolio) of other assets and then monitoring asset prices until a significant mispricing is detected. Trading then relies on the implicit assumption that the markets of the assets under consideration are theoretically *arbitrage free* (i.e. hold to the no arbitrage condition). This implies that any significant arbitrage opportunities will rapidly attract sufficient market participants whose profit generating activities will eliminate the opportunity. This guarantees that arbitrage trading generates profits almost immediately.

If we consider the price of an arbitrary asset, denoted by Y, and some replicating combination of assets that can be considered as a "synthetic" asset of Y, denoted by $SA(Y)$, then the difference in price (i.e. the mispricing) can be related to the cost of trading. We can represent the no-arbitrage condition in the general form:

$$\left|Y_t - SA(Y_t)\right| < TransactionCost \tag{8.1}$$

where *TransactionCost* represents the net costs involved in constructing (buying) the synthetic asset, $SA(Y)$ and selling asset Y (or vice versa). Clearly, arbitrage opportunities only exist if the no-arbitrage condition is temporarily violated so that the payoff outweighs the transaction cost.

Equation (8.1) forms the basis of the no-arbitrage pricing approach conventionally used in the pricing of derivatives, such as options and futures. In this case, the expected price of the derivative is defined by the price of the appropriate replicating portfolio (or synthetic asset). The mispricing is then the absolute price difference between the observed derivative price and the replicating portfolio.

In well-developed markets, classical arbitrage opportunities may be few so, in practice, most arbitrage traders also use strategies that have some level of risk which are expected to produce consistent profits under normal market conditions. This framework for arbitrage trading is similar to classical arbitrage and is often termed *statistical arbitrage*. In this type of arbitrage trading, a mispricing is based on statistical price discrepancies with a trading strategy that offers attractive risk-return characteristics rather than risk-free profits. For example, a simple statistical arbitrage strategy may consider the credit-spread between a government bond and a corporate bond. If the historical yield spread is within a 10 to 40 basis point range then it may be considered that a risk arbitrage opportunity occurs if the spread drops below 10 basis points. The classical arbitrage opportunity only occurs if the spread becomes negative. The assumption of the statistical arbitrage opportunity is that the spread will return to within the historical yield range so provide an opportunity to exploit the statistical regularities of the market without true arbitrage taking place. In general, the statistical arbitrage equivalent of the no-arbitrage condition, described in equation (8.1), takes the form:

$$E\left[\left|Y_t - SA(Y_t)\right|\right] < TransactionCost \tag{8.2}$$

In this definition the expected mispricing is <u>not</u> a true mispricing but based on some statistical relationship identified from historical price data and so there is no guarantee that a mispricing will produce an immediate profit or even short term price convergence. Thus this style of trading involves some degree of risk. One advantage of this style of trading is that the amplitude of the mispricing may be relatively large compared to the transaction cost and so trading opportunities are more numerous and potentially exploitable by a wider group of financial investors.

In this chapter we consider a form of statistical arbitrage which incorporates the use of modelling techniques to develop trading strategies based on identifying and exploiting statistical regularities in asset price dynamics. The development of these statistical arbitrage models typically involves three components (Burgess, 1999):

- identify appropriate combination (portfolio) of assets,
- develop a forecasting model to predict the change in the statistical mispricing, and
- optimise a trading strategy to exploit the predictive information while minimising transaction costs.

Modelling techniques can be applied to each of these three tasks. Methods for constructing statistical mispricings between assets and testing for some predictable component have been investigated by Burgess (1999). In this chapter we concentrate on tasks two and three which involve predicting and exploiting the identified mispricing. This problem neatly fits with our developed methodology to jointly optimises a trading strategy and a forecasting model.

In section 8.2 we discuss the identification of statistical arbitrage relationships by considering the construction of a statistical mispricing based on either principal component analysis (PCA) or cointegration analysis and also review the application of tests for predictability. In section 8.3 we discuss the application of forecasting methods to predict the mispricing return and decision models to optimise the trading strategy given different trading conditions. We also discuss the application of our joint optimisation methodology for a given statistical mispricing. In section 8.4 we apply our methodology to the optimisation of realistic examples of statistical arbitrage trading strategies for predictive models. Results on out-of-sample data show that "path dependent" trading rules provide more consistent performance than naïve trading rules and that joint optimisation improves performance in the presence of transaction costs.

8.2 Identifying statistical arbitrage relationships

The identification of statistical arbitrage relationships requires the construction of a time series of some linear combination of assets, which contains some deterministic component of the asset price dynamics. One method of identifying a deterministic component involves constructing and detecting a mispricing time series which exhibits some predictable behaviour. Forecasting methods can then be applied to the time series in order to predict the future behaviour. This model identification process can be broken down into two stages:

- Construction of a statistical mispricing
- Tests for return predictability

In the first subsection we discuss multivariate tools to construct a statistical mispricing and in the second, tests for return predictability.

8.2.1 Construction of a statistical mispricing

The construction of a statistical mispricing is based on the analysis of the price relationships within a group of assets. This is analogous to relative value trading, such as "pairs" trading (i.e. two assets and some arbitrary chosen relationship), which are frequently used by practitioners to gain some measure of price relationships in relative rather than absolute terms. Multivariate methods can be applied to the identification of relationships between groups of assets and in this subsection we focus on two: Principal Component Analysis and Cointegration.

Principal Component Analysis (PCA) is a multivariate technique for decomposing the information in a set of variables. Sequences of linear combinations (factors) are generated which account for the greatest possible amount of total variability in the data set. In this context, PCA can be used to identify combinations of assets which contain some asset specific component of the price dynamics. A statistical mispricing, M_t for a group of assets Y_j then takes the form:

$$M_t = \sum_j \beta_j Y_{j,t} \tag{8.3}$$

where β_j is coefficient for each asset j.

The statistical mispricing, M_t, can be considered in the same manner as any conventional asset price time series and equivalent to a value of a fixed mispricing portfolio of assets. From the statistical arbitrage perspective, this is analogous to other relative value techniques, such as Arbitrage Pricing Theory (Ross, 1976). In this sense, the linear combinations of assets act as observable risk factors. The application of PCA for the construction of a statistical mispricing is described in a number of papers, including Burgess (1995), Schreiner (1997), Towers (1998), Tjangdjaja et al. (1998) and Towers and Burgess, (1998).

An alternative approach is to use part of cointegration modelling framework to attempt to construct a mean reverting time series. In this process, some of the assumptions of the traditional cointegration technique can be relaxed in order to test for predictable behaviour rather than a stationary error-correction effect. Thus, the stationary tests of cointegration analysis, for example, the Dickey-Fuller test and the Cointegrating Regression Durbin-Watson, can be dropped and replaced by tests for potential predictability, as discussed in section 2.4 and also in the next subsection.

For example, given a target asset T, a statistical mispricing M_t, can be considered as a portfolio of assets $\{T, Y_1, Y_2, Y_3, \ldots\}$ with respective weightings $\{1, -\beta_1, -\beta_2, -\beta_3, \ldots\}$ where β is cointegrating vector. The statistical mispricing can be considered as the value of the target asset T relative to the linear combination of constituent assets. This method of constructing a statistical mispricing using cointegration analysis has been extended to adaptive cointegrating vectors (using smoothing techniques or adaptive regression techniques). In the case of high dimensional problems, with many potential combinations of assets, techniques may be required to search through potential permutations. For example, in the case of the FTSE 100, there is a vast number of potential combinations of assets, with over 1 million possible combinations involving a target and a replicating portfolio containing just two assets. Obviously an exhaustive search across all combinations is computationally impractical and so a guided search mechanism is required to identify combinations which exhibit significant mean-reverting behaviour. One practical method of tackling this issue is by applying a stepwise linear regression approach to estimate the mispricing relationships as suggested by Burgess (1999).

8.2.2 Testing for Return Predictability

In this section we discuss tests for detecting potential return predictability in the dynamics of the statistical mispricing time series. A number of statistical tests have been developed for identifying non-random walk behaviour in a time series (for more details see section 2.4).

In the case of a statistical mispricing time series, tests for predictability against random walk behaviour can be developed using autocorrelation functions (ACF) and "portmaneau" statistics, such as, the Q-statistic and the Variance-Ratio (VR) statistic. From a statistical arbitrage perspective, predictability is considered in the form of mean-reverting behaviour, which measures the rate at which the statistical mispricing is expected to converge to zero. The power of these tests have been compared for bias corrected mean-reversion, as described by Burgess (1999). Results show that the *VR* statistic is the most powerful test for the identification of potential predictability in relative asset prices.

In the next section we assume that an appropriate statistical mispricing has been constructed and some mean reverting behaviour is identified. Forecasting methods are then applied to the statistical mispricing in order to predict future behaviour. We also examine methods to approximate the optimal trading strategy given predicted returns in the presence of transaction costs.

8.3 Optimising a Trading Strategy for a Statistical Mispricing

In this section we discuss development of a forecasting model for a statistical mispricing. We describe the application of general forecasting techniques to predict the change in the statistical mispricing. We discuss the optimisation of the trading strategy given the forecasting model and discuss the application of parameterised trading rules and reinforcement learning. In addition we discuss the application of our joint optimisation methodology in order to optimise a trading strategy for a forecasting model of a statistical mispricing. We describe the development of forecasting models given the statistical mispricing, the control of model design factors and the choice of decision modelling technique given the expected trading environment.

8.3.1 Constructing the Forecasting model

Forecasting methods can be applied to the statistical mispricing time series in order to forecast the expected behaviour and thus optimise the characteristics of any predictability. The obvious forecasting model is to rely on the inherent mean-reverting behaviour of the identified mispricing time series. This model assumes that the future price changes will on average tend to reduce the mispricing between the target asset and the synthetic asset. This is the basis for arbitrage relationships that are expected to obey the no-arbitrage condition. This "implicit" forecasting model requires no model parameters can takes the form (Burgess, 1999):

$$\Delta M_{t+1} \propto -M_t + \varepsilon \tag{8.5}$$

where ΔM_{t+1} is the future change in the mispricing time series.

Other forecasting models can be built using a range of techniques, as discussed in chapter 3. Smoothing or moving average models can be constructed from the mispricing time series to predict the next value. For example, the exponential smoothing model takes the form:

$$M_{t+1} = \alpha M_t + (1-\alpha)\hat{M}_t + \varepsilon \tag{8.4}$$

where α is the smoothing parameter. The predicted change in the mispricing is then simply computed as the difference between the current mispricing and the predicted mispricing.

More advanced univariate models can be developed by incorporating ARMA modelling terms, which takes the general form:

$$\Delta M_{t+1} = \alpha + \sum_{i=0}^{p} \beta_i M_{t-i} + \sum_{j=0}^{q} \gamma_j \Delta M_{t-j} + \sum_{k=0}^{r} \delta_k \varepsilon_{t-k} + \varepsilon \tag{8.5}$$

Other multivariate modelling techniques, such as linear regression analysis or neural networks, can be applied to this framework, which, in general, takes the form:

$$\Delta M_{t+1} = f(M_t, \Delta M_i, Z; \theta) + \varepsilon \tag{8.6}$$

where θ is the vector of model parameters and Z is the vector of exogenous variables.

The construction of forecasting models for a statistical mispricing may also have a number of additional design factors that require specification. It is considered that these design factors influence the economic value of the forecasting model with respect to some trading strategy. In the case of statistical arbitrage trading systems, these factors may include the optimisation criterion, forecast horizon and forecast object. These factors may be particularly important as they influence the characteristics of the predicted returns. For example, the optimisation criterion may consist of two statistical characteristics, namely prediction autocorrelation and predictive correlation, which may both influence the economic value of the predicted returns. It is considered that statistical arbitrage trading, which involves exploiting predictability to overcome trading costs, may be particularly sensitive to the characteristics of the predicted returns. In this thesis we have proposed a method to control these types of design factors for a trading system consisting of a forecasting model and a trading model. This is achieved using *meta* parameters within an iterative optimisation procedure, as discussed in chapter 6. This forms our methodology to perform a joint optimisation over both forecasting and decision models. In the next subsection we discuss the application of developed decision modelling techniques to optimise the trading strategy.

8.3.2 Modelling the Trading Strategy

For trading systems, the exploitation of a forecasting model of asset returns is often critically dependent on the optimisation of the trading strategy that can trade-off the benefits of exploiting predictability while minimising trading costs. This sensitivity is particular exacerbated in the case of statistical arbitrage trading, in which predictive models are employed to achieve excess profits at low risk levels. The implementation of statistical arbitrage strategies typically involves rebalancing short and long positions in a portfolio of assets ensuring all trades are completed simultaneously. This type of trading is particularly prone to market impact effects, which cause wide variations in transaction costs and can significantly influence trading performance. Statistical arbitrage trading is typically conducted on relatively high frequency data (i.e. daily or intra-day time scales) in order to capture any predictable behaviour between assets. For trading opportunities to exist mispricing fluctuations must exceed the expected cost of trading.

In this thesis we have developed decision-modelling techniques for optimising a trading strategy under stable and varying transaction costs using parameterised trading rules and reinforcement learning respectively. For statistical arbitrage trading the development of a forecasting model means that we can directly apply our decision models to the predicted mispricing return rather than simple heuristic trading rules which have been used in other studies (e.g. Burgess, 1999). In the case of variable transaction costs, arising from market impact, a transaction cost model is required to estimate the expected cost of trading. In principle, our reinforcement learning algorithm may then be used to optimise the trading strategy given the state-action space. In the case of stable trading costs, the path dependent trading rules developed in chapter 5 forms a natural modelling framework to control the trading position for a portfolio of assets. Thus we can apply the three path dependent trading rules, which are described in equations (5.3), (5.4) and (5.5), and are designed for the purpose of optimising a trading strategy given some predicted return series. These decision-modelling techniques provide a framework to approximate the optimal trading strategy for predicted returns of a statistical mispricing in the presence of either stable or varying transaction costs. In the next section we apply our modelling techniques to identified statistical mispricings from within the FTSE 100 and evaluate trading performance.

8.4 Empirical Experiments

In this section we test our joint optimisation methodology by considering statistical arbitrage models constructed for equity markets as part of the HAT (High Performance Arbitrage Detection and Trading) project. This project was undertaken by a consortium including London Business School, Reuters, Dresdner and BNP from January 1997 to December 1998. The purpose of the project was to develop an integrated prototype system for real time detection, monitoring and exploitation of arbitrage opportunities across a spectrum of markets and instruments. The role of the computational finance group at London Business School was to develop analytical tools for construction and detection of potential arbitrage opportunities. A major component of the work focused on identifying statistical arbitrage opportunities in major equity and fixed income markets.

Real time market data was collected and processed for different classes of financial instruments over a range of time scales from one minute up to daily data. Techniques for detecting statistical arbitrage relationships were employed using cointegration analysis as discussed in section 8.2. The modelling process for constructing a statistical mispricing is as follows (Burgess and Towers, 1998):

- specify the universe of assets, the time-period over which the cointegration analysis will be performed, and the number of constituent assets
- take each asset in turn as the target asset and perform a stepwise regression to identify the constituent assets, which form a combination which most closely tracks the price movements of the target asset
- generate the synthetic asset price and the mispricing
- repeat this process for each target asset before moving on to build the forecasting models

Given a mispricing, a trading system can then be constructed using a forecasting model and a trading strategy. These two stages can be implemented using forecasting methods and trading strategies, as discussed in the previous section. This produces a trading system that allows our joint optimisation methodology to be employed to improve trading performance. In the next subsection we apply our methodology to a set of statistical mispricings that have been detected within the FTSE 100. In these evaluations we restrict our attention to consider stable transaction cost environments and so employ our path dependent trading rules, as described in chapter 5.

8.4.1 Intraday Equities (FTSE 100)

In this section we present the results of applying our methodology to a problem of exploiting statistical mispricings within an equity market. The equity market data used in the experiments was collected from the FTSE 100 (UK). The data set consists of 1100 observations of hourly prices collected from a live Reuters data-feed from 9 a.m. to 4 p.m. daily during the period from the 15 May 1998 to 4 December 1998. The first 400 data points were used for constructing and testing the statistical mispricings and the remainder of the data for optimising model parameters and out-of-sample evaluation. To achieve this the second period was split into two data sets, an in-sample period of 500 observations to optimise the forecasting model and the trading rule parameters and the final 200 observations as the out-of-sample period. After filtering of merged/discontinued stocks from the FTSE 100 the remaining universe of assets included the index plus 96 constituents as shown below:

ABF	AL	ALLD	ANL	ASSD	AVZ	BA	BAA	BARC	BASS	BAY	BCI	BG	BGY	BLND	BLT
BOC	BOOT	BP	BS	BSCT	BSY	BT	BTR	CBRY	CCM	CNA	CW	DGE	DXNS	EMI	ETP
GAA	GARD	GEC	GKN	GLXO	GUS	HAS	HFX	HSBA	ICI	III	KGF	LADB	LAND	LGEN	LLOY
LSMR	LVA	MKS	NAM	NGG	NPR	NU	NWB	NXT	ORA	PO	PRU	PSON	PWG	RBOS	RCOL
REED	RIO	RNK	RR	RSA	RTK	RTO	RTR	SB	SBRY	SCTN	SDR	SEBE	SFW	SHEL	SLP
SMIN	SPW	STAN	SVT	TOMK	TSCO	TW	ULVR	UNWS	UU	VOD	WLMS	WLY	WTB	WWH	ZEN

Figure 8.1 illustrates the universe of 96 selected assets (using Reuters codes) from the FTSE 100.

A cointegration based framework was used to construct each statistical mispricing which incorporated stepwise linear regression to direct the search for suitable combinations across the potential space of possible combinations, as discussed in section 8.2. Furthermore, due to stability issues the number of constituent assets in the cointegrating vector was limited to 4. Using this method, 50 statistical mispricings were selected on the basis of the degree of mean reverting behaviour, which was tested against random walk hypothesis using the Variance Ratio statistic (Burgess, 1999). This approach has the advantage of producing a combined out-of-sample period of 10,000 observations, which provides enough data for an extensive empirical analysis.

Given this set of statistical mispricings, we apply our methodology to construct a forecasting model for each mispricing and a trading strategy to convert the mispricing return into a trading position. In addition, given the length of the data set, it is not practical to optimise the forecast horizon so for these experiments we restrict the forecasting horizon to a single time period of one hour. A simple forecasting model was considered sufficient to demonstrate the advantages of our approach. This was specified using an exponential smoothing model, which only requires one model parameter. The forecasting model was considered to have two characteristics that influence trading performance, namely predictive accuracy and prediction smoothness, which were measured using predictive correlation and prediction autocorrelation. We assumed stable transaction costs and applied a set of five trading rules which were specified as follows:

- Heuristic rule 1 (H1) – "sign" rule, equation (5.2) with $k = 0$
- Heuristic rule 2 (H2) – "linear proportional" rule, equation (5.2) with $k = 1$
- Parameterised Rule 1 - exponential smoothing rule with parameter θ, equation (5.3)
- Parameterised Rule 2 - simple moving average rule with parameter h, equation (5.4)
- Parameterised Rule 3 - step size rule with parameter λ, equation (5.5)

These trading systems are be optimisation in two ways, either using single or joint optimisation. Single optimisation involves simply optimising the forecasting model to maximise predictive correlation and then optimising the parameterised trading rule. In the case of joint optimisation, we apply our methodology in order to trade-off the two statistical characteristics of predictive correlation and prediction autocorrelation.

A general trading structure is implemented for a trading system which includes a controllable level of transaction costs. The trading performance is measured in terms of cumulative profit and annualised Sharpe Ratio. For these empirical evaluations we specified the optimisation criterion as the Sharpe Ratio, which we considered to be representative of the goals of risk adverse trading strategies. An example of one of the 50 identified statistical mispricings is defined as follows:

$$M_t = NAM - 63.334 - 0.182 * GLXO + 0.206 * HFX - 0.238 * UU \tag{8.7}$$

where *NAM, GLXO* and *HFX* and *UU* represent the price of stock in Nycomed Amersham, Glaxo Plc., Halifax Group and United Utilities respectively.

The analysis produced for this mispricing is shown in figures 8.3 and 8.4 with a summary of the results in table 8.1.

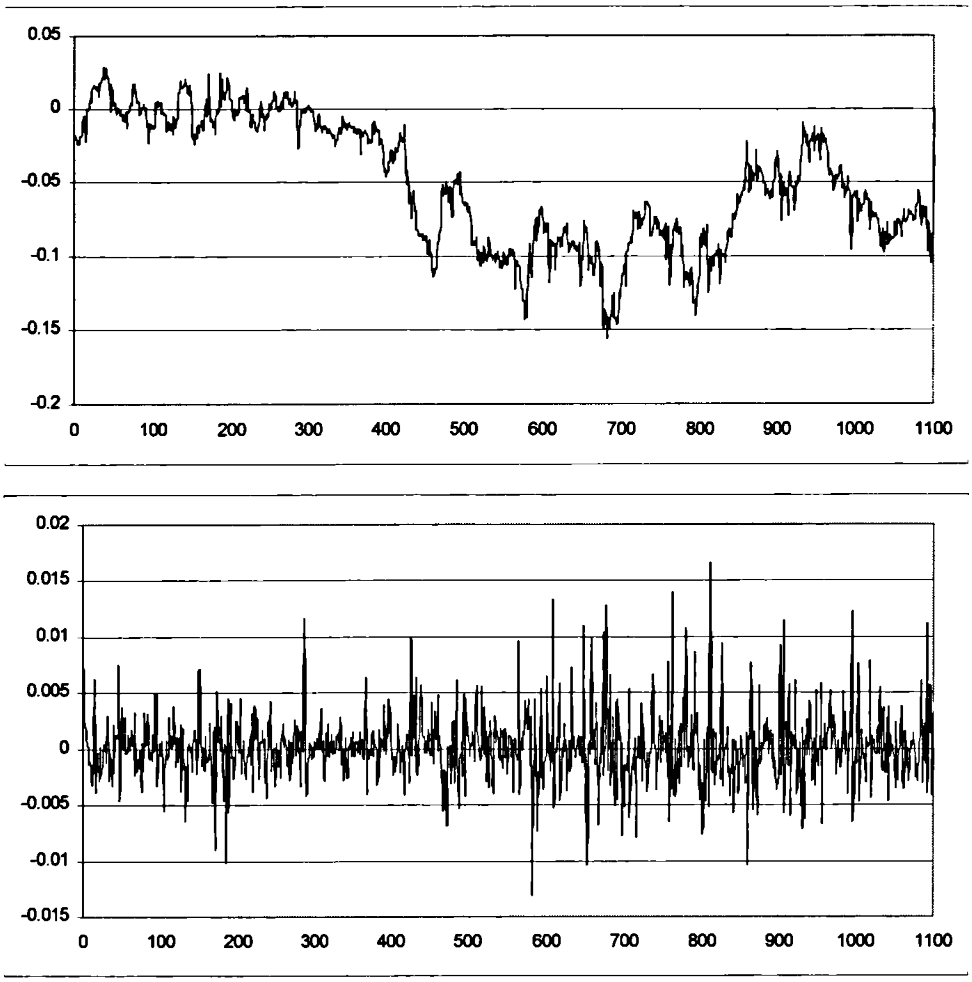


Figure 8.3 depicts an example of a statistical mispricing (top) and the predicted mispricing returns from a forecasting model (bottom).

The top graph shows the statistical mispricing for the assets described in equation (8.7). By inspection, this indicates that the series has some mean-reverting behaviour over the in-sample data set. The bottom diagram shows the predicted returns from a forecasting model constructed using an exponential smoothing model as described in equation (8.4). We specify the optimisation criteria to minimise forecast error over the in-sample period. This results in a model parameter, α, of 0.610, which gives an in-sample predictive correlation of 23.8% and prediction autocorrelation of 14.8%. Trading strategies are then optimised for the three parameterised trading rules over the same in-sample period. This results in smoothing parameters of 0.95, 10 and 0.009 for the three parameterised trading rules, as described in equations (5.3), (5.4) and (5.5) respectively. The cumulative profit for the three parameterised rules and the two heuristics rules are shown in figure 8.4 for both in-sample and out-of-sample periods.

Figure 8.4 depicts the cumulative profit of the five trading rules for both in-sample (top) and out-of-sample periods (bottom).

The top graph depicts the cumulative profit over the in-sample period for the five trading rules. As expected, the three optimised trading rules (rules 3,4 and 5) outperform the two heuristic trading rules (rule 1 and 2) with PTR1 (rule 3) producing the highest performance with 35.07%. The in-sample performance ranking (from highest to lowest) is Rule 3,5,4,2 and 1. The graph shows that performance is consistent over the trading period with few sharp spikes to distort results. The bottom graph depicts the out-of-sample cumulative profit. This shows some variation in performance rankings compared to the in-sample period. The out-of-sample performance ranking (highest to lowest) is Rule 5,2,3,4,1. This highlights the need for further testing which can be achieved over longer data sets or across other mispricings. Table 8.1. summarises the performance of trading systems for different trading rules using both single or joint optimisation methods.

Rule	Method	Model Parameters		Performance			
				In-sample		Out-of-sample	
		Forecast	Trading	Profit	SR	Profit	SR
H1	single	0.610		-13.17%	-1.460	1.51%	0.431
H2	single	0.610		8.51%	0.425	39.07%	4.123
PTR1	single	0.610	0.950	35.07%	2.519	16.68%	3.534
	joint	0.050	0.100	39.45%	2.747	22.61%	4.180
PTR2	single	0.610	10	18.18%	1.324	10.38%	2.693
	joint	0.050	1	39.62%	2.717	23.96%	4.252
PTR3	single	0.610	0.009	28.62%	2.930	43.71%	8.064
	joint	0.550	0.008	54.31%	4.018	42.89%	6.930

Table 8.1 summaries the results of applying different trading systems to the statistical mispricing described in equation (8.7), with transaction costs of 0.2%.

The results show that over the in-sample period the two heuristic trading rules give poor in-sample performance, in terms of Sharpe Ratio, of –1.460 and 0.425. As expected, optimising the three path dependent trading rules using the single optimisation method improves in-sample performance with Sharpe Ratios of 2.519, 1.324 and 2.930 respectively. The joint optimisation method was implemented to trade-off predictive correlation and prediction autocorrelation. For the three parameterised trading rules this resulted in increased prediction smoothing, with θ dropping to 0.05, 0.05 and 0.55 respectively, and reduced trading rule smoothing, with PTR parameters falling to 0.1, 1 and 0.008 respectively. This change in the forecast model parameter to 0.05 and 0.55 reduces the predictive correlation to 14.3% and 23.7% respectively. However, the prediction autocorrelation increases from 14.8% to 88.6% and 20.2% respectively. The joint optimisation has the effect of boosting prediction autocorrelation at the expense of reducing predictive correlation and trading position smoothness. For each rule, this results in an increased in-sample performance of 2.747, 2.717 and 4.013, which equates to a percentage improvement of 9.05%, 105.14% and 37.13% respectively.

The out-of-sample performance is less conclusive with a wide variation in performance, however, on average, heuristic rules give a Sharpe Ratio of 2.28 compared to single optimisation of parameterised rules of 4.76 and jointly optimised rules of 5.12. For this example, the parameterisation of the trading rule increases performance by 88% and joint optimisation by an additional 8%. These results purely demonstrate the potential value of using parameterised trading rules and the joint optimisation method over a single mispricing. We completed a more extensive study by generating trading systems for a series of 50 statistical mispricings. The results of this study are summarised in table 8.2.

All (50)	**Performance**							
Trading	Single Optimisation				Joint Optimisation			
Rule	In-sample		Out-of-sample		In-sample		Out-of-sample	
	Profit	SR	Profit	SR	Profit	SR	Profit	SR
H1	-33.3%	-4.58	-13.7%	-5.24				
H 2	-20.3%	-1.67	-17.3%	-3.63				
PTR 1	22.8%	1.17	0.03%	0.89	32.7%	1.87	0.06%	1.54
PTR 2	12.7%	0.49	0.01%	0.18	30.3%	1.71	0.04%	1.21
PTR 3	36.5%	2.15	0.07%	0.69	51.9%	3.57	0.03%	0.88

Table 8.2 compares the performance, in terms of both cumulative profit and annualised Sharpe Ratio, of different trading rules and the two optimisation methods across all models with transaction cost of 0.2%

Table 8.2 shows the average in-sample and out-of-sample performance, in terms of both cumulative profit and annualised Sharpe Ratio, across all trading systems for 50 statistical mispricings, with transaction costs of 0.2%. The three parameterised trading rules produce positive performance both in-sample and out-of-sample compared to negative returns from the two heuristic rules. In each case there is slight deterioration in performance between in-sample and out-of-sample performance. In addition, the joint optimisation of both forecasting and trading outperforms the single optimisation approach. For PTR1 the average out-of-sample Sharpe Ratio increases from 0.89 to 1.54, which is a percentage increase of 73%. Similarly, for PTR2 performance increases from 0.18 to 1.21, which is a percentage improvement of 572%. For PTR3 performance increases from 0.69 to 0.88, which is a percentage improvement of 27%. Overall, joint optimisation improves average performance, across all three rules, by 106%. These results indicate using real data the advantage of our parameterised trading rules and also of our joint optimisation methodology. Further experiments were conducted to analyse the influence of the level of transaction costs, as given in table 8.3.

	Performance of Trading Rules (Sharpe Ratio)							
Costs	Single optimisation					Joint optimisation		
bps	H1	H2	PTR1	PTR2	PTR3	PTR1	PTR2	PTR3
0	3.11	4.09	3.86	3.83	3.71	3.44	3.25	2.82
10	-1.18	0.21	1.60	1.11	1.34	1.85	1.54	1.01
20	-5.23	-3.63	0.89	0.18	0.69	1.54	1.21	0.88
30	-8.81	-7.30	-0.26	-1.60	0.05	1.00	0.52	0.21
40	-11.84	-10.69	-1.23	-3.09	-0.52	1.00	0.80	-0.01
50	-14.32	-13.74	-2.23	-4.61	-1.12	0.99	0.72	-0.37
60	-16.33	-16.46	-3.24	-6.01	-1.49	0.80	0.57	-0.48
70	-17.95	-18.85	-4.31	-7.30	-1.98	0.72	0.53	-0.58

Table 8.3 compares the performance of the two optimisation methods and the 3 parameterised trading rules across all models for transaction costs up to 70 basis points.

Table 8.3 shows the average Sharpe Ratios out-of-sample for all 50 trading models across trading rules for both single and joint optimisation methods. For no trading costs, the parameterised trading rules do not provide any significant improvement over the heuristic rules. Similarly, joint optimisation provides no benefit over the single optimisation approach, as expected. As transaction costs increase, however, trading performance deteriorates across all trading strategies with the parameterised trading rules significantly outperforming heuristic trading rules for costs as low as 10 basis points. At the 10 basis point level, average performance of the heuristic rules is –0.97 compared to 1.35 for parameterised trading rules using single optimisation. In addition, joint optimisation enhances performance with average Sharpe Ratio increasing from 1.35 to 1.47, which is an 8.9% increase. These results illustrate that joint optimisation can significantly improve performance for non-zero transaction costs.

8.5 Summary

In this chapter we have applied our methodology to jointly optimise a trading strategy and a forecasting model to the problem of exploiting statistical mispricings within a group of assets. We have discussed the concept of statistical arbitrage trading for predictive models as an extension of classical arbitrage trading. Multivariate methods have been described for constructing a statistical mispricing based on PCA and cointegration analysis, and also tests for potential predictability. Given a statistical mispricing, we have described how statistical arbitrage trading systems can be constructed based on a pair of forecasting and decision models. Potential forecasting methods are described and also trading strategies based on parameterised trading rules and reinforcement learning in the presence of stable or varying trading conditions respectively.

We have conducted empirical experiments on a set of 50 identified statistical mispricings from the constituents of the FTSE 100 using hourly data collected from May 1998 to December 1998. We assume stable transaction costs and use a simple exponential smoothing model to forecast the change in the mispricing. For comparison purposes, five different trading strategies are implemented using two heuristic rules based on the "sign" or "magnitude" of the predicted return, in addition to the three path dependent trading rules, which were developed in chapter 5. The parameters of the forecasting model and the trading strategy are optimised using two methods: a single optimisation approach and our joint optimisation methodology. The single optimisation approach optimises the forecasting model to minimise forecast error and then optimise the selected trading strategy to maximise performance in terms of annualised Sharpe Ratio. The joint optimisation method assumes that the optimisation criteria for the forecasting model has two relevant characteristics: predictive correlation and prediction autocorrelation. A meta parameter is used to jointly optimise the forecasting model and the trading strategy, as described in chapter 6.

Overall, the results show that trading strategies can be developed for statistical mispricings that lead to promising trading performance. As expected, profitability is influenced by transaction costs, with an increased level of costs leading to deterioration in trading performance. However, the three path dependent trading rules consistently outperform the two heuristic trading rules with additional performance improvements gained from the joint optimisation methodology. For transaction costs of 10 basis points, the average Sharpe Ratio is increased from –0.97 to 1.35 by using parameterised trading rules. Furthermore, joint optimisation increases performance from 1.35 to 1.47, an increase of 8.9%. We believe that these results on real data convincingly demonstrate the advantages of our approach.

9 Conclusions

In this thesis we have proposed and developed decision technologies for trading predictability in financial markets. We have presented novel learning techniques that are tailored to implement and optimise trading strategies, which offer advances upon the existing state of the art techniques. More specifically, we have developed modelling techniques to optimise a trading strategy for a forecasting model of asset returns based on both parameterised trading rules and reinforcement learning depending on the trading conditions. Furthermore, we have developed a methodology to jointly optimise a forecasting model and a trading strategy in order to maximise trading performance.

The basis for this work is that predictability exists in financial markets and that forecasting models can be constructed to capture these deterministic components of asset price dynamics. We justify this approach on the basis of a review of the theoretical arguments and empirical experiments that seem to suggest that although financial markets are relatively efficient there is significant evidence of some degree of predictability. In order to exploit predictability we propose using a two-stage modelling framework for the two tasks of forecasting and trading. We compare this approach to trading systems which use a single model for both of these tasks. We indicate using simulation techniques and statistical analysis that separate models for the two tasks of forecasting and trading typically outperform single model systems under realistic assumptions.

One of the main themes of this thesis is the development of "intelligent" decision modelling tools for trading predictability. In particular, we have developed a framework for optimising trading rules typically used as profitability tests for forecasting models of asset returns. A class of parameterised trading rules has been devised which encompasses two commonly used heuristic rules which capture information of the sign and magnitude of the predicted return. In this approach, two parameters are used to control the shape and magnitude of the trading position given a predicted return. Furthermore, this class of trading rules has been extended to "path dependent" trading rules in order to optimise trading in the presence of significant but stable transaction costs. Experiments using Monte-Carlo simulation were conducted to investigate the performance improvements achieved through optimising these rules against heuristic techniques. We have demonstrated, through controlled simulations, that these trading strategies can be optimised to maximise profitability given some level of predictability inherent in a generic forecasting model. The performance enhancements are explained in terms of a bias-variance like trade-off between exploiting predictability and minimising trading costs.

In order to optimise trading strategies for predicted returns in the presence of varying transaction costs we develop an explicit sequential decision model using reinforcement learning. In this framework we presume that trading costs can be influenced by frictional forces which are inherent in financial markets, such as market impact which is not traditionally reflected in decision models for trading systems. We propose a methodology based on reinforcement learning which optimises trading positions given state information from predictions, trading positions and other exogenous factors. More specifically, a Q-learning algorithm is modified to take account of the independence between asset prices and trading decisions to learn across a region of possible states in each time period. We use simulations to investigate the effects of market impact, in the form of a trading restriction, on the profitability of optimised trading strategies. Results show that our RL model, with a neural network acting as a function approximator, significantly outperforms the optimal "myopic" policy in the presence of a trading restriction.

Another important consideration in this thesis is the construction of forecasting models for trading. We present a meta parameter approach to control the design of the forecasting model in order to optimise the model conditionally given a particular trading strategy. In particular, we focus on controlling the optimisation criteria of the forecasting model and the forecast horizon which indirectly determine the performance of a trading system. We illustrate, using synthetic examples, how these two model design factors may be controlled using an iterative optimisation procedure. Our results show that conditional optimisation of a forecasting model for a given trading strategy can significantly improve trading performance in the presence of transaction costs.

Furthermore, the two modelling stages are combined into a joint optimisation methodology which forms an integrated modelling approach to encompass the implicit construction of statistical forecasting models with the explicit development of models for approximating optimal trading strategies for predicted returns. We examine the inter-related nature of the two tasks of forecasting and decision-making and develop a procedure to jointly optimise the two modelling stages in order to maximise trading performance. The modelling process to jointly optimise the construction of forecasting models and decision models offers a significant methodological step forward in the development of stochastic control systems for the dynamic trading of predictive models. We demonstrate, using simulation experiments, that joint optimisation can significantly improve performance against the traditional single step optimisation.

In addition to simulation experiments, we have demonstrated the potential of the developed decision modelling technologies and our joint optimisation approach to a particular form of trading commonly referred to as statistical arbitrage. This is an extension of traditional riskless arbitrage which is based on explicit relationships between financial assets. We conducted an extensive empirical assessment of our techniques to realistic examples of statistical mispricings identified within financial markets. The purpose was to provide a practical perspective to this work to complement controlled simulation experiments. A general trading structure was employed which allows fixed transaction costs to be taken into account and so explicitly provides a means of assessing the economic advantage of our developed techniques. The value of our analysis in more complex strategies may be approximated through leveraging results or overlaying existing investment strategies.

A rigorous methodology was employed to construct and detect statistical mispricings using a cointegration framework. This was achieved by collecting and processing hourly data from the constituents of the FTSE 100 over a 7-month period. The data was split into in sample and out-of-sample periods and 50 statistical mispricings, which each showed significant mean-reverting behaviour, were identified over this former data set. An exponential smoothing model is used to predict the change in the mispricing and the path dependent trading rules employed to optimise the trading strategy. The forecasting model and trading strategies are optimised using both a standard single optimisation approach which optimises predictive accuracy and our joint optimisation approach which trades-off two predictive characteristics, namely predictive accuracy and prediction smoothness. For these trading systems, the performance of our developed parameterised trading rules is compared against commonly used heuristic rules. The results across all models on out-of-sample data are highly promising with path dependent trading rules significantly outperforming heuristic rules. For transaction costs of 10 basis points, the Sharpe Ratio was increased from 0.97 to 1.35. Our results for the joint optimisation approach are less impressive but still significant with the average Sharpe Ratio increased from 1.35 to 1.46, which is an increase of 8.9%

These empirical results indicate the importance of our decision modelling techniques for exploiting predictability in financial markets. It should be noted, however, that although the empirical results are impressive the true influence of a real trading environment may not been fully taken into account. We believe that the true value of these techniques can only tested in a real live trading environment, where actual trades are completed using real transaction prices and costs. However, we believe that the methods developed in this thesis have wide trading applications and offer significant potential for active fund managers, arbitrageurs and hedge fund managers.

From a methodological perspective we believe that the work in this thesis raises a number of potential areas of future research. In particular, it is considered that interesting further work could be completed by further developing reinforcement learning techniques to the learning of trading strategies in imperfect markets. In our simulation studies we only considered the standard Q-learning algorithm with fixed parameter values, which is sufficient to show the key advantages of RL. However, more extensive experiments would be worthwhile to enhance and tailor these learning methods and apply them to more specific trading systems.

Another area of potential research is in the development of joint optimisation methods for forecasting models and trading strategies. In particular, we consider the integration of other forecasting techniques, in addition to simple smoothing models, and also the control of more model design factors are interesting avenues of further research. In addition, it is thought that the development of more complex systems, involving multiple design factors, may justify the implementation of more advanced, heuristic optimisation techniques.

So, in conclusion, we believe that the exploitation of predictability in financial markets requires the rigorous modelling and integration of all aspects of the decision-making process. Simply concentrating on developing forecasting models and employing heuristic trading rules may not be sufficient, especially as predictability is often low in financial markets and where trading profits are highly sensitive to transaction costs. From this perspective, the development of reliable modelling tools for trading predictability is an important area of research for both academics and practitioners. In this respect, we have made some progress in developing an integrated modelling framework for trading systems and applying new decision modelling techniques to this fascinating area of investment finance.

Appendix A – Table of Simulation Results

Iterations	Forecasting Model				Characteristics		PTR	Utility
	α	*Cost Fun.*	θ_1	θ_2	β	φ	θ	*Ratio*
initial	1.0000	0.3000	0.2000	0.3000	0.3000	0.0875	-	0.0947
iter 1 stage1	1.0000	0.3000	0.2000	0.3000	0.3000	0.0875	0.4799	0.1223
iter 1 stage2	0.9041	0.3083	0.2745	0.4196	0.2801	0.5734	0.4799	0.1698
iter 2 stage1	0.9041	0.3083	0.2745	0.4196	0.2801	0.5734	0.4930	0.1699
iter 2 stage2	0.8878	0.3134	0.2839	0.4324	0.2754	0.6140	0.4930	0.1723
iter 3 stage1	0.8878	0.3134	0.2839	0.4324	0.2754	0.6140	0.5011	0.1723
iter 3 stage2	0.8771	0.3172	0.2898	0.4401	0.2723	0.6375	0.5011	0.1735
iter 4 stage1	0.8771	0.3172	0.2898	0.4401	0.2723	0.6375	0.5053	0.1735
iter 4 stage2	0.8695	0.3200	0.2938	0.4452	0.2701	0.6527	0.5053	0.1741
iter 5 stage1	0.8695	0.3200	0.2938	0.4452	0.2701	0.6527	0.5060	0.1741
iter 5 stage2	0.8639	0.3222	0.2966	0.4488	0.2685	0.6631	0.5060	0.1744
iter 6 stage1	0.8639	0.3222	0.2966	0.4488	0.2685	0.6631	0.5088	0.1744
iter 6 stage2	0.8596	0.3239	0.2988	0.4515	0.2673	0.6707	0.5088	0.1746
iter 7 stage1	0.8596	0.3239	0.2988	0.4515	0.2673	0.6707	0.5106	0.1746
iter 7 stage2	0.8563	0.3253	0.3004	0.4536	0.2663	0.6765	0.5106	0.1748
iter 8 stage1	0.8563	0.3253	0.3004	0.4536	0.2663	0.6765	0.5125	0.1748
iter 8 stage2	0.8537	0.3263	0.3017	0.4551	0.2656	0.6808	0.5125	0.1749
iter 9 stage1	0.8537	0.3263	0.3017	0.4551	0.2656	0.6808	0.5131	0.1749
iter 9 stage2	0.8517	0.3272	0.3026	0.4563	0.2650	0.6842	0.5131	0.1749
Iter 10 stage1	0.8517	0.3272	0.3026	0.4563	0.2650	0.6842	0.5137	0.1749
iter 10 stage2	0.8501	0.3279	0.3034	0.4573	0.2646	0.6867	0.5137	0.1750
iter 11 stage1	0.8501	0.3279	0.3034	0.4573	0.2646	0.6867	0.5144	0.1750
iter 11 stage2	0.8488	0.3284	0.3040	0.4580	0.2642	0.6887	0.5144	0.1750
iter 12 stage1	0.8488	0.3284	0.3040	0.4580	0.2642	0.6887	0.5138	0.1750
iter 12 stage2	0.8478	0.3288	0.3045	0.4586	0.2639	0.6903	0.5138	0.1750
iter 13 stage1	0.8478	0.3288	0.3045	0.4586	0.2639	0.6903	0.5148	0.1750
iter 13 stage2	0.8470	0.3291	0.3048	0.4591	0.2637	0.6916	0.5148	0.1750
iter 14 stage1	0.8470	0.3291	0.3048	0.4591	0.2637	0.6916	0.5151	0.1750
iter 14 stage2	0.8464	0.3294	0.3051	0.4594	0.2635	0.6926	0.5151	0.1750
iter 15 stage1	0.8464	0.3294	0.3051	0.4594	0.2635	0.6926	0.5149	0.1750
iter 15 stage2	0.8459	0.3296	0.3054	0.4597	0.2634	0.6934	0.5149	0.1750
iter 16 stage1	0.8459	0.3296	0.3054	0.4597	0.2634	0.6934	0.5149	0.1750
iter 16 stage2	0.8455	0.3298	0.3056	0.4600	0.2633	0.6940	0.5149	0.1751
iter 17 stage1	0.8455	0.3298	0.3056	0.4600	0.2633	0.6940	0.5150	0.1751
iter 17 stage2	0.8451	0.3300	0.3057	0.4602	0.2632	0.6945	0.5150	0.1751
iter 18 stage1	0.8451	0.3300	0.3057	0.4602	0.2632	0.6945	0.5152	0.1751
iter 18 stage2	0.8449	0.3301	0.3059	0.4603	0.2631	0.6949	0.5152	0.1751
iter 19 stage1	0.8449	0.3301	0.3059	0.4603	0.2631	0.6949	0.5156	0.1751
iter 19 stage2	0.8447	0.3302	0.3060	0.4604	0.2630	0.6953	0.5156	0.1751
iter 20 stage1	0.8447	0.3302	0.3060	0.4604	0.2630	0.6953	0.5154	0.1751
iter 20 stage2	0.8445	0.3303	0.3060	0.4605	0.2630	0.6955	0.5154	0.1751

iter 21 stage1	0.8445	0.3303	0.3060	0.4605	0.2630	0.6955	0.5151	0.1751
iter 21 stage2	0.8443	0.3303	0.3061	0.4606	0.2629	0.6958	0.5151	0.1751
iter 22 stage1	0.8443	0.3303	0.3061	0.4606	0.2629	0.6958	0.5151	0.1751
iter 22 stage2	0.8442	0.3304	0.3062	0.4607	0.2629	0.6959	0.5151	0.1751
iter 23 stage1	0.8442	0.3304	0.3062	0.4607	0.2629	0.6959	0.5150	0.1751
iter 23 stage2	0.8441	0.3304	0.3062	0.4607	0.2629	0.6961	0.5150	0.1751
iter 24 stage1	0.8441	0.3304	0.3062	0.4607	0.2629	0.6961	0.5149	0.1751
iter 24 stage2	0.8441	0.3304	0.3062	0.4608	0.2629	0.6962	0.5149	0.1751
iter 25 stage1	0.8441	0.3304	0.3062	0.4608	0.2629	0.6962	0.5148	0.1751
iter 25 stage2	0.8440	0.3305	0.3063	0.4608	0.2628	0.6963	0.5148	0.1751
iter 26 stage1	0.8440	0.3305	0.3063	0.4608	0.2628	0.6963	0.5149	0.1751
iter 26 stage2	0.8440	0.3305	0.3063	0.4608	0.2628	0.6963	0.5149	0.1751
iter 27 stage1	0.8440	0.3305	0.3063	0.4608	0.2628	0.6963	0.5150	0.1751
iter 27 stage2	0.8439	0.3305	0.3063	0.4609	0.2628	0.6964	0.5150	0.1751
iter 28 stage1	0.8439	0.3305	0.3063	0.4609	0.2628	0.6964	0.5150	0.1751
iter 28 stage2	0.8439	0.3305	0.3063	0.4609	0.2628	0.6964	0.5150	0.1751
iter 29 stage1	0.8439	0.3305	0.3063	0.4609	0.2628	0.6964	0.5151	0.1751
iter 29 stage2	0.8439	0.3305	0.3063	0.4609	0.2628	0.6965	0.5151	0.1751
iter 30 stage1	0.8439	0.3305	0.3063	0.4609	0.2628	0.6965	0.5151	0.1751
iter 30 stage2	0.8438	0.3305	0.3063	0.4609	0.2628	0.6965	0.5151	0.1751
iter 31 stage1	0.8438	0.3305	0.3063	0.4609	0.2628	0.6965	0.5151	0.1751
iter 31 stage2	0.8438	0.3305	0.3064	0.4609	0.2628	0.6966	0.5151	0.1751
iter 32 stage1	0.8438	0.3305	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 32 stage2	0.8438	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 33 stage1	0.8438	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 33 stage2	0.8438	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 34 stage1	0.8438	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 34 stage2	0.8438	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 35 stage1	0.8438	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 35 stage2	0.8438	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 36 stage1	0.8438	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 36 stage2	0.8438	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 37 stage1	0.8438	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 37 stage2	0.8437	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 38 stage1	0.8437	0.3306	0.3064	0.4609	0.2628	0.6966	0.5152	0.1751
iter 38 stage2	0.8437	0.3306	0.3064	0.4609	0.2628	0.6967	0.5152	0.1751
iter 39 stage1	0.8437	0.3306	0.3064	0.4609	0.2628	0.6967	0.5152	0.1751
iter 39 stage2	0.8437	0.3306	0.3064	0.4609	0.2628	0.6967	0.5152	0.1751
iter 40 stage1	0.8437	0.3306	0.3064	0.4609	0.2628	0.6967	0.5152	0.1751
iter 40 stage2	0.8437	0.3306	0.3064	0.4610	0.2628	0.6967	0.5152	0.1751

Bibliography

Abu-Mostafa, Y. S., 1995, Financial market applications of learning from hints, in A. N. Refenes (ed), *Neural Networks in the Capital Markets*, John Wiley and Sons Ltd., Chichester, 221-232

Akaike, H., 1973, Information theory and an extension of the maximum likelihood principle, in *Second International Symposium on Information Theory* (ed. B. N. Petrov and F. Czaki), Akademiai Kiado, Budapest, 267-81.

Akaike, H., 1974, A New Look at the Statistical Model Identification, *IEEE Transactions on Automatic Control*, AC-19, 716-723.

Almgren, R., and Chriss. N., 1999, Optimal Execution of Portfolio Transactions.

Back, A. D., and Weigend, A. S., 1998, Discovering structure in finance using independent component analysis, in A-P. N. Refenes *et al* (Eds.) *Decision Technologies for Computational Finance*, Kluwer Academic Publishers, Dordrecht., 309-322.

Barto A. G., Sutton R. S., and Anderson C. N., 1993, Neuronlike elements that can solve difficult learning control problems, *IEEE transactions on Systems, Man and Cybernetics*, 13, 835-846.

Bates, J. M., and Granger, C. W. J., 1969, The combination of forecasts, *Operations Research Quarterly*, 20, 319-25

Bellman, R. E., 1957, *Dynamic Programming*, Princeton University Press, Princeton, NJ.

Bengio, Y., 1997, Training a neural network with a financial criterion rather than a prediction criterion, in Weigend et al (eds) *Decision Technologies for Financial Engineering*, World-Scientific, Singapore, 36-48.

Bentz, Y., 1997, *Factor models in equity investment management: a review*, Working Paper, Decision Technology Centre, London Business School.

Bentz, Y., 1999, *Identifying and modelling conditional factor sensitivities: an application to equity investment management*, Unpublished PhD Thesis, London Business School.

Bentz, Y., and Connor, J. T., 1998, Unconstrained and constrained time-varying factor sensitivities in equity investment management, in A-P. N. Refenes *et al* (Eds.) *Decision Technologies for Computational Finance*, Kluwer Academic Publishers, Dordrecht., 291-308.

Bentz, Y., Refenes, A. N., and De Laulanie, J-F., 1996, Modelling the performance of investment strategies, concepts, tools and examples, in Refenes et al (eds), *Neural Networks in Financial Engineering*, World Scientific, Singapore, 241-258

Bergerson K., and Wunsch D. C., 1991, A commodity trading model based on a neural network-expert system hybrid, *Proc. IEEE International Conference on Neural Networks,* 1991, 1289-1293, reprinted in (Trippi and Turban, 1993)

Bertsimas, D., and Lo. A. W., 1998, Optimal control of execution costs, *Journal of Financial Markets*, 1, 1-50.

Bishop, C. M., 1995, *Neural networks for Pattern Recognition*, Clarendon Press, Oxford

Black, F., 1976, The pricing of commodity contracts, *Journal of Financial Economics*, 3, 167-179.

Black, F., and Scholes, M., 1973, The pricing of options and corporate liabilities, *Journal of Political Economy,* 81, 637-654

Black F., 1986, Noise, *Journal of Finance*, 41, 529-544.

Bolland, P. J., 1998, *Robust neural estimation and diagnostics*, Unpublished PhD Thesis, London Business School.

Bolland, P. J., and Connor, J. T., 1996, Identification of FX arbitrage opportunities with a non-linear multivariate Kalman filter, in Refenes et al (eds), *Neural Networks in Financial Engineering*, World Scientific, Singapore, 122-134

Bollerslev, T., 1986, Generalised autoregressive conditional heteroskedasticity, *Journal of Econometrics*, 31, 307-328.

Bosarge, W. E., 1991, Adaptive processes to exploit the nonlinear structure of financial markets, Presented at the Santa Fe Institute of Complexity Conference: *Neural Networks and Pattern Recognition in Forecasting Financial Markets,* February 15, 1991, reprinted in (Trippi and Turban, 1993)

Box, G. E. P., and Jenkins, G. M., 1970, *Time Series Analysis: Forecasting and Control*, San Francisco: Holden-Day, (revised edn, 1976)

Box, G. E. P., and Pierce, D. A., 1970, Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association.*, **70**, 1509-26.

Bramante, R., Colombo R., and Gabbi. G., 1997, Are Neural Networks and Econometric Forecasts Good for Trading? Stochastic Variance as a Filter Rule, *Decision Technologies for Computational Finance*, Kluwer.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone C. J., 1984, *Classification and Regression Trees*, Wadsworth and Brooks/Cole, Monterey.

Breiman, L., 1996, Bagging predictors, *Machine Learning*, 24, 123-140

Brock, W., Lakonishok, J., and LeBaron, B, 1992, Simple Technical Trading Rules and the Stochastic Properties of Stock Returns, Journal of Finance, 47, 1731-1764.

Brown, R. G., 1963, *Smoothing, Forecasting and Prediction*, Prentice-Hall, Englewood Cliffs, NJ.

Brown, R. L., Durbin, J., and Evans, J. M., 1975, Techniques for testing the constancy of regression relationships over time (with discussion), *Journal of the Royal Statistical Society, series B,* 37, 149-192.

Bunn, D. W., 1989, Forecasting with more than one model, *Journal of Forecasting*, **8**, 161-166.

Burgess, A. N., 1995, Non-linear model identification and statistical significance tests and their application to financial modelling, in *IEE Proceedings of the 4th International Conference on Artificial Neural Networks*, Cambridge, 312-317

Burgess, A. N., 1996, Statistical yield curve arbitrage in eurodollar futures using neural networks, in Refenes et al (eds), *Neural Networks in Financial Engineering*, World Scientific, Singapore, 98-110

Burgess, A. N., 1997, Asset allocation across european equity indices using a portfolio of dynamic cointegration models, in Weigend et al (eds) *Decision Technologies for Financial Engineering*, World-Scientific, Singapore, 276-288.

Burgess, A. N., 1998, Controlling nonstationarity in statistical arbitrage using a portfolio of cointegration models, in A-P. N. Refenes *et al* (Eds.) *Decision Technologies for Computational Finance*, Kluwer Academic Publishers, Dordrecht., 89-107.

Burgess, A. N., 1999, *A Computational Methodology for Modelling the Dynamics of Statistical Arbitrage*, Unpublished PhD thesis, Decision Technology Centre, London Business School.

Burgess, A. N., and Towers N., 1998, Statistical Arbitrage Models for Equity and Fixed Income Markets, Unpublished Technical Report, HAT project, London Business School.

Burgess, A. N., and Refenes, A. N., 1996, Modelling non-linear cointegration in international equity index futures, in Refenes et al (eds), *Neural Networks in Financial Engineering*, World Scientific, Singapore, 50-63

Campbell, J, Y., Lo, A. W., and MacKinlay, A. C., 1999, *The Econometrics of Financial Markets*, Princeton University Press.

Choey, M., and Weigend, A. S., 1997, Nonlinear trading models through Sharpe Ratio Maximization, in Weigend et al (eds) *Decision Technologies for Financial Engineering*, World-Scientific, Singapore, 3-22.

Chow, K. V., and Denning, K. C., 1993, A simple multiple variance ratio test, *Journal of Econometrics*, vol 58, no 3, 385-401

Clare, A.D., Maras, M. and Thomas S.H. (1995) The integration and efficiency of international bond markets. *Journal of Business Finance and Accounting*, 22(2), 313-22.

Cohen, K., Maier, S., Schwartz, R., and Whitcomb, D., 1986, *The Microstruture of Securities Markets*, Prentice Hall, Englewood Cliffs, NJ.

Connor, J. T., Bolland, P. J., and Lajbcygier, P., 1997, Intraday modelling of the term structure of interest rates, in Weigend et al (eds) *Decision Technologies for Financial Engineering*, World-Scientific, Singapore, 225-232.

Cook, G., 1999, Transaction cost modelling for Portfolio Trading, Unpublished Technical Report, First Quadrant, London.

Cootner, P., 1974, The Random Character of Stock Market Prices, MIT Press.

Crites, R. H., and Barto, A. G., 1996, Improving elevator performance using reinforcement learning, in Touretzky, D. S., *et al* (Eds.) *Advances in Neural Information Processing Systems 8*, MIT Press, Cambridge MA, 430-436.

DeBondt, W., and Thaler, R., 1985, Does the Stock Market Overact? *Journal of Finance*, 40, 793-805.

DeBondt, W., and Thaler, R., 1987, Further Evidence on Investor Overreaction and Stock Market Seasonality, *Journal of Finance*, 42, 557-582.

DeLong, B., Shleifer A., Summers, L., and Waldmann, R., 1989, Positive Feedback Investment Strategies and Destabilising Rational Speculation, Working Paper 2880, NBER.

Dempster, A. P., Laird, N. M, and Rubin, D. E., 1977, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society B.*, 39, 1-38.

Dickey, D. A. and Fuller, W. A., 1979, Distribution of the estimators for autoregressive time-series with a unit root, *Journal of the American Statistical Association*, 74, 427-431.

Diebold, F. X., 1998, *Elements of Forecasting*, South-Western College Publishing, Cincinnati, Ohio

Dutta, S., and Shashi, S., 1988, Bond rating: A non-conservative application of neural networks, *in Proc. ICNN-88 San Diego,* 24-27 July 1988, Vol. II, 443-450

Eckbo, B. E., and Liu, J., 1993, Temporary Components of Stock Prices: New Univariate Results, *Journal of Financial and Quantitative Analysis,* Vol. 28, NO. 2, 161-176

Efron, B., and Tibshirani, R. J., 1993, *An Introduction to the Bootstrap,* Chapman and Hall, New York.

Elton, E. J., and Gruber, M. J., 1971, Dynamic Programming Applications in Finance, *The Journal of Finance*, Vol 26. No. 2.

Engle, R. F., 1982, Autoregressive conditional heteroskedasticity with estimates of the variance of the UK inflation, *Econometrica*, 50, 987-1007.

Engle, R. F., and Brown, S. J., 1986, Model Selection for Forecasting, *Applied Mathematics and Computation*, 20, 313-327

Engle, R. F., and Granger, C. W. J., 1987, Cointegration and error-correction: representation, estimation and testing, *Econometrica*, 55, 251-276

Fama E., 1965, The Behaviour of Stock Market Prices, *Journal of Business*, 38, 34-105.

Fama E., 1970, Efficient Capital Markets: A review of Theory and Empirical Work, *Journal of Finance*, 25, 383-417.

Fama E., 1991, Efficient Capital Markets II, Journal of Finance, Vol. 26, No. 5, pp. 1575-1617.

Fama E., and MacBeth J., 1973, Risk, Return and Equilibrium Tests, *Journal of Political Economy*, Vol 81, No. 3, pp. 607-636.

Fama E., and Blume M., 1966, Filter Rules and stock Market Trading Profits, *Journal of Business*, 39, 226-241.

Fisher, L., 1966, Some New Stock Market Indexes, *Journal of Business*, 39, 191-225.

Fletcher R., 1987, *Practical Methods for Optimisation.* Wiley

French, K. R., 1980, Stocks returns and the Weekend Effect, *Journal of Financial Economics*, Vol. 8, pp. 55-70.

Friedman, J.H. and Stuetzle, W., 1981. Projection pursuit regression. *Journal of the American Statistical Association.* Vol. 76, pp. 817-823.

Friedman, J.H., 1991. Multivariate Adaptive Regression Splines (with discussion). *Annals of Statistics.* Vol 19, num. 1, pp. 1-141.

Fuller, W. A., 1976, *Introduction to Statistical Time Series*, New York: Wiley.

Gatev E. and Ross S. A., 2000, Rebels, Conformists, Contrarians and Momentum Traders, National Bureau of Economic Research, Working Paper 7835.

Gardner, E. S., 1985, Exponential smoothing: The state of the art (with discussion), *Journal of Forecasting*, 4, 1-38.

Gill, P. E., and Murray W. (ed.), 1974. *Numerical methods for Constrained Optimisation*, Academic Press.

Gibbons M. R. and Hess P. J., 1981, Day of the Week Effects and Asset Returns, Journal of Business, Vol 54, pp. 579-596.

Gibbons M. R., 1982, Multivariate Tests of Financial Models: A New Approach, *Journal of Financial Economics*, 14, 217-236.

Gonzalez-Miranda, F. and Burgess, A. N., 1993, Modelling implied volatilities using neural networks, *presented at First International Workshop on Neural Networks in the Capital Markets,* November 18-19, 1993, London

Granger, C. W. J., 1989, Combining Forecasts – Twenty Years Later, *Journal of Forecasting*, **8**, 167-173.

Greene, W. H., 1993, *Econometric Analysis*, Prentice-Hall, New Jersey.

Grossman S., 1976, On the Efficiency of Competitive Stock Markets where Trades have Diverse Information, *Journal of Finance*, 31, 573-585.

Grossman S., and Stiglitz, J., 1980, On the possibility of Informationally Efficient Markets, *American Economic Review*, 70, 3930408.

Gultekin M. N. and Gultekin N. B., 19883, Stock Market Seasonality: International Evidence, Journal of Financial Economics, Vol. 12, pp.469-481.

Haerdle, W., 1990. *Applied nonparametric regression*. Cambridge University Press.

Harris L., 1986, A Transaction Data Study of Weekly and Intra-daily Patterns in Stock Returns, Journal of Financial Economics, Vol. 14, pp.99-117.

Harris D. and Inder B, 1994, "A test of the null hypothesis of cointegration", pp 133-152 in *Nonstationary Time Series Analysis and Cointegration,* Oxford University Press (Ed. C. P. Hargreaves)

Harrison, P. J. and Stevens C. F., 1971, A Bayesian approach to short-term forecasting, *Operational Research Quarterly*, 22, 341-362

Harvey, A. C., 1989, *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press, Cambridge.

Harvey, A. C., 1993, *Time Series Models,* second edition, Harvester Wheatsheaf, London.

Hatanaka, M., 1975, On the global identification of the dynamic simultaneous equations model with stationary disturbances, *international Economic Review*, 16, 545-554

Holt, C. C., 1957, Forecasting seasonals and trends by exponentially weighted moving averages, *ONR Research Memorandum No 52, Carnegie Institute of Technology.*

Hull, J. C., 1993, *Options, Futures and other derivative securities*, Prentice-Hall

Hutchinson, J., Lo, A. and Poggio, T., A non-parametric approach to pricing and hedging derivative securities via learning networks, *Journal of Finance,* XLIX:3 (July 1994).

Jacobs, B. I., and Levy, K. N., 1988, Disentangling Equity Return Regularities: New Insights and Investment Opportunities, *Financial Analysts Journal*, May-June 1988, 18-43.

Jennergren R. and Korsvold P., 1975, The Non-Random Character of Norwegian and Swedish Stock Market Prices, in Elton and Gruber, *International Capital Markets*, Amsterdam: North-Holland.

Johansen, S., 1988, Statistical analysis of cointegration vectors, *Journal of Economic Dynamics and Control*, 12, 131-154.

Johansen, S., 1991, "Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models", *Econometrica,* Vol 59, 1551-81

Jolliffe, I. T., 1986, *Principal Component Analysis*, Springer-Verlag, New York.

Jones C. D., Pearce O. K., and Wilson J. W., 1987, Can Tax-Loss Selling explain the January Effect? A Note, *Journal of Finance*, Vol. 42, No. 2,, pp. 453-461.

Kato K. and Shallheim, J. 1985, Seasonal and Size Anomalies in the Japanese Stock Market, *Journal of Financial and Quantitative Analysis*, Vol. 20, No. 2, pp. 243-260.

Kalman, R. E., 1960, A new approach to linear filtering and prediction problems, *Journal of Basic Engineering, Transactions ASME, Series D,* 82, 35-45.

Kalman, R. E. and Bucy R. S., 1961, New results in linear filtering and prediction theory, *Journal of Basic Engineering, Transactions ASME, Series D,* 83, 95-108

Kollias C. and Metaxas, K., 1997, Selecting Relative Value Stocks with Nonlinear Cointegration, *Decision Technologies for Computational Finance*, Kluwer.

Lakonishok, J. and Smidt. S., 1988, Are Seasonal Anomalies Real? A Ninety Year Perspective, *Review of Financial Studies*, Vol 1, pp. 435-445.

Lakonishok, J. and Shapiro A. C., 1986, Systematic Risk, Total Risk and Size as Determinants of Stock Market Returns, *Journal of Business Finance*, Vol. 10, No. 1, pp. 115-132.

Lapedes A. and Farber, R., 1987, Nonlinear signal processing using neural networks, *Proc. IEEE Conference on Neural Information Processing Systems - Natural and Synthetic*

LeBaron, B., 1996, Technical Trading Rule Profitability and Foreign Exchange Intervention, Working Paper, 5505, NBER, Cambridge, MA.

Lee, T-H., White, H. and Granger, C. W. J., 1993, Testing for neglected nonlinearity in time series models, *Journal of Econometrics*, 56, 269-290

Lehmann, B. N., 1990, Fads, Martingales and Market Efficiency, *Quarterly Journal of Economics*, 105, 1-28.

LeRoy S. F., 1973, Risk Aversion and the Martingale Property of Stock Returns, *International Economic Review*, 14, 436-446

Levy, R., 1967, Relative strength as a Criterion for Investment Selection, *Journal of Finance*, Vol. 22, pp. 595-610.

Lo A. W. and MacKinlay A. C., 1988, Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test, *The Review of Financial Studies*, Vol 1, No. 1, 41-66

Lo A. W. and MacKinlay A. C., 1989, The Size and Power of the Variance Ratio Test in Finite Samples: A Monte Carlo Investigation, *Journal of Econometrics*, **40**, 203-238.

Lo, A. W. and MacKinlay, A. C., 1990, Data-Snooping Biases in Tests of Financial Asset Pricing Models, *Review of Financial Studies*,Vol.3, No.3

Lo, A. W., and MacKinlay, A. C., 1995, Maximising predictability in the stock and bond markets, Working Paper No. 5027, National Bureau of Economic Research

Lo, A. W., and MacKinlay, A. C., 1999, *A Non-Random Walk down Wall Street*, Princeton University Press.

Lucas, R. E., 1978, Asset Prices in an Exchange Economy, *Econometrica*, 46, 1429-1446

Lyung, G. M. and Box, G. E. P., 1978, On a measure of lack of fit in time series models. *Biometrika*, **65**, 297-304.

Markellos, R. N., 1997, *Nonlinear Equilibrium Dynamics*, Working Paper 97/6, Department of Economics, Loughborough University, UK.

Markowitz, H. M., 1952, Portfolio Selection, *Journal of Finance*, 7, 77-91.

Markowitz, H. M., 1959, *Portfolio Selection: Efficient Diversification of Investments*. John Wiley and Sons, New York.

Meier, D. C., Pfeifer, R., Demostene, R. and Scheier C., 1993, Is mean-reversion on stock indices a linear artifact ?, *Proc. First International Workshop on Neural Networks in the Capital Markets,* November 18-19, 1993, London Business School

Merton, R. C., 1980, On Estimating the Expected Return on the Market: An Exploratory Investigation, *Journal of Financial Economics*, Vol 8 (4), pp. 323-361.

Moody J. E, 1992, The effective number of parameters: an analysis of generalisation and regularization in nonlinear learning systems, in J. E. Moody, S. J. Hanson and R. P. Lipmann (eds), *Advances in Neural Information Processing Systems 4*, 847-54, Morgan Kaufmann, San Mateo, US

Moody, J. E., and Saffell, M., 1999, Minimizing Downside Risk via Stochastic Dynamic Programming, to appear in Y. S. Abu-Mostafa *et al* (Eds.) *Computational Finance – Proceedings of the Sixth International Conference*, Leonard N. Stern School of Business, January 1999.

Moody, J. E., and Utans, J., 1992, Principled architecture selection for neural networks: Application to corporate bond rating prediction, in J. E. Moody, S. J. Hanson and R. P. Lipmann (eds), *Advances in Neural Information Processing Systems 4*, Morgan Kaufmann Publishers, San Mateo, CA, 683-690

Moody, J. E., and Wu, L., 1994, Statistical analysis and forecasting of high frequency foreign exchange rates, *Proc. Neural Networks in the Capital Markets,* Pasadena, November 16-18, 1994

Moody, J. E., and Wu, L., 1997, Optimization of trading systems and portfolios, in Weigend et al (eds) *Decision Technologies for Financial Engineering*, World-Scientific, Singapore, 23-35.

Moody, J. E., and Wu, L., 1997b, What is the "true price"? - state space models for high-frequency FX data, in Weigend et al. (eds) *Decision Technologies for Financial Engineering*, World-Scientific, Singapore, 346-358.

Moody, J. E., Wu, L., Liao, Y., and Saffell, M., 1998, Performance Functions and Reinforcement Learning for Trading Systems and Portfolios, *Journal of Forecasting,* Vol. 17, 441-470.

Moody, J. E., Saffell, M., Liao, Y., and Wu, L. 1997 Reinforcement Learning for Trading Systems and Portfolios: Immediate vs Future Rewards, in Refenes et al. (eds) *Decision Technologies for Computation Finance*, Kluwer Academic Publishers, 129-140.

Muth, J. F., 1960, Optimal properties of exponentially weighted forecasts, *Journal of the American Statistical Association*, 55, 299-305

Nelson, C. R. and Kang, H., 1981, Spurious periodicity in inappropriately detrended series. *Econometrica*, 49, 741-751

Nelson, D., 1991, Conditional heteroskedasticity in asset returns: A new approach, *Econometrica*, 59(2), 347-370.

Nerlove, M. and Wage, S., 1964, On the optimality of adaptive forecasting, *Management Science*, 10, 207-224.

Neuneier, R., 1996, Optimal Asset Allocation using Adaptive Dynamic Programming, *Advances in Neural Information Processing Systems 8*, Morgan Kaufmann Publishers, San Mateo, CA.

Neuneier, R., 1998, Enhancing Q-learning for Optimal Asset Allocation, *Advances in Neural Information Processing Systems 10*, 936-942, Morgan Kaufmann Publishers, San Mateo, CA.

Newbold, P., and Granger, C. W. J., 1974, Experience with forecasting univariate time-series and the

Poterba, J. and Summers, L., 1988, Mean Reversion in Stock Prices: Evidence and Implications, *Journal of Financial Economics*, Vol. 22, No. 1, pp. 27-59.

Puterman M. L., 1993, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley and Sons Ltd., Chichester.

Rauscher F. A., 1997, Multi-task learning in a neuralvector error correction appraoch for exchange rate forecasting, *Decision Technologies for Computational Finance*, Kluwer

Reinganum M. R., 1983, The Anomalous Stock Market Behaviour of Small Firms in January: Empirical Tests of Tax-Loss Selling Effect, *Journal of Financial Economics*, Vol. 12, pp.89-104.

Refenes, A. N., 1995, (ed) *Neural Networks in the Capital Markets*, John Wiley and Sons Ltd., Chichester.

Refenes A. N., and Azema-Barac, M., 1994, Neural network applications in financial asset management, *Neural Computing and Applications*, 2, 13-39.

Refenes, A. N., Bentz, Y., Bunn, D. W., Burgess, A. N., and Zapranis, A. D., 1994, Backpropagation with discounted least squares and its application to financial time-series modelling, *Proc. Neural Networks in the Capital Markets*, Pasadena, November 16-18, 1994.

Refenes, A. N., Bentz, Y., Bunn, D. W., Burgess, A. N., and Zapranis, A. D., 1997, Backpropagation with discounted least squares and its application to financial time-series modelling, *Neurocomputing*, Vol. 14, No. 2, 123-138.

Refenes, A. N., Bentz, Y. and Burgess, N., 1994, Neural networks in Investment Management, *Journal of Communications and Finance*, 8, April 95-101

Refenes, A. N., Burgess A. N., Bentz, Y., 1997, Neural Networks in Financial Engineering: a Study in Methodology, *IEEE Transaction on Neural Networks*, Vol. 8, No. 6, 1222-1267.

Refenes, A. N., Zapranis, A. D. and Francis, G., 1995, Modelling stock returns in the framework of APT: A comparative study with regression models, in (ed.) Refenes (1995), 101-125.

Refenes, A. N., and Zaidi, A., 1995, Managing Exchange-Rate Pprediction Strategies with Neural Networks, in (ed.) Refenes (1995), 213-219.

Ripley, B. D., 1996, *Pattern Recognition and Neural Networks*, Cambridge University Press, England.

Roll, R., 1982, The Turn of the Year Effect and the Return Premium of Small Firms, *Journal of Portfolio Management*.

Rosenblatt, F., 1962, *Principles of neurodynamics*,. New York: Spartan.

Ross, S. A., 1976, The Arbitrage Pricing Theory of Capital Asset Pricing, *Journal of Economic Theory*, **13**, pp. 341-360

Rumelhart, D., E., Hinton, G. E., and Williams R., J., 1986. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland and the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 1:Foundations, pp. 318-362. Cambridge MA: MIT Press.

Samuelson, P., 1965, Proof that Properly Anticipated Prices Fluctuate Randomly, *Industrial Management Review*, 6, 41-49.

Satchell S. and Timmerman, A., 1995, An Assessment of the Economic Value of Nonlinear Foreign Exchange Rate Forecasts, *Journal of Forecasting*, 14(6), 477-498.

Schreiner, P., 1998, *Statistical Arbitrage in Euromark Futures using Intraday Data*, unpublished M.Sc. Thesis, Department of Mathematics, King's College London.

Sharpe, W. F., 1964, Capital Asset Prices: A Theory of Market Equilibrium, *Journal of Finance*, 19, 425-442.

Sharpe, W. F., 1966, Mutual Fund Performance, *Journal of Business*, January 1966, Vol. 39, No. 1 (January), 119-138.

Slutsky, E., 1927 , The summation of random causes as the source of cyclic processes. *Econometrica*, 5, 105-146

Sortino, F. A. and Forey, H. J., 1996, On the use and misuse of downside risk, *Journal of Portfolio Management*, 22, 35-44.

Srinivas, N., and Deb, K., 1995, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, *Evolutionary Computation*, 2(3), 221-248.

Sullivan, R., Timmerman, A., and White, H., 1998, *Data-Snooping, Technical Trading Rule Performance, and the Bootstrap*, UCSD Working Paper, 97-31.

Summers, L. H., 1986, Does the Stock Market Rationally Reflect Fundamental Values?, *Journal of Finance*, 41, 591-600.

Sutton, R. S., 1988, Learning to predict by the method of temporal differences, *Machine Learning*, 3, 9-44.

Tesauro, G. J., 1992, Practical issues in temporal difference learning, *Machine Learning*, 8 (3/4), 257-277.

Theil, H. and Wage, S., 1964, Some observations on adaptive forecasting, *Management Science*, 10, 198-206.

Titterington, D. M., 1985, Common Structure of Smoothing Techniques in Statistics, *International Statistical Review*, **53**, 2, pp. 141-170

Tjangdjaja, J., Lajbcygier, P., and Burgess, N., Statistical Arbitrage Using Principal Component Analysis For Term Structure of Interest Rates, in Xu, L. *et al* (Eds.), *Intelligent Data Engineering and Learning*, Springer-Verlag, Singapore, 43-53.

Towers, N., and Connor, J., 1998, Fitting No Arbitrage Term Structure Models using a Regularisation Term, in Refenes, A-P *et al.* (Eds.), *Decision Technologies for Computational Finance*, Kluwer Academic Publishers, Dordrecht, 323-332.

Towers, N., 1998, *Statistical Fixed Income Arbitrage*, Deliverable Report D4.6, ESPRIT project "High performance Arbitrage detection and Trading" (HAT), Decision Technology Centre, London Business School.

Towers, N., 1999, *Joint optimisation of trading strategies and forecasting models*, Unpublished PhD Transfer document, Decision Technology Centre, London Business School.

Towers, N., 2000, Trading Predictability in Financial Markets: A Methodological Review, Working Paper, Decision Technology Centre, London Business School.

Towers, N., and Burgess, A. N., 1998, Optimisation of Trading Strategies using Parametrised Decision Rules, in Xu, L. *et al* (Eds.), *Intelligent Data Engineering and Learning*, Springer-Verlag, Singapore, 163-170.

Towers, N., and Burgess, A. N., 1999a, Implementing trading strategies for forecasting models, *Proceedings Computational Finance 1999,* MIT press.

Towers, N., and Burgess, A. N., 1999b, A framework for applying Reinforcement Learning to Investment Finance, Working Paper, Decision Technology Centre, London Business School.

Towers, N., and Burgess, A. N., 2000a, Learning Trading Strategies for Imperfect Markets, Working Paper, Decision Technology Centre, London Business School.

Towers, N., and Burgess, A. N., 2000b, Optimising Forecasting Models for Trading, Proceedings from the Joint Conference Computational Finance 2000/ Forecasting Financial Markets 2000, to be published by John Wiley within the 'Financial Economics and Quantitative Analysis' series.

Watkins, C., 1989, Learning from delayed rewards, PhD Thesis, Cambridge University.

Weigend A. S., Huberman B. A. and Rumelhart, D. E., 1992, Predicting sunspots and exchange rates with connectionist networks, in *Nonlinear modelling and forecasting*, Eds. Casdagli M. and Eubank S., Addison-Wesley

Weigend, A. S., and Shi, S., 1998, *Predicting Daily Probability Distributions of S&P500 Returns*, Working Paper IS-98-23/S-98-36, Leonard N. Stern School of Business.

Weisweiller, R., 1986, *Arbitrage*, Wiley, New York, USA.

Werbos, P. J., 1974, Beyond regression: new tools for prediction and analysis in the behavioural sciences. Ph.D. Thesis, Harvard University, Boston, MA.

Werbos P.J., 1990, Backpropagation through time: What it does and how to do it, *Proc IEEE, Vol 78, No. 10*, Oct 1990.

White, H., 1988, Economic prediction using neural networks: The case of IBM daily stock returns, *Proc. IEEE International Conference on Neural Networks, July 1988,* reprinted in (Trippi and Turban, 1993)

White, H., 1997, *A Reality Check for Data Snooping*, San diego, NRDA Technical Report 97-01.

Winkler, R. C., and Makridakis, S., 1983, The combination of forecasts, *Journal of the Royal Statistical Society A,* 146, 150-57

Winters, P. R., 1960, Forecasting sales by exponentially weighted moving averages, *Management Science*, 6, 324-342

Wong, M. A., 1993, *Fixed-Income Arbitrage*, John Wiley and Sons, Inc: New York.