



## **Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin**

### **Copyright statement**

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

### **Liability statement**

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

### **Access Agreement**

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

# Improved Optimisation of Density Forecasting Models using Multi-Objective Search with Applications in Risk Management

Michael Carney

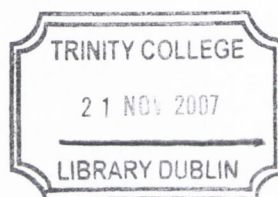
A thesis submitted to the University of Dublin, Trinity College

in fulfillment of the requirements for the degree of

Doctor of Philosophy

September 2006





THESIS  
8194

Improved Optimisation of Density  
Forecasting Models using  
Multi-Objective Search with Applications  
in Risk Management

Approved by  
Dissertation Committee:

---

---

---

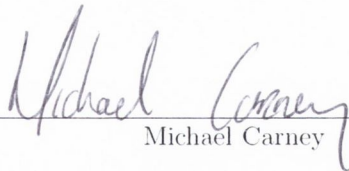
---

---

## Declaration

I, the undersigned, declare that this work has not previously been submitted to this or any other University, and that unless otherwise stated, it is entirely my own work. This thesis may be borrowed or copied upon request with the permission of the Librarian, University of Dublin, Trinity College.

The copyright belongs jointly to the University of Dublin, Trinity College and Michael Carney



Michael Carney

Dated: April 22, 2007

# Acknowledgements

It is a pleasure to thank the many people who helped and supported me throughout this thesis.

First, I would like to thank my supervisor Prof. Pádraig Cunningham. Throughout my thesis he provided encouragement, sound advice, good company, and lots of good ideas. I would have been lost without him. Secondly, I would like to thank my examiners Prof. Fionn Murtagh and Dr. Andrew Butterfield for reading my thesis so thoroughly and providing interesting and useful points of view on my work. Thirdly, I would like to thank the other members of the Machine Learning Group who were always willing to listen and discuss technical problems at the price of a pint. Especially, Brian MacNamee, Bridget Kane, Conor Nugent, Derek Greene, Donal Doyle, Ken Bryen, John Loughrey, Lorcan Coyle, Mike Davy, Neil MacDonnell and Peter Byrne.

I would especially like to thank my parents. For the past 24 years throughout my schooling and university my parents have provided unbelievable encouragement and support. I owe you big time! I'd like also to thank Lorna, John, Alyson, Gemma, Neil, Pablo and Oonagh, my brothers and sisters, for being great craic and providing loads of advice (and accommodation).

Finally, but definitely most importantly, I would like to thank my darling Mary for being so supportive and wonderful throughout my PhD. I don't know how you did it but you read the whole thesis and didn't complain once. You're great! Thanks also to Mary's family who have been fantastic throughout my PhD

Is that everyone?



**Michael Carney**

*University of Dublin, Trinity College*

*September 2006*

# Abstract

Density forecasting is becoming an increasingly popular method of prediction. Density forecasting models produce a probability density function estimate of a future event rather than a standard point estimate. Given that predictions are inherently uncertain, density forecasting is a logical means of presenting predictions and is the preferred forecasting method in most real-world domains where prediction models are widely used e.g. meteorology and finance. The additional information provided by a density forecast enables the prediction user to make optimal decisions under uncertainty.

The attributes of a good density forecasting model are accuracy at estimating the location of the target variable and empirical validity of probability estimates. The first attribute is popularly known as sharpness and is generally assessed by the amount of density a predicted distribution gives to the target variable. The latter attribute is typically referred to as calibration and refers to a density forecasting models capacity to comply with the intuitive assumption that a 90% prediction of an event should result in the event 9 out of 10 times. Calibration is a vital factor affecting the integrity of a predictive model. Alarming, it is often neglected, relegated to a minor role or entirely excluded from most of the density forecasting literature. However, it seems irrational for a prediction user to use a density forecasting model that shows poor calibration.

There are two main reasons for the neglect of calibration. First, assessing calibration in a meaningful way is not straightforward and secondly, integration of an assessment mechanism into an optimisation framework is difficult. Up until now, optimisation of density forecasting models has varied little from the standard maximum likelihood method used by point estimators. As shown in this thesis, the standard density forecasting optimisation approach is a generalisation of the sum of squares method and optimises sharpness only. Careful analysis of the effects of applying this approach to the resulting model shows that there are common problems that occur that result in poor and misleading model

predictions, particularly in terms of calibration.

In this thesis, the relationship between sharpness and calibration is analysed. This is made possible by the introduction of a new calibration metric that can successfully rank density forecasting models on this dimension. It is shown that sharpness and calibration are conflicting, meaning they do not converge to the same global optimum. To address this, a new formulation of how density forecasting models should be optimised is proposed that simultaneously optimises both sharpness and calibration. A general framework is constructed around this new concept that can be applied to most density forecasting techniques in use today. Rather than returning a single “optimum” model, the proposed framework uses a multi-objective evolutionary algorithm to return a set of optimal models that are trade-offs across the two objectives. This allows the model user to select the model that best suits their loss function.

To validate this new approach to density forecasting, two implementations of the framework are described in detail. The resulting density forecasting models based on the new framework are known as Pareto Mixture Density Networks and Pareto GARCH. A Mixture Density Network is a neural network density forecasting model and GARCH models are econometric density forecasting models that are very popular in finance. It is shown that the Pareto Mixture Density Networks successfully improves results over the standard Mixture Density Network model on a medical decision support application. Pareto GARCH results are compared against the state of the art and are shown to produce significantly better results on a risk management application.

# Contents

|                                                                  |      |
|------------------------------------------------------------------|------|
| Acknowledgements                                                 | iv   |
| Abstract                                                         | vi   |
| List of Figures                                                  | xii  |
| List of Tables                                                   | xv   |
| Associated Publications                                          | xvi  |
| Glossary                                                         | xvii |
| Chapter 1 Introduction                                           | 1    |
| 1.1 Motivation . . . . .                                         | 1    |
| 1.2 Thesis Statement . . . . .                                   | 3    |
| 1.3 Thesis Contributions . . . . .                               | 3    |
| 1.4 Thesis Overview . . . . .                                    | 4    |
| Chapter 2 Review of Density Forecasting Models                   | 7    |
| 2.1 Introduction . . . . .                                       | 7    |
| 2.2 Regression . . . . .                                         | 9    |
| 2.2.1 Point and Density Forecasting . . . . .                    | 11   |
| 2.3 Density Forecasting Models . . . . .                         | 13   |
| 2.3.1 Prediction Intervals . . . . .                             | 13   |
| 2.3.2 Neural Network Approaches to Density Forecasting . . . . . | 15   |
| 2.3.3 HMM Models . . . . .                                       | 26   |
| 2.3.4 Gaussian Processes . . . . .                               | 27   |
| 2.3.5 ARCH and GARCH models . . . . .                            | 27   |

|                                                                    |                                                         |           |
|--------------------------------------------------------------------|---------------------------------------------------------|-----------|
| 2.3.6                                                              | Meteorological Models . . . . .                         | 30        |
| 2.4                                                                | Discussion and Summary . . . . .                        | 31        |
| <b>Chapter 3 Density Forecast Evaluation</b>                       |                                                         | <b>33</b> |
| 3.1                                                                | Introduction . . . . .                                  | 33        |
| 3.2                                                                | Calibration and Sharpness . . . . .                     | 33        |
| 3.2.1                                                              | Negative Log Predictive Density . . . . .               | 35        |
| 3.2.2                                                              | Continuous Rank Probability Score . . . . .             | 37        |
| 3.2.3                                                              | Wilson Score . . . . .                                  | 38        |
| 3.2.4                                                              | Probability Integral Transform Histogram . . . . .      | 39        |
| 3.3                                                                | Discussion . . . . .                                    | 42        |
| <b>Chapter 4 Review of Multi-Objective Evolutionary Algorithms</b> |                                                         | <b>43</b> |
| 4.1                                                                | Introduction . . . . .                                  | 43        |
| 4.2                                                                | Optimisation Methods . . . . .                          | 43        |
| 4.2.1                                                              | Evolutionary Algorithms . . . . .                       | 46        |
| 4.2.2                                                              | Genetic Algorithms . . . . .                            | 48        |
| 4.2.3                                                              | Evolutionary Strategies . . . . .                       | 50        |
| 4.2.4                                                              | Other Evolutionary Algorithms . . . . .                 | 52        |
| 4.3                                                                | Multi-objective Evolutionary Algorithms . . . . .       | 53        |
| 4.3.1                                                              | A priori method . . . . .                               | 54        |
| 4.3.2                                                              | A posteriori method . . . . .                           | 56        |
| 4.4                                                                | Evolutionary Neural Networks . . . . .                  | 59        |
| 4.5                                                                | Evolutionary GARCH . . . . .                            | 60        |
| 4.6                                                                | Discussion and Summary . . . . .                        | 60        |
| <b>Chapter 5 Theory and Contributions</b>                          |                                                         | <b>63</b> |
| 5.1                                                                | Introduction . . . . .                                  | 63        |
| 5.2                                                                | Common Density Forecast Optimisation Problems . . . . . | 64        |
| 5.2.1                                                              | Error Function Sharpness . . . . .                      | 64        |
| 5.2.2                                                              | Over-fitting . . . . .                                  | 68        |
| 5.2.3                                                              | NLPD Sensitivity to Outliers . . . . .                  | 68        |
| 5.2.4                                                              | Discussion . . . . .                                    | 71        |
| 5.3                                                                | A New Calibration Score . . . . .                       | 72        |



|                  |                                                                                                                  |            |
|------------------|------------------------------------------------------------------------------------------------------------------|------------|
| 5.3.1            | Empirical Comparison: $A^2$ and Statistical Consistency . . . . .                                                | 74         |
| 5.3.2            | Test of $A^2$ Robustness . . . . .                                                                               | 75         |
| 5.4              | Conflicting Objectives . . . . .                                                                                 | 76         |
| 5.5              | A Multi-objective Methodology for Density Forecast Optimisation . . . . .                                        | 78         |
| 5.6              | Conclusions . . . . .                                                                                            | 79         |
| <b>Chapter 6</b> | <b>Implementation</b>                                                                                            | <b>81</b>  |
| 6.1              | Introduction . . . . .                                                                                           | 81         |
| 6.2              | A Multi-objective Evolutionary Algorithm for Optimisation of Density Fore-<br>casting Models . . . . .           | 82         |
| 6.2.1            | Implementation: Pareto MDN . . . . .                                                                             | 83         |
| 6.2.2            | Selection . . . . .                                                                                              | 84         |
| 6.2.3            | Adaptation . . . . .                                                                                             | 84         |
| 6.2.4            | Implementation: Pareto GARCH . . . . .                                                                           | 87         |
| 6.3              | A Multi-objective Evolutionary Algorithm for Generalised Optimisation of<br>Density Forecasting Models . . . . . | 88         |
| 6.4              | Analysis of the Pareto MDN on Artificial Data . . . . .                                                          | 90         |
| 6.4.1            | Experiment Details . . . . .                                                                                     | 91         |
| 6.4.2            | Experiment Results . . . . .                                                                                     | 91         |
| 6.4.3            | Analysis of the Calibration Score . . . . .                                                                      | 93         |
| 6.5              | Sample Pareto GARCH Result . . . . .                                                                             | 95         |
| 6.6              | An Evolutionary Algorithm for Optimising Density Forecasting Models . . .                                        | 97         |
| 6.6.1            | Implementation: EMDN . . . . .                                                                                   | 97         |
| 6.7              | Reducing Optimisation Time with the Cauchy Distribution . . . . .                                                | 98         |
| 6.8              | Conclusions . . . . .                                                                                            | 100        |
| <b>Chapter 7</b> | <b>Finance: Value-at-Risk</b>                                                                                    | <b>101</b> |
| 7.1              | Introduction . . . . .                                                                                           | 101        |
| 7.2              | Value-at-Risk . . . . .                                                                                          | 103        |
| 7.3              | Setup of Experiments . . . . .                                                                                   | 105        |
| 7.4              | Experiment 1: Standard GARCH vs Pareto GARCH . . . . .                                                           | 105        |
| 7.4.1            | Model Evaluations . . . . .                                                                                      | 109        |
| 7.4.2            | Value-at-Risk Evaluation . . . . .                                                                               | 112        |
| 7.4.3            | Discussion . . . . .                                                                                             | 113        |



7.5 Experiment 2: Pareto GARCH vs Pareto MDN . . . . . 115

7.5.1 Value-at-Risk Evaluation . . . . . 119

7.5.2 Discussion . . . . . 119

7.6 Experiment 3: Sliding Window Analysis . . . . . 120

7.6.1 Value-at-Risk Evaluation . . . . . 120

7.6.2 Discussion . . . . . 122

**Chapter 8 Medical Decision Support: Anticoagulant Drug Therapy 123**

8.1 Introduction . . . . . 123

8.2 Anticoagulant Drug Therapy . . . . . 124

8.3 Experiment . . . . . 125

8.4 Predicting Density Functions for Anticoagulant Drug Therapy . . . . . 127

8.5 Evaluation . . . . . 129

8.5.1 Comparative Evaluation . . . . . 130

**Chapter 9 Conclusions 133**

9.1 Introduction . . . . . 133

9.2 Thesis Contributions . . . . . 134

9.3 Future Work . . . . . 135

9.3.1 Dominance Uncertainty . . . . . 135

9.3.2 Model Uncertainty . . . . . 135

9.3.3 Different Objective Functions . . . . . 136

9.4 Concluding Remarks . . . . . 136

# List of Figures

|     |                                                                      |    |
|-----|----------------------------------------------------------------------|----|
| 2.1 | Point versus Density Forecasting . . . . .                           | 10 |
| 2.2 | Neural Network Architecture Components . . . . .                     | 15 |
| 2.3 | Basic Multi Layer Perceptron Architecture . . . . .                  | 16 |
| 2.4 | Predicting Local Error Bars . . . . .                                | 19 |
| 2.5 | Pearson Crab Mixture Model . . . . .                                 | 20 |
| 2.6 | Basic Mixture Density Network Architecture . . . . .                 | 21 |
| 2.7 | Recurrent Mixture Density Network Architecture . . . . .             | 23 |
| 2.8 | Gaussian Mixture Random Vector Functional Link . . . . .             | 26 |
| 2.9 | Gaussian Process Example . . . . .                                   | 28 |
| 3.1 | NLPD Quality Score Example . . . . .                                 | 36 |
| 3.2 | CRPS Quality Score Example . . . . .                                 | 38 |
| 3.3 | Wilson Score Example . . . . .                                       | 39 |
| 3.4 | Constructing the Probability Integral Transform Histogram . . . . .  | 40 |
| 3.5 | Interpreting Probability Integral Transform Histograms . . . . .     | 41 |
| 4.1 | Flow diagram of standard EA. . . . .                                 | 47 |
| 4.2 | Sample Parameter Space to Objective Function Space Mapping . . . . . | 55 |
| 4.3 | Flow diagram of standard MOEA. . . . .                               | 57 |
| 5.1 | Logistic Function . . . . .                                          | 65 |
| 5.2 | Comparison of Error Scoring Methods . . . . .                        | 67 |
| 5.3 | Sample PIT histogram from Over-fitted MDN . . . . .                  | 69 |
| 5.4 | Sine Wave Data . . . . .                                             | 70 |
| 5.5 | Sample PIT histogram from under confident MDN . . . . .              | 71 |
| 5.6 | Quantile Ranking Comparison . . . . .                                | 75 |

|      |                                                                          |     |
|------|--------------------------------------------------------------------------|-----|
| 5.7  | Demonstration of the Robustness of $A^2$ Score . . . . .                 | 76  |
| 5.8  | Comparison of $NLPD$ and $A^2$ error surfaces . . . . .                  | 78  |
| 5.9  | Theoretical Depiction of a Density Forecast Model Pareto Front . . . . . | 79  |
| 6.1  | MDN Node and Layer Reference . . . . .                                   | 83  |
| 6.2  | Demonstration of MDN weight deletion . . . . .                           | 85  |
| 6.3  | Demonstration of MDN node deletion . . . . .                             | 86  |
| 6.4  | Sample plot of artificial data set. . . . .                              | 90  |
| 6.5  | Pareto MDN Objective Space . . . . .                                     | 92  |
| 6.6  | Comparison of Density Forecast Prediction and True Conditional Density . | 93  |
| 6.7  | Kullback-Leibler versus $A^2$ Score Example . . . . .                    | 95  |
| 6.8  | Sample Pareto GARCH(1,1) output. . . . .                                 | 96  |
| 6.9  | Comparison of Cauchy and Gaussian Distributions . . . . .                | 98  |
| 6.10 | Convergence with different mutation methods . . . . .                    | 99  |
| 7.1  | Sample Price and Return Series . . . . .                                 | 102 |
| 7.2  | Sample 5% 1-day Value-at-Risk . . . . .                                  | 104 |
| 7.3  | IBM Log Returns Data . . . . .                                           | 106 |
| 7.4  | IBM In-Sample Objective Space . . . . .                                  | 108 |
| 7.5  | IBM Out-of-sample Objective Space . . . . .                              | 109 |
| 7.6  | IBM Out-of-sample PIT histograms . . . . .                               | 110 |
| 7.7  | IBM Autocorrelation Function Plot $z - \bar{z}$ . . . . .                | 111 |
| 7.8  | IBM Autocorrelation Function Plot $(z - \bar{z})^2$ . . . . .            | 112 |
| 7.9  | IBM Autocorrelation Function Plot $(z - \bar{z})^3$ . . . . .            | 113 |
| 7.10 | IBM Autocorrelation Function Plot $(z - \bar{z})^4$ . . . . .            | 114 |
| 7.11 | DEM/GBP Exchange Rate Data Set . . . . .                                 | 115 |
| 7.12 | DEM/GBP GARCH Pareto Front In-Sample and Out-of-Sample . . . . .         | 116 |
| 7.13 | DEM/GBP MDN Pareto Front In-Sample and Out-of-Sample . . . . .           | 117 |
| 7.14 | DEM/GBP GARCH PIT Histograms In-Sample and Out-of-Sample . . . .         | 118 |
| 7.15 | DEM/GBP MDN PIT Histograms In-Sample and Out-of-Sample . . . . .         | 118 |
| 7.16 | USDGBP In and Out-of-sample Objective Space . . . . .                    | 120 |
| 7.17 | Box plot of percentile estimate accuracy . . . . .                       | 121 |
| 8.1  | Unconditional distribution of patient INR values . . . . .               | 126 |

8.2 Scenario test INR probability density predictions . . . . . 128

# List of Tables

- 4.1 Example of cross-over . . . . . 48
- 6.1 Nomenclature for optimisation algorithms . . . . . 81
- 6.2 Pareto MDN Parameter Settings . . . . . 91
- 6.3 Model Results Comparison on Synthetic Data . . . . . 93
- 7.1 The *NLPD*,  $A^2$  and *CRPS* scores achieved by the 6 models on the in and out-of sample data. . . . . 109
- 7.2 Quantile Exceedance for Assorted Model Types . . . . . 112
- 7.3 Spearman Rank Correlation Results for Pareto GARCH Objectives In and Out-of-Sample . . . . . 117
- 7.4 DEM/GBP percentage of times quantile exceeded . . . . . 119
- 7.5 Sliding Window Analysis Results over 10 Runs . . . . . 122
- 8.1 The fifteen attributes compiled for each case in our input set. . . . . 125
- 8.2 Probabilistic estimate of patient condition . . . . . 127
- 8.3 Out-of-sample performance on Warfarin data . . . . . 129
- 8.4 Classifier comparison on Warfarin data: Specificity and Sensitivity . . . . . 131



# Associated Publications

Carney, M., Cunningham, P., Dowling, J., & Lee, C. (2005). Predicting probability distributions for surf height using an ensemble of mixture density networks. *In Proceedings of the 22nd International Conference in Machine Learning* (pp. 113-121)

Carney, M., & Cunningham, P. (2006). Calibrating Probability Density Forecasts with Multi-objective Search. *To Appear: In Proceedings of the 17th European Conference on Artificial Intelligence, Riva del Garda, Italy*

Carney, M., & Cunningham, P. (2006). Making Good Probability Estimates for Regression. *To Appear: In Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany*

Carney, M., & Cunningham, P. (2006). Evaluating Density Forecasting Models. *To Appear: In Proceedings of the 17th Irish Conference on Artificial Intelligence and Cognitive Science, Belfast*

Carney, M., Cunningham, P. & Lucey, B. M. (2006). Making Density Forecasting Models Statistically Consistent. *Submitted to: Journal of Quantitative Finance.*

Carney, M., & Cunningham, P. (2006). The Benefits of Using a Complete Probability Distribution when Decision Making: An Example in Anticoagulant Drug Therapy. *Forthcoming: Journal of Medical Decision Making.*

# Glossary

|                |                                                        |
|----------------|--------------------------------------------------------|
| A <sup>2</sup> | Anderson Darling Score                                 |
| ARCH           | Autoregressive Conditional Heteroskedastic             |
| CRPS           | Continuous Rank Probability Score                      |
| EA             | Evolutionary Algorithms                                |
| EANN           | Evolutionary Artificial Neural Network                 |
| EMDN           | Evolutionary Mixture Density Network                   |
| EP             | Evolutionary Programming                               |
| ES             | Evolutionary Strategy                                  |
| GA             | Genetic Algorithm                                      |
| GARCH          | Generalized Autoregressive Conditional Heteroskedastic |
| GM-RVFL        | Gaussian Mixture Random Vector Functional Link         |
| MDN            | Mixture Density Network                                |
| MLP            | Multi-Layer Perceptron                                 |
| MOEA           | Multi-objective Evolutionary Algorithms                |
| NLL            | Negative Log Likelihood                                |
| NLPD           | Negative Log Predictive Density                        |
| PIT            | Probability Integral Transform                         |

# Chapter 1

## Introduction

*“Reports that say that something hasn’t happened are always interesting to me, because as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns - the ones we don’t know we don’t know.”*

Donald Rumsfeld, US Secretary of Defence.

### 1.1 Motivation

Regression is a supervised learning problem where the fundamental task is to predict some continuous variable. There are innumerable examples of this type of prediction problem from the medical domain to financial and environmental modelling. The typical way regression models express an estimate of a continuous random variable is to present the prediction in terms of a point estimate. For example, a point prediction model trained to forecast the price of Microsoft shares might produce a point estimate of \$27.90 for tomorrow’s price. Presenting predictions for regression problems in this way has many advantages that make the problem tractable and the predictions convenient to evaluate and interpret. However, one should remember that although predictions attempt to reduce uncertainty they do not eliminate it. Therefore, predictions should be probabilistic by nature, taking the form of probability distributions over future events (Dawid, 1984).

Furthermore, there are many prediction problems where simple ‘point’ estimates are inadequate to guide action. For example, in meteorology and in finance it is often important to know the probability of a variable falling within (or outside) different ranges. It might be that a 20% probability of flooding may be enough to place emergency services



on alert, or the best returns in financial trading might come from events that have a small but significant probability. These types of applications have encouraged a growing interest in probability density forecasting.

Probability density forecasting, or, more commonly, density forecasting, is the area of regression that attempts to produce estimates of the complete conditional probability density function as opposed to just providing a point estimate of the conditional mean. It is one of the most popular prediction methodologies for real world problems because it enables the prediction user to make an optimal decision regardless of their personal loss function (Diebold et al., 1998b). In applications where taking large risks is unacceptable, such as in medical decision making and financial trading, density forecasting provides a means of quantifying risks so that measures can be taken to reduce the possibility of extreme, adverse results.

The advantages and applications of density forecasting models outlined above are well established, for example (Cooke, 1906; Brier, 1950). However, there still exist a number of challenges in the density forecasting domain. For example, evaluation of density forecasting models is difficult because the model output is very different to the actual realisation. Many density forecasting practitioners propose their own evaluation metrics that are specific to their problem domain. However, Dawid (1984) and Gneiting et al. (2003) have suggested a universal method of evaluation. They argue that the objectives of a density forecast are to be as *sharp* as possible and *calibrated* in the long run. Sharpness refers to the concentration of the predicted distributions and calibration refers to the statistical consistency between the distributional forecasts and the target variables. There is a strong relationship between sharpness and calibration in density forecasting and bias and variance in point forecasting (Geman et al., 1992; Friedman, 1997). In fact, sharpness and variance are equivalent in certain circumstances for example, when the predicted density is uni-modal. However, sharpness and calibration are general evaluation metrics for any density forecasting models where as bias and variance can only be used to assess the quality of density forecasts in very specific scenarios.

A second challenge is that the most common method of optimisation of density forecasting models, the *negative log-likelihood*, is known to have problems in certain scenarios. This method is based on a generalisation of the popular maximum likelihood approach to point estimation. Density forecasts are penalised based on the predicted density at the actual target. Problems with this approach have been highlighted in numerous papers.

For example, Weigend and Shi (2000) and Candela et al. (2006) identify that the negative log-likelihood is sensitive to extreme outliers, returning sub optimal results when these are present in the data. The negative log-likelihood is also prone to more common model optimisation problems such as over-fitting and model bias. All of these deficiencies can be diagnosed if the model is properly evaluated taking into account both sharpness and calibration. However, this form of evaluation is often not performed and the density forecasting model is used without a complete understanding of the prediction biases. This can result in poor location estimates and/or inaccurate probability estimates. Misleading probabilistic estimates can lead to worse decisions being made by the prediction user than if they were presented with a good point estimate. Therefore, the calibration accuracy of density forecasting models should be a high priority for density forecasters.

## 1.2 Thesis Statement

This thesis claims that by optimising density forecasting models in terms of sharpness and calibration simultaneously, common deficiencies such as over-fitting and sensitivity to outliers can be overcome and better results can be achieved. To accomplish this, a general multi-objective framework for optimisation of density forecasts is proposed. The framework combines the standard negative log-likelihood objective function with a new calibration score to optimise two popular density forecasting modelling techniques using a multi-criteria stochastic search algorithm. The output of the new framework is a set of optimal models with different trade-offs between negative log-likelihood and calibration known as a Pareto front. Synthetic data analysis shows that this new optimisation approach can identify and correct common optimisation defects to return better models to the prediction user without augmenting the model with additional parameters.

Evaluation of the framework is carried out on both finance and medical decision making problems. The new framework results are compared with the state-of-the-art optimisation approaches and shown to improve performance significantly in these problem domains.

## 1.3 Thesis Contributions

In this section the main contributions of the thesis are summarised:

- *A new calibration quality score for density forecasts* is proposed (the  $A^2$  score) based



on the Anderson-Darling goodness-of-fit test for uniformity. It is shown that this scoring metric is robust, simple to calculate and effective even when sample sizes are small.

- *Calibration and sharpness*, the objectives of a good probability density forecasting model, *are shown to be conflicting* for the first time. This means that sharpness and calibration do not converge to the same global minimum.
- *A new methodology for density forecasting model optimisation* is proposed using a multi-objective evolutionary algorithm and a sharpness and calibration score. This technique optimises directly on calibration while only allowing minimum deterioration on sharpness. The optimisation algorithm outputs a set of optimal models with varying degrees of trade-off between sharpness and calibration so the prediction users can select the model that best suits their goals. Excellent improvement in results is demonstrated and how they can be applied to improve risk management is discussed. This is achieved without effecting the modeller's ability to interpret the model or changing the number of parameters in the model.
- Two implementations of the proposed algorithm *Pareto MDN and Pareto GARCH are described*. Various design decisions are discussed and a final implementation is described in detail. A number of experiments are performed to test the proposed approach. The results of these experiments validate the method and can also be used to *draw general inferences* such as the suitability of machine learning techniques for density forecasting model optimisation in financial and medical applications.

## 1.4 Thesis Overview

**Chapter 2** In this chapter regression and density forecasting are formally introduced. A comparison of density and traditional point forecasting motivates the use of probabilistic predictions. The relationship between point forecasting and density forecasting within the maximum likelihood framework is also shown. A review of density forecasting models is provided with descriptions of the most popular methods. Two models are selected, Mixture Density Networks (Bishop, 1994) and Generalised Autoregressive Conditional Heteroskedastic (Bollerslev, 1986) models, to be used as sample density forecasting models in subsequent experiments and evaluations.

**Chapter 3** explores the goals of density forecasting. Two objectives are determined to be desirable, calibration, the empirical validity of probabilistic forecasts and sharpness, the concentration of predicted densities. Different evaluation metrics from the density forecast evaluation literature are described and three are suggested for use when evaluating model performance. The Continuous Rank Probability Score (Hersbach, 2002), the Negative Log Predictive Density (Good, 1952) and the Probability Integral Transform Histogram (Dawid, 1984).

**Chapter 4** discusses various optimisation methods with particular attention paid to evolutionary approaches. This chapter reviews the state-of-the-art in evolutionary optimisation for both uni and multi-objective problems. Neural network and GARCH models optimised using evolutionary algorithms are also discussed. This review provides preparatory information for the solution discussed in subsequent chapters.

**Chapter 5** provides examples of the most common difficulties that occur when optimising density forecasting models. These include, over-fitting of the conditional variance, sensitivity to outliers and bias toward sharpness. These problems represent some of the most significant challenges in density forecasting. The remainder of this chapter describes the solution proposed in this thesis. First, a new objective function for optimisation of calibration, the  $A^2$  score, is described. Second, it is shown that calibration and sharpness are conflicting. Finally, a new multi-objective optimisation methodology for density forecasting is described.

**Chapter 6** provides a detailed description of the proposed multi-objective evolutionary approach in algorithmic form. Implementation details for a Mixture Density Network and a Generalised Autoregressive Conditional Heteroskedastic model are described. The resulting models are called Pareto MDN and Pareto GARCH models. Synthetic data experiments provide an initial indication that the proposed algorithms can successfully improve density forecast quality.

**Chapter 7** evaluates the proposed optimisation method on a risk management application. Three experiments are performed. First, comparison with the standard optimisation algorithm, a new Continuous Rank Probability Score optimised GARCH model and the Pareto GARCH model is performed. Second, the Pareto GARCH and Pareto MDN are compared on another financial data set. Finally, a third experiment shows how the proposed approach can consistently produce better results than the current state-of-the-art approach.

**Chapter 8** presents density forecasting as a solution to a medical decision problem. In the medical domain, prediction confidence is very important and regression model solutions are typically discretised to reduce the number of prediction failures. Probabilistic forecasts offer an alternative to this and a demonstration of how they can be used to improve performance is provided in this chapter. An anticoagulant drug therapy problem is used as the application domain and the Pareto MDN model is compared with a number of alternative classification approaches. A calibrated Pareto MDN model is shown to considerably outperform alternative prediction models in this application.

**Chapter 9** summarises the thesis and provides a synopsis of the contributions of this thesis. Finally, future research directions are discussed.



## Chapter 2

# Review of Density Forecasting Models

*“[Those who] limited their inquiries to Averages, and do not seem to revel in more comprehensive views. Their souls seem as dull to the charm of variety as that of a native of one of our flat English counties, whose retrospect of Switzerland was that, if the mountains could be thrown into its lakes, two nuisances would be got rid of at once.”*

Francis Galton, on the charms of statistics, *Natural Inheritance* p. 62.

### 2.1 Introduction

Density forecasts in themselves are not new. The meteorological literature has been encouraging their use since the start of the last century. For example, Cooke (1906) argued that each meteorological prediction should be accompanied by a single number that “will indicate, approximately, the weight or degree of probability which the forecaster himself attaches to that particular prediction”. Even when studies have focused on point expectations, rather than complete probability distributions, densities have frequently existed implicitly; for example, given the conditional mean forecast, a basic estimate of variance by way of the mean squared error and an accompanying assumption of normality provides a complete specification of the forecast model’s density function.

Despite the inherent logic behind using a probability density forecasting model over a point estimation model there has been a notable lack of interest in this aspect of regression within computational statistics until recently. Diebold et al. (1998b) identify three possible



reasons why this may have occurred:

1. *Restrictive assumptions* - Until recently, due to the computational complexity of making density forecasts, very restrictive assumptions had to be made about the data. For example, the number of parameters that could be estimated and the distributions that could be assumed were very limited.
2. *Lack of demand* - In the past there was seemingly less demand for density forecasts. In the two primary application areas of density forecasting, finance and meteorology, density forecasting models tend to come from the statistics field with models from machine learning playing a very minor role.
3. *Difficulty of evaluating the problem* - Daily, we use and accept probabilistic predictions for common prediction tasks; the weatherman tells you there is a 70% chance of rain, or a business journalist tells you there is a 40% chance of an interest rate hike in November, for example. But if it does rain, and if interest rates do increase, were those predictions right or wrong?

Fortunately, the difficulties outlined by Diebold et al. (1998b) are being addressed. Models have advanced sufficiently in the last decade so that less restrictive assumptions have to be made by the modeller. Non-linear functions with complex conditional density dynamics can now be accurately modelled by density forecasting models. This new flexibility and accuracy has stimulated demand for density forecasts and increased the possible applications that can be addressed. For example, the recent growth in the area of risk management has widened the focus of researchers in density forecasting, looking for new and better methods for quantifying risks. Diebold et al. (1998b) go so far as to say that the ‘*booming area of risk management is essentially dedicated to density forecasting*’. Furthermore, Timmermann (2000) comments that, ‘density forecasting is fast becoming an important tool for decision makers in situations where loss functions are asymmetric and forecast errors follow non-Gaussian distributions’. Now that a critical mass of researchers are focusing on the problem of density forecasting new application specific and generic evaluation techniques are emerging.

This chapter reviews the state-of-the-art of density forecast modelling techniques. The relationship between point forecasting and density forecasting is explained within a Maximum Likelihood framework and two popular density forecasting models are selected for

experimentation in subsequent chapters. First, however, a formal definition of the problem being addressed in this thesis is provided.

## 2.2 Regression

All the prediction problems in this thesis share a common set of concepts and terminology. Furthermore, since the literature does not use consistent notation, it is helpful to start by describing the problem in terms of the notation that will be used throughout the rest of the thesis.

In general, a forecast is made by presenting a forecasting model with a vector of input parameters. The forecasting model produces an estimate of the corresponding target/response value associated with that input vector. Formally, the regression problems addressed in this thesis can be stated as estimating the parameters,  $\mathbf{w}$ , for a model,  $f$ , given a set of  $M$  training examples,  $X = \{(\mathbf{x}_i, t_i)\}_{i=1}^M$ , where the  $i^{\text{th}}$  example is described by the input vector  $\mathbf{x}_i \in \mathbb{R}^n$  and the associated response or target variable  $t_i \in \mathbb{R}$ . It is assumed that there is only one dependent variable. This is to maintain notational simplicity and clarity, and because all examples in the evaluation sections have only one dependent variable. The adaptation of most equations containing the dependent variable to a multiple dependent variable version is trivial. At points in this document sequenced or time series data are referred to and in these cases the reader can assume that  $x_i = t_{i-1}$ .

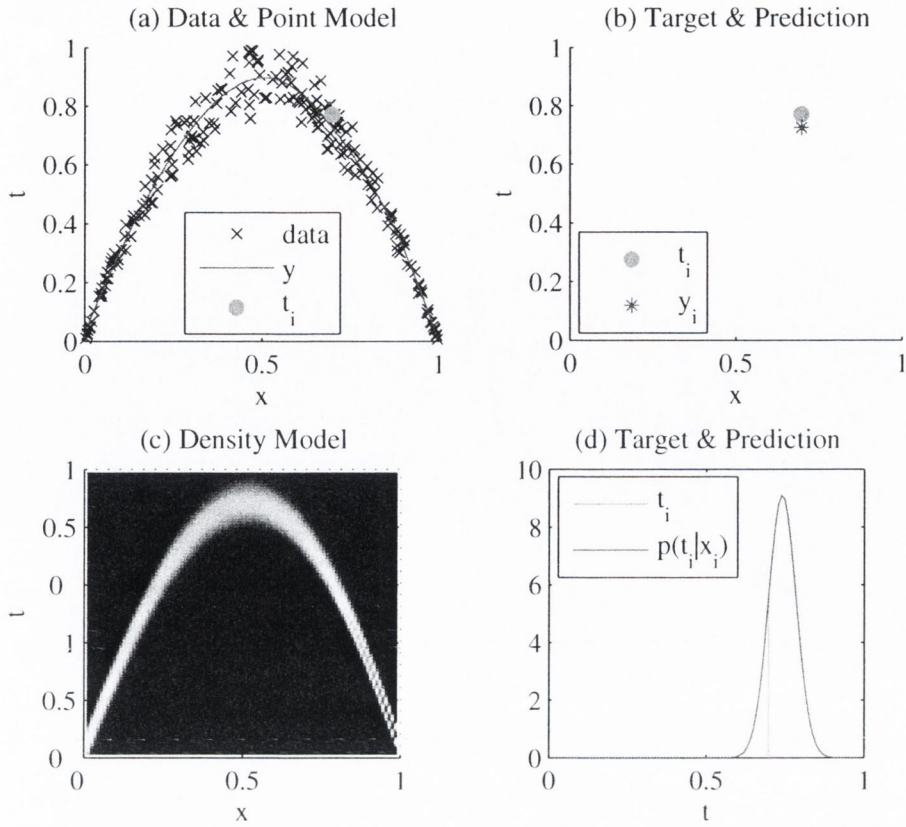
With this setup, a point forecasting model will produce an estimate  $y_i$  for the target variable  $t_i$  given the input vector  $\mathbf{x}_i$ , where the scalar  $y_i$  is,

$$y_i = f(\mathbf{w}; \mathbf{x}_i) \tag{2.1}$$

Similarly, a density forecasting model will produce an estimate  $p(t_i|\mathbf{x}_i)$  for the target variable  $t_i$  given the input vector  $\mathbf{x}_i$  where the probability density function  $p(t_i|\mathbf{x}_i)$  is,

$$p(t_i|\mathbf{x}_i) = f(\mathbf{w}; \mathbf{x}_i) \tag{2.2}$$

In the density forecasting model case the assumption is that there is some true data generating distribution at each instance and the resulting target variable  $t_i$  is a random realisation of that generating distribution. This distribution is known as the true conditional density function, or the probability of the target given the input vector. When a sum-of-squares error function is used during optimisation of a point estimator, the pre-



**Figure 2.1:** Shown are the models produced using both a point and density forecasting technique. Plot (a) shows the underlying data set and the conditional mean model produced using a sum-of-squares regression model. Also identified is a single data point that will be used for contrasting individual prediction outputs. Plot (b) shows the output from the point forecasting model for the individual data point. Plot (c) shows the model produced by a density forecasting model for the data. A contour map is shown because the resulting model has a 3 dimensional shape,  $x$ ,  $t$ , and density. Plot (d) shows a prediction from the density model for the individual data point.

diction  $y_i$  from the model is an estimate of the conditional mean of the data generating distribution,  $\langle t_i | \mathbf{x}_i \rangle$

Explaining the intuition behind a density forecast is aided by a graphical example. Figure 2.1 demonstrates a possible density forecast problem using a simple synthetic data example. The data is taken from the logistic function data set (see Section 5.2). This data has heteroskedastic variance. A comparison of the functions produced by a point estimator and a density estimator are shown in plots (a) and (c). Plots (b) and (d) compare model outputs for a single prediction. Plot (d) shows how a density forecast presents a complete probability distribution rather than a single point expectation. This represents an explicit probability for every conceivable outcome of  $t_i$  rather than the mean outcome.



This demonstration shows how a density forecasting model produces a comprehensive specification of the uncertainty surrounding a prediction beyond what a typical point forecast produces.

The specific detail of how a density forecast is produced varies with the particular application; for example, some macro-economic forecasts of inflation or economic growth are based on survey responses. Others use a combination of mathematical models and expert opinions. However, the most common approach is to use a mathematical model to estimate a density forecast with the set up described above. In the next section, the most popular theoretical approach to optimisation of point forecasting is introduced and its connection with density forecasting is shown. The resulting negative log-likelihood approach to density forecast optimisation is the most widely used optimisation method for regression problems.

### 2.2.1 Point and Density Forecasting

The objective in point forecasting regression is to find a curve of best fit for the data available. The point estimate approach to solving this problem, as least squares and maximum likelihood suggest, is to use a sum-of-squares approach. This curve can have many forms, however, it can be generalised to be some function  $f$  that takes a vector of parameters  $\mathbf{w}$ . With this set up, the fitting of a function to the available data amounts to minimizing,

$$E(\mathbf{w}|X) = \sum_{i=1}^M [t_i - y_i]^2 \quad (2.3)$$

where,  $y_i = f(\mathbf{x}_i; \mathbf{w})$ . The resulting function,  $f$ , can be used to estimate the target variable for previously unobserved examples.

Maximum Likelihood Theory constructs a principled argument behind the least squares approach to regression (Aldrich, 1997). The way in which the maximum likelihood framework derives the sum-of-squares error term (Equation 2.3) is as follows.  $p(t|x)$  is the probability of the output given the input, and  $p(x)$  is the input density. The log likelihood is

$$\begin{aligned} L(\mathbf{w}|X) &= \log \prod_{i=1}^M p(t_i, \mathbf{x}_i) \\ &= \log \prod_{i=1}^M p(t_i|\mathbf{x}_i) + \log \prod_{i=1}^M p(\mathbf{x}_i) \end{aligned}$$



This is true because it is known that  $p(t, x) = p(t|x)p(x)$ . We can ignore the second term,  $\log \prod_{i=1}^M p(\mathbf{x}_i)$ , since it does not depend on our estimator. When a likelihood function contains exponents, such as above, it is typical to convert the loss function from being a maximisation task to a minimisation task by negating the function. This is usually called the error function  $E$ . Then, by simple algebraic manipulation and the assumption that the predicted density is Gaussian we can say;

$$E(\mathbf{w}|X) = -\log \prod_{i=1}^M p(t_i|\mathbf{x}_i) \tag{2.4}$$

$$\begin{aligned} &= -\log \prod_{i=1}^M \frac{1}{\sqrt{2\pi}\sigma} \exp -\frac{(t_i - f(\mathbf{x}_i; \mathbf{w}))^2}{2\sigma^2} \\ &= -\log \left( \frac{1}{\sqrt{2\pi}\sigma} \right)^M \exp -\frac{1}{2\sigma^2} \sum_{t=1}^M (t_i - f(\mathbf{x}_i; \mathbf{w}))^2 \\ &= M \log(\sqrt{2\pi}\sigma) + \frac{1}{2\sigma^2} \sum_{t=1}^M (t_i - f(\mathbf{x}_i; \mathbf{w}))^2 \end{aligned} \tag{2.5}$$

If, in Equation 2.5 above, we drop the first part of the expression because it does not depend on  $\mathbf{w}$  and if we assume that  $\sigma$  is constant for all predictions and so is a constant also and can be removed, the result is Equation 2.3, the sum-of-squares error function.

This approach has a strong theoretical grounding when it can be reasonably assumed that the deviations of the targets from the expectations of the true generating function are independently, identically and normally distributed with standard deviation  $\sigma$ . By removing the need to estimate variance, the result is a very simple error function. However, removing the effect of variance is a profoundly restricting assumption in scenarios where the conditional variance of the data is known to be heteroskedastic or the generating distribution is non-Gaussian. Density forecasting on the other hand, does not restrict the conditional variance to being a constant or assume a specific distribution type.

To optimise a density forecasting model it is possible to simply adapt the maximum likelihood approach above. The error function for optimisation of density forecasting models is Equation 2.4 above. This variation or generalisation of the maximum likelihood is known as the *negative log-likelihood* and is the most common error function used in density forecasting. Its close relationship with point forecasting means that the various optimisation algorithms used for these methods can be easily adapted and applied to density forecasting models.

In summary, the derivation and comparison above shows how density forecasting relates to the most common form of regression optimisation, point forecasting, and shows how density forecasting models are a generalised form of point forecasting.

## 2.3 Density Forecasting Models

### 2.3.1 Prediction Intervals

Prediction Intervals<sup>1</sup> are a common and popular way of expressing the uncertainty in predictions. Although the terms prediction interval and density forecast are often interchanged, in this thesis they will be treated as distinctly different prediction types. In this document, prediction intervals refer to an indirect approach to uncertainty forecasting that determines an estimate of the conditional probability by way of estimating a specific interval to the data whereas density forecasting will refer to a direct estimation of the conditional density function. It is possible with prediction intervals to (post estimation of the interval) fit a distribution to the prediction based on the interval, however, this is more of an optional appendage than the primary focus of this technique.

For a problem that is defined as;

$$t = g(\mathbf{x}) + \epsilon \tag{2.6}$$

where,  $t$  is the observed target variable,  $g(\mathbf{x})$  is the true data generating process and  $\epsilon$  is noise with zero mean, the aim in regression is to train a model that predicts  $f(\mathbf{x}; \mathbf{w}) = g(\mathbf{x})$  or the conditional mean of the data. There are two components of uncertainty that can be modelled in this scenario. The first is concerned with the accuracy of the estimate of the true conditional mean. This is the distribution of the quantity  $g(\mathbf{x}) - f(\mathbf{x}; \mathbf{w})$ . These estimates are called confidence intervals (see (Heskes, 1997b)). The second form of uncertainty estimation is concerned with the confidence in the predictions of the actual targets i.e. the distribution of  $t - f(\mathbf{x}; \mathbf{w})$ . The interval estimates of the unknown future value of the target variable are called prediction intervals.

Prediction intervals<sup>2</sup> are of more practical use because they are concerned with the accuracy with which we can predict the observed values. The primary aim of these ap-

---

<sup>1</sup>Also known as forecast limits (Wei, 1990), prediction bounds (Brockwell & Davis, 1991), or forecast regions (Hyndman, 1995).

<sup>2</sup>To add to the confusion Granger and Newbold (1986) uses the term confidence interval to refer to prediction intervals.

proaches is to estimate prediction intervals with the correct coverage over the test sample i.e. a correct model that produces a 95% interval should cover 95% of the data points.

Typically, prediction intervals are found using a re-sampling technique such as the bootstrap method (Efron & Tibshirani, 1993). Chatfield (1993) has provided a comprehensive review of the various other approaches to prediction interval forecasting. Bootstrapping is a statistical method for estimating the sampling distribution of an estimator by sampling with replacement from the original sample. A prediction interval is estimated by creating  $N$  bootstrap sample sets from the original training data and training  $N$  models on those different sample sets. The spread, or the disagreement, in the estimates of the resulting models is used to estimate the prediction interval. Other re-sampling methods include splitting the past data into different disjoint sets to obtain an estimate of the interval, for example see (Williams & Goodman, 1971). There are a number of varying approaches, for example, see (Stine, 1985), (Thombs & Schucany, 1990), and (Carney et al., 1999). These re-sampling methods are particularly useful when dealing with problems that are very complex, non-linear, or where the data set is small (Veall, 1989).

Although prediction intervals are a useful means of acquiring insight into the possible range of outcomes there are a number of restrictions that should be highlighted. The distribution of errors may not be Gaussian; they may in fact be asymmetric or heavy-tailed. A prediction interval estimate is based on an assumption that there is zero bias in the model and so if the generating distribution is in fact asymmetric the result is a poor interval estimate. Alternatively, if the generating function is derived from a heavy-tailed distribution, as is common in financial data, the prediction intervals will perform poorly on estimates of rare events, resulting in the model being over-confident. Granger and Newbold (1986) suggest that the problem of heavy-tail distributions can be overcome by reducing the range of the interval i.e. they suggest a 50% interval, rather than the more popular 95% interval. Although this solves the problem, it is unlikely that this is a practical solution for forecast users as the resulting prediction interval only has a 50% chance of containing the observation! Estimation of prediction intervals is a useful means of ascertaining the uncertainty in a given prediction. It provides the forecast user with a meaningful range of possible outcomes. However, it can be seen as a stepping stone between point forecasting and the focus of this thesis - density forecasting.



### 2.3.2 Neural Network Approaches to Density Forecasting

#### Introduction to Neural Networks

The artificial neural network (NN) is a very powerful and flexible universal approximator. Many approaches to density forecasting are adaptations of NNs. In order to understand these methods it is necessary to first understand how neural networks work. Readers who are familiar with the way neural networks operate can skip to the next sub-section.

The ability of the human brain to perform complex tasks such as motor control or pattern recognition far faster and more accurately than we can mimic with current computational methods has been of great interest to researchers in Artificial Intelligence for decades. Neuroscience suggests that the human brain can achieve this by way of a complex, non-linear, adaptive and parallel architecture. The Artificial Neural Network (NN) attempts to emulate the way the brain functions within a mathematical framework. The result is a flexible statistical model that can model high-dimensional non-linear data. Unfortunately, like the human brain, NNs are “black boxes”. Although, NNs can produce predictions that are highly accurate, the manner in which the NN represents the relationship between the independent and dependent variables cannot be easily interpreted. This issue is often cited as the reason why NNs are not used in domains where the transparency of the prediction of a model is critical. However, in the case of density forecasting where the prediction is in fact a function, the interpretation of the meaning of any model type is questionable even if it were attainable.

In this section, a number of different NNs are described. In order to allow the reader to relate the different approaches with each other the architecture of each is shown and a universal legend is used, see Figure 2.2.

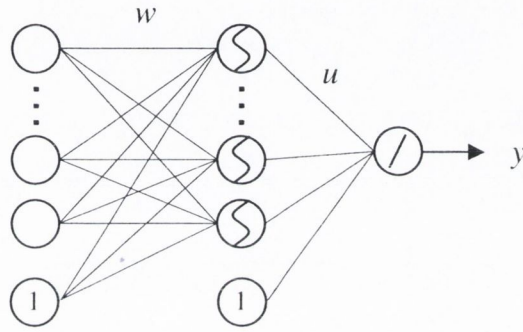
- represents a standard node
- ⊗ is the transfer function:  $\delta(x) = \frac{1}{1+\exp^{-x}}$
- ① is the bias term
- ⌞ is a linear function:  $l(x) = \sum_i x_i$
- Ⓢ is a softmax function:  $s(x_j) = \frac{\exp x_j}{\sum_i \exp(x_i)}$
- ⊗ is a Gaussian density function

**Figure 2.2:** Shown are the different components that can be used to design a NN.

The most commonly used neural network is the feed-forward Multi-Layer Perceptron (MLP). This is typically trained using a back-propagation learning algorithm. Figure 2.3



shows the basic design of an MLP. Here there is one hidden layer, however, there can be more than one hidden layer in an MLP. The various parts of the architecture are often named after their biological equivalents. The nodes of the hidden layer are called neurons and the connecting edges are called synaptic weights. The MLP receives an input pattern  $\mathbf{x}$  which is propagated forward through the network. As it passes through the network the input patterns are transformed to produce an output pattern  $y$ . Simply put, the neural network represents a function  $f$  that maps the input patterns to the output patterns,  $f(\mathbf{x}) \mapsto y$ .



**Figure 2.3:** The standard set up for a feed-forward MLP with one hidden layer. The network weights are denoted by  $w$  and  $u$ . The outputs from the hidden layer combined to produce the function estimate,  $y$ . This can then be used as a prediction for the target variable.

The hidden layer, so called because the user does not usually observe the inputs and outputs of this layer, has a specified number of neurons that represent transfer functions. The most common transfer function is the sigmoid function,  $\delta$ . In the case of  $j$  nodes the sigmoid function is;

$$\delta_j(\mathbf{x}) = \frac{1}{1 + e^{-(\text{bias}_j + \sum_{i=1}^{|\mathbf{x}|} w_{i,j} x_i)}} \quad (2.7)$$

where,  $w_{i,j}$  is the value of the synaptic weights for the edge from input value  $i$  to the  $j^{\text{th}}$  neuron and  $\text{bias}_j$  represents the value for the synaptic weight between the bias and the  $j^{\text{th}}$  neuron. The bias term is fixed for all patterns and has the same effect that the intercept term has in linear regression (Fieldsend, 2003).

The training process for an MLP proceeds as follows. Input patterns are presented to the network and propagated through the layers to produce an approximation of the target value  $y$ . Generally, the error function described in Equation 2.3 is used to determine the error in the model. This value is then propagated back through the network proportional

to the partial derivative of the error at that neuron. This process is based upon the *delta rule*, the cornerstone of the back-propagation algorithm. A full description of the derivation of the *delta rule* is included in Faussett (1994) or Bishop (1995). For the purpose of this short introduction it suffices to include the formula for weight updating;

$$\Delta w_{i,j} = -\eta \frac{\partial E}{\partial w_{i,j}} \quad (2.8)$$

where,  $\eta$  is a parameter set by the modeller called the learning rate. Each pattern in the training set is passed through the MLP and the weights are adjusted. Each full pass of the training set is known as an epoch and this process is repeated for a user-defined number of iterations.

There are a number of variations of the back propagation optimisation algorithm. Gradient descent is the simplest form but requires parameter setting by the user, see Section 4.2 for more information on optimisation methods. As mentioned above, the gradient descent method is a first derivative approach to optimisation. There have been a number of techniques suggested that are often called second derivative approaches. These attempt to reduce the convergence time during optimisation. Using the second derivative it is possible to find the optimum direction to search at each time step. However, finding and using the second derivatives are expensive procedures. Therefore, these approaches are not always better than standard gradient descent. Second derivative methods also require fewer if any parameters to be set by the modeller other than the architecture of the network. This is often of practical use as determining the best parameters for the gradient descent approach is often extremely difficult. For more information on second derivative methods see, for example, (Moller, 1993; Charalambous, 1992).

### Indirect Conditional Density Estimation

Attempts have been made by Specht (1990) and Neuneier et al. (1994) to model the conditional density  $p(t|x)$  by applying the definition:

$$p(t|x) = \frac{p(t, x)}{p(x)} \quad (2.9)$$

This is called the indirect approach because it does not directly model the conditional density. Instead, two networks are employed, one for modelling the distribution of the input vector,  $p(x)$ , the other for modelling the joint distribution of the input vector and the target,  $p(t, x)$ . The advantage of this approach is that it requires only learning the

unconditional probabilities, which can be accomplished by using, for example, a simple one-hidden-layer neural network, thereby considerably reducing the complexity of the optimisation problem. The downfall of this approach is that the predictions are poor relative to those achieved by a direct method (Husmeier, 1999).

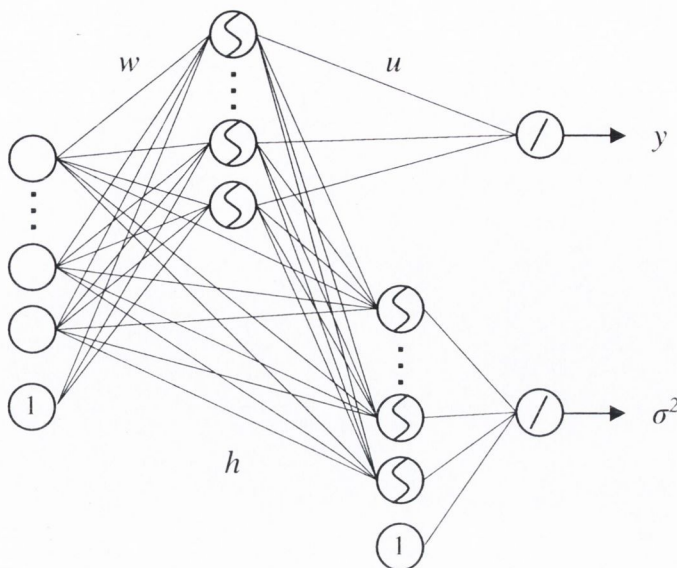
### Predicting Local Error Bars

The first direct attempt at making density forecasts using a connectionist approach can be attributed to Weigend and Nix (1994). In their paper, it is suggested that two one-hidden-layer networks can be used to predict a conditional mean value and the input-dependent standard deviation (local error bars). Their solution uses two connected networks, the first network takes the input vector and generates an output that represents the conditional mean estimate. The second network takes the input vector and generates an output representing the variance of the noise distribution. They suggest that training should be done in stages. First, they suggest training the network to learn the conditional mean, then the conditional variance and finally the conditional mean and variance together. Figure 2.4 shows how the two networks are connected. This approach is restricted by the fact that the underlying distribution is assumed to be Gaussian. The convenience of assuming a Normal distribution is attractive but it applies some restrictions, for example, the higher order moments of skewness and kurtosis are assumed to be regular. However, the training strategy is robust and performs well even with few data. Later, this approach was further developed by Nix and Weigend (1995).

Bishop and Qazaz (1996) improved on this approach by first integrating out the weights of the first hidden layer before optimising the weights of the second hidden layer. However, the input-dependent variance estimate is still restrictive in the sense that there is an assumption that the underlying noise distribution is Gaussian. This means that these models fail to produce accurate estimates of the conditional density when the true density is multi-modal or skewed.

In another paper, Weigend and Srivastava (1995) relate the density forecasting problem to predicting a ‘soft histogram’ of conditional probabilities for a set of predefined bins. This reduces the problem to a multi-class probability estimation task. Although some success was achieved and the restriction of assuming a Gaussian distribution had been removed, the difficulty with this approach is defining a good set of bins prior to estimation. In complex non-linear problems this approach would require a prohibitively large number of





**Figure 2.4:** The first network with weights  $w$  takes the input vector  $x$  and generates  $y$  which represents the regression function. The second network with weights  $h$  also takes the input vector  $x$ , and generates an output representing the variance of the noise distribution  $\sigma^2$ . Finally, all the weights are trained by maximum likelihood (Nix & Weigend, 1995).

bins in order to successfully capture the data generating process.

### Mixture Density Networks

The Mixture Density Network (MDN) was first introduced by Bishop (1994). The MDN is an adapted NN that produces an estimate of a conditional density function rather than a single point estimate. The predicted probability density function is presented as a Gaussian Mixture Model (GMM). GMM's provide a means of approximating any continuous distribution arbitrarily well by a finite mixture of normal densities with common variance (or covariance matrix in the multivariate case). They provide a convenient, semi-parametric framework in which to model unknown distributional shapes (Mclachlan, 2000) (see Figure 2.5). A GMM with  $k$  component densities is formulated as follows:

$$p(t|\mathbf{x}) = \sum_{i=1}^k \alpha_i(\mathbf{x}) \phi_i(t|\mathbf{x}) \quad (2.10)$$

where,  $\alpha_i(\mathbf{x})$  is the weight assigned to the density  $i$ , or *mixing coefficient*, the set of  $\alpha$  values must sum to 1, i.e.  $\sum_{i=1}^k \alpha_i(\mathbf{x}) = 1$ , so that the resulting GMM is a correct probability



density.  $\phi_i$  represents a Gaussian density function;

$$\phi_i(t|\mathbf{x}) = \frac{1}{(2\pi)^{1/2}\sigma_i(\mathbf{x})} \exp \left\{ -\frac{\|t - y_i\|^2}{2\sigma_i(\mathbf{x})} \right\} \quad (2.11)$$

where,  $y_i$  represents the mean of the  $i^{\text{th}}$  Gaussian component.

Figure 2.6 shows the basic architecture of an MDN. In this case a two component GMM is predicted for each target. This requires the network architecture to have six output variables that represent the mixing coefficient, mean and variance of the 2 component Gaussians.

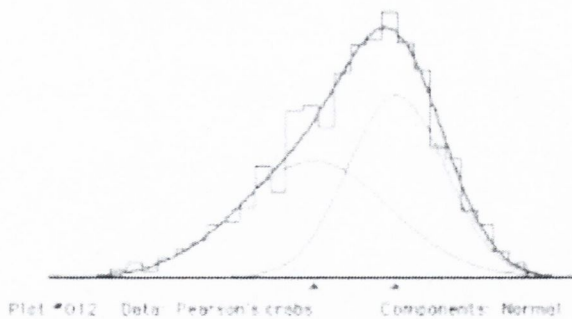
Due to the fact that the outputs of the NN represent constituent parameters of a density function, each output must conform to the constraints of those parameters. To achieve this the following transforms are applied to the NN outputs. The mixing coefficients are transformed by way of the ‘softmax’ function (Jacobs et al., 1991);

$$\alpha_i = \frac{\exp(\alpha_i(\mathbf{x}))}{\sum_{j=1}^k \exp(\alpha_j(\mathbf{x}))} \quad (2.12)$$

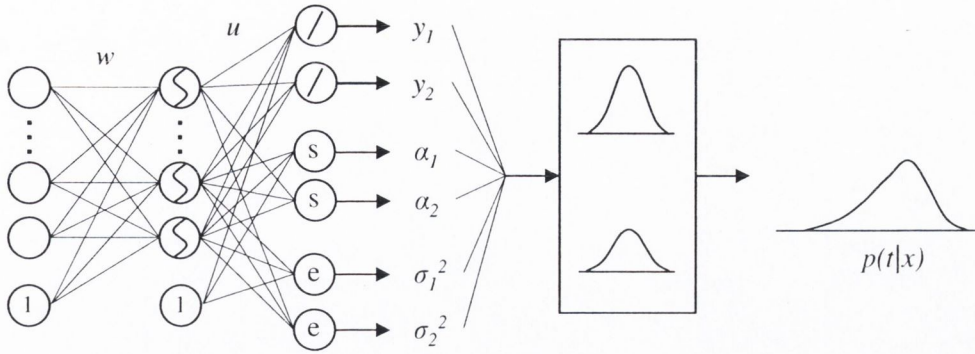
The softmax function ensures that the quantities  $\alpha_i$  are positive and are values between 0 and 1.

Similarly, the variance of each component probability density function must be a positive value. To constrain the values to be positive they are put through an exponential transformation;

$$\sigma_i^2 = \exp(\sigma_i^2) \quad (2.13)$$



**Figure 2.5:** This figure shows the first Gaussian Mixture Model as discovered by Pearson. While studying a sample of crab body lengths in 1896 he discovered that the histogram of values could be accurately described by a weighed combination of two Gaussian distributions. Using this information he was able to divide the species of crab being studied into two separate sub-species.



**Figure 2.6:** The Mixture Density Network architecture is very similar to the standard MLP shown in Figure 2.3 with the exception of the output weights. In this example a 2 Gaussian component mixture model is estimated. The output vector is combined to produce a GMM that represents the predicted conditional density.

Finally, the predicted mean of each component Gaussian is the only output that is not transformed.

It is the negative log-likelihood error function (see Section 3.2.1) that is used to determine the error of each pattern presented to the network. The contribution to error of the  $q^{\text{th}}$  pattern presented is;

$$E_q = -\log \left\{ \sum_{i=1}^k \alpha_i(\mathbf{x}_q) \phi_i(t|\mathbf{x}_q) \right\} \quad (2.14)$$

Bishop (1994) gives a full derivation of how this error is back-propagated through the MDN.

A major difference between the simple point estimation that gradient descent is renowned for and the task of estimating the conditional probability is the complexity of the local curvature of the error function. The simple approach of gradient descent, following the negative gradient to reduce the error in the network, performs extremely poorly when optimising an MDN. The common gradient descent problems are encountered much more frequently e.g. getting stuck in a local minimum or, extremely long convergence times. This is usually ameliorated by increasing the learning rate of the NN, however, this results in either all but one of the mixing coefficients or all but one of the covariances reducing to zero, or both. This means only one Gaussian remains and the network can only predict the conditional mean. Conversely, a small learning rate results in all the Gaussians becoming identical and fitting the same part of the conditional density function.

Bishop (1994) avoids the problems above by using a more complex optimisation procedure than simple gradient descent. Specifically, he uses the Scaled Conjugate Gradient (SCG) method of Moller (1993). This is a supervised learning algorithm that implements super-linear convergence. SCG is fully automated with no user dependent parameters and avoids a time consuming line-search. It achieves this by using the Hessian to find a suitable search direction and to adjust the learning rate (step size). The idea behind conjugate gradient methods is to choose search directions that complement each other. This avoids the possibility of undoing the minimisation of previous iterations. The Hessian matrix is a matrix of second order derivatives of the error with respect to the network weights. It contains information on how the gradient changes in different directions in weight space. In Section 4.2 gradient and non-gradient approaches to optimisation are discussed in detail.

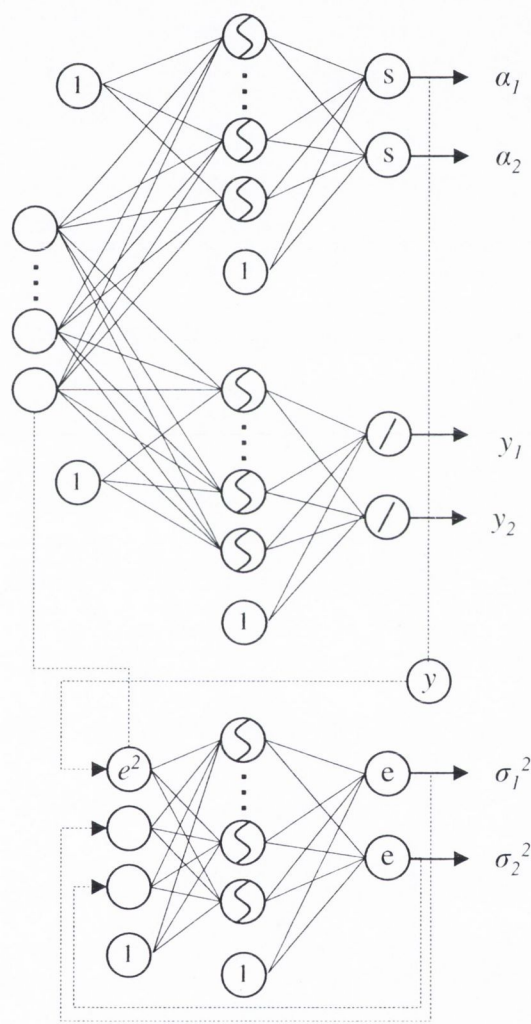
It is interesting to briefly mention at this point that in Chapter 4 an evolutionary computation method of optimisation of the MDN will be described. Evolutionary computation is exceptionally good at finding global minima for complex error surfaces. Although, the motivation for using an evolutionary optimisation technique is for reasons other than the local error complexity, this alone would suffice in justifying an evolutionary approach.

MDNs have been applied successfully to solve many problems. In Bishop's original work he demonstrates how MDNs can successfully solve multi-modal problems with a synthetic and real world kinematics example. Later, Cornford et al. (1999) applied MDNs to the meteorological problem of wind vector prediction. Richmond (2001) uses an MDN to perform an acoustic-to-articulatory inversion. This is the difficult procedure of mapping acoustic signals to some articulatory space that relates to physiological parts of voice. Their investigation showed that the MDN is very well suited to delivering the required functionality for performing the inversion mapping. An application of MDNs for surf prediction is described in the paper (Carney et al., 2005).

Another popular area that MDNs are applied to is finance; a survey follows. Locarek-Junge and Prinzler (1997) apply MDNs to the problem of risk management. They suggest that the MDN is a promising new approach for providing accurate estimates of risk. The MDNs ability to predict non-Gaussian conditional densities is highlighted as the most desirable characteristic of the model. Schittenkopf et al. (1998) evaluates the performance of MDNs with that of GARCH models (see Section 2.3.5) on the problem of predicting the variance (volatility) of daily data taken from the Austrian stock exchange. The paper demonstrates that the MDN performs well in cross validation but poorly when test sets



are over long periods. In (Schittenkopf et al., 2000), they adapt the MDN framework so that it has a recurrent structure. Figure 2.7 shows the structure of a Recurrent MDN. The paper compares results obtained, using historical data from the FTSE 100, with results achieved from GARCH models that assume different distributions. The models are evaluated using seven different error measures on five disjoint sets out-of-sample. No model shows a statistically significant performance improvement, although, the GARCH- $t$  ( $Student-t$  distribution) and the Recurrent MDN perform best on all tests.



**Figure 2.7:** Recurrent Mixture Density Network as described in (Schittenkopf et al., 2000). The network has been adapted such that the variances ( $\sigma^2$ ), means ( $y$ ) and mixing coefficients ( $\alpha$ ) are each given a separate MLP. The most notable change however, is the fact that the residual error from the mean values is included as an input to the MLP that predicts variance, this is represented in diagram as a node labelled  $e^2$ . This is similar to the way Weigend and Srivastava (1995) estimates variance in their local error bar model



Bartlmae and Rauscher (2000) apply the MDN to the problem of market risk in the German stock index. They propose a number of methods of evaluating results and show that the MDN model compares favourably with the performance of a GARCH model. Schittenkopf and Dorffner (2001) adapt the MDN further, this time to include “short cut weights” that bypass the hidden layer. In this paper the application is to predict options prices for the FTSE 100. They show that their Recurrent MDN model performs significantly better in out-of-sample pricing in comparison to the basic and extended Black-Scholes<sup>3</sup> model. Finally, Miazhyńska et al. (2003) present a variety of extensions of the Recurrent MDN and test these models in the context of a risk management application. Most notably they introduce an MDN with linear hidden layer transfer functions. The best model over all tests and datasets was a linear MDN.

The Conditional Density Estimation Network (CDEN) (Ormoneit, 1993) is another method of predicting conditional probability density functions using an MLP. It was first suggested contemporaneously with the MDN and it takes an almost identical form. Again, this approach was mainly applied in the domain of finance, see, for example (Ormoneit & Neuneier, 1996; Neuneier, 1995). For simplicity throughout the thesis MDN will refer to a model of type CDEN or MDN as they are equivalent.

## Ensemble of Mixture Density Networks

Carney et al. (2005) apply MDNs to the problem of surf prediction. It is demonstrated that the additional information that one garners from the whole conditional probability is useful when decision making. Further, it is shown that by using an ensemble method the results of the MDN models can be improved and the likelihood of poor performance due to sporadic non-convergence can be reduced. Two approaches were employed, bagging and boosting.

*Bagging* is an abbreviation for bootstrap aggregation. It uses the bootstrap, a statistical re-sampling technique, to generate multiple, diverse, training sets for networks of an ensemble. A bagged neural network ensemble generates training data for each ensemble member by sampling from the bootstrap empirical distribution. A training set is created for each neural network in the ensemble and the networks are trained on this data.

As mentioned above, sporadically, MDNs do not converge to a good solution due to a very poor random starting position. To reduce the effect of these weak solutions on

---

<sup>3</sup>Black-Scholes is the standard method of pricing options. See (Scholes, 1973) for more information.

the ensemble prediction we integrate the member solutions using a form of balancing (Heskes, 1997a). The training sets in a bagged ensemble are generated by sampling with replacement from the original training set  $T$ . The probability that the individual training vector, taken from  $T$ , will not be part of a bootstrap re-sampled training set,  $T^{bootstrap}$ , is  $(1 - 1/N)^N \approx 0.368$  where  $N$  is the number of training vectors in  $T$ . These omitted training vectors make a new out-of-bootstrap (oob) test set,  $T^{obb}$ , for the ensemble member trained with  $T^{bootstrap}$ . For each ensemble member the error  $E^{obb}$  is calculated over the vectors in its  $T^{obb}$ . We use these error values to weight each members contribution to the ensemble prediction. Assuming there are  $M$  ensemble members;

$$p(t|\mathbf{x}) = \sum_{i=1}^M \sum_{j=1}^C \frac{E_i^{obb} \alpha_{i,j}(\mathbf{x})}{\sum_{k=1}^M E_k^{obb}} \phi_{i,j}(t|\mathbf{x}) \quad (2.15)$$

where,  $E$  is the Continuous Ranked Probability Score error function for assessing density forecasts, (see Section 3.2.2). The new aggregated GMM produced by the ensemble still obeys the properties of a probability distribution.

*Boosting* is another ensemble method. (Carney et al., 2005) implement the *Adaboost.R2* algorithm for boosting for MDNs (Drucker, 1997). The means of obtaining diverse training sets is distinctly different in boosting to bagging. In boosting, training of ensemble members is carried out sequentially. The vectors that have been ‘difficult’ for ensemble members to estimate i.e. obtain large errors, are more likely to appear in the training sets of subsequent ensemble members. This restricts ensemble members from being trained in parallel, however, it has the advantage of reducing the bias of the resulting ensemble.

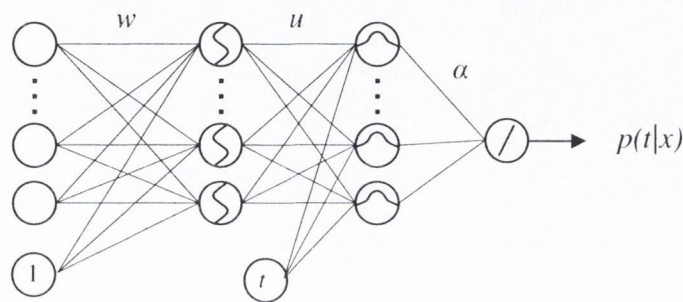
To combine ensemble members a weighted median approach is used, the better performing members are assigned a heavier weighting. Integration is achieved in a similar way to the balancing method above; however, instead of using out-of-bootstrap vectors alone the whole training set is used to calculate the weightings.

## GM Random Vector Functional Link

Husmeier and Taylor (1998); Husmeier (1999) introduce the Gaussian Mixture Random Vector Functional Link (GM-RVFL) network. This uses a two hidden layer neural network to make estimates of the conditional density. Husmeier (1999) argues that in order to be a truly universal approximator the NN should at least have a second hidden layer. The GM-RVFL also differs in its training methodology, the target variable,  $t$ , is included as



input data as the aim is to predict the conditional probability density function and not the target variable. Another distinct difference between this approach and others is that the second hidden layer consists of Gaussian distributions. Therefore, the aim of the network is to mix the Gaussian distributions of the second layer in order to model the distribution of the target. The conditional variance of each of the Gaussians is adapted to the whole set of training data while the conditional mean and the mixing coefficients depend on the network inputs.



**Figure 2.8:** GM-RVFL is a two hidden layer MLP. The second layer contains nodes that are Gaussian distributions.

### 2.3.3 HMM Models

If the regression problem is embedded in a time series or sequential data it is possible to use Hidden Markov Models (HMM) to estimate a conditional probability density function. HMMs use the concept of a state space to use historical data to make predictions of future events. The state space contains all possible states - similar states are close to one another and dissimilar states are far away from one another within state space. Predictions are made by estimating how close conditions in the past were to the present conditions and using the most similar states to estimate what will happen in the future. The structure of HMMs is similar to finite state machines but both transitions and outputs are probabilistic. A Markov process is used to model the training data. This is a commonly used method in probability theory for representing a specific type of stochastic process.

The estimation of the conditional density function for HMMs uses a density estimation method called the Expectation-Maximisation algorithm (Baum et al., 1970). This estimation method is typically used for density estimation in classification tasks and, therefore, is quite different to the other approaches described in this chapter. For a complete de-



scription and explanation of how HMMs can be used for density forecasting see (Fraser & Dimitriadis, 1993).

### 2.3.4 Gaussian Processes

The goal in most regression scenarios is to build a function that models the available data as well as possible to make inferences about new unseen examples. Most point forecasting models try to describe the conditional mean dynamic of the data in the form of a function. A Gaussian process (GP), however, models data by giving a prior probability to every possible function with higher probabilities given to functions that are more likely. This is achieved by modelling the data with a special type of stochastic process where the underlying distribution is a Gaussian.

Rasmussen and Williams (2005) define GPs as, “... a collection of random variables, any finite number of which have a joint Gaussian distribution”. Rather than a standard Gaussian distribution or multivariate Gaussian distribution that describes random variables which are scalars or vectors, a Gaussian process governs the properties of functions. This structure is very suitable for density forecasting in regression. A GP defines a distribution over functions and, when correctly specified, inference can take place directly in the space of functions. Typically, this concept of a GP for a real process,  $f(\mathbf{x})$ , can be described by its mean function,  $m(\mathbf{x})$ , and covariance function,  $k(\mathbf{x}, \mathbf{x}')$ , as follows:

$$m(\mathbf{x}) = E[f(\mathbf{x})], \quad (2.16)$$

$$k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \quad (2.17)$$

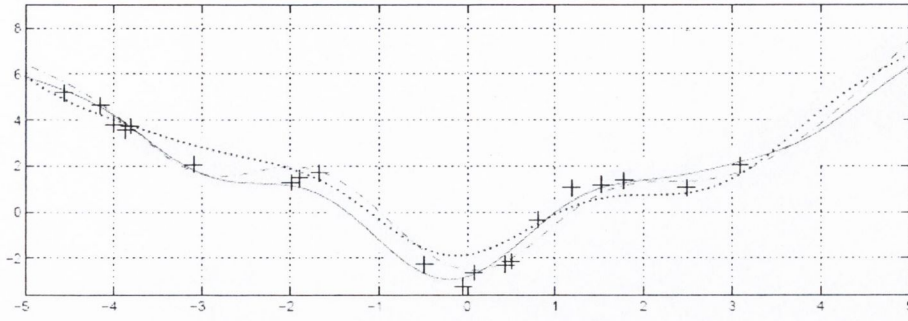
with the GP defined as:

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (2.18)$$

With this setup the modeller is contending with seemingly infinite dimensional objects, however, GPs can produce a model that resides in this space while requiring only to work with finite dimensional objects. A full explanation of how a GP achieves this is beyond the scope of this section and the reader is referred to (Rasmussen & Williams, 2005) for details. Figure 2.9 shows how a GP models a set of data.

### 2.3.5 ARCH and GARCH models

The standard tools employed for the purpose of estimating conditional variance in finance are ARCH (Engle, 1982) or GARCH (Bollerslev, 1986) type models. ARCH stands for au-



**Figure 2.9:** Shown is the output from a Gaussian Process. There are twenty data points(+). Three functions drawn at random from the posterior found after the GP is trained on the data. The shaded area shows the 95% confidence region. This example is taken from (Rasmussen, 2004).

to regressive conditional heteroskedastic and GARCH is Generalized ARCH. The precursor to ARCH was a rolling standard deviation calculated using a fixed number of the most recent observations. For example, tomorrow's volatility would be calculated as an equally weighted average of the past months squared residuals. Engle (1982) extends this idea with ARCH by accounting for the intuition that residual values from one month are less indicative of tomorrow's volatility than more recent observations. He also accounted for the fact that setting a specific time window, such as the one month period in the example, is not intuitive as events that happened more than one month ago may still be applicable. To account for these factors, the ARCH model lets these weights be parameters to be estimated allowing the historical data to decide which weights are best to forecast the conditional variance.

Although the ARCH model was the first to set its parameters empirically, the GARCH model, an improved formulation of the ARCH model, is the model most commonly used in industry. The weights used in the GARCH model gradually decline over time but never go to zero. The simplest GARCH model can be described as three parameter weights on  $\kappa$  the long-run average variance,  $\beta$  the new information for this period that is captured by the most recent squared residual and  $\alpha$  the variance predicted for this period (Engel, 2001). The GARCH model allows weightings to be directly applied to the  $P$  previous  $\alpha$ 's and the  $Q$  previous  $\beta$ 's. Formally, the GARCH model is described as;

$$\sigma_t^2 = \kappa + \sum_{i=1}^P \alpha_i \sigma_{t-i}^2 + \sum_{j=1}^Q \beta_j \epsilon_{t-j}^2 \tag{2.19}$$

The weightings  $\alpha$  are applied to the  $P$  previous forecasts  $\sigma_{t-\{1\dots P\}}^2$  and the  $\beta$  weightings are applied to the last  $Q$  periods' squared residuals  $\epsilon_{t-\{1\dots P\}}^2$ . This type of model is commonly accompanied by a constant mean model.

$$y_t = C + \epsilon_t \quad (2.20)$$

This equation contains the final parameter  $C$  for our decision vector. The vector representation of a GARCH(P,Q) takes the form  $\mathbf{x} = (C, \kappa, \alpha_1, \dots, \alpha_P, \beta_1, \dots, \beta_Q)$ . This is the representation that will be used by our optimisation procedure described in Chapter 6.

From this simple formulation a vast number of extensions have been suggested, for example see (Nelson, 1991; Higgins & Bera, 1992; Glosten et al., 1993; Engle, 2002). One of the most commonly used of these extensions is the GJR model or Threshold GARCH (Glosten et al., 1993). The GJR model adds the capability of a GARCH model to take into account another stylized fact of equity return series; positive or negative residuals have different impacts on volatility. For example, it is common that negative news causes a large drop in equity prices but positive news causes only a small increase. The GJR model is used extensively in Chapter 7 and so the formulation of the GJR model is included here.

$$\sigma_t^2 = \kappa + \sum_{i=1}^P \alpha_i \sigma_{t-i}^2 + \sum_{j=1}^Q \beta_j \epsilon_{t-j}^2 + \sum_{j=1}^Q L_j S_{t-j}^* \epsilon_{t-j}^2 \quad (2.21)$$

Where,

$$S_{t-j}^* = \begin{cases} 1 & \epsilon_{t-j} < 0 \\ 0 & \text{otherwise.} \end{cases}$$

This model has one additional parameter  $L$  the leverage term, all other parameters are the same as the GARCH(P,Q) model. The leverage term captures the characteristic asymmetry that is seen in many return series. Again, the model can be accompanied by a constant mean model. Therefore, the vector representation of the GJR(P,Q) model takes the form  $\mathbf{x} = (C, \kappa, \alpha_1, \dots, \alpha_P, \beta_1, \dots, \beta_Q, L_1, \dots, L_Q)$ .

Finally, it is not only the conditional variance model that can be predicted, various conditional mean models can be used. Again, in Chapter 7 an ARMA(R,M) conditional mean model is used. Similar to ARCH models, the ARMA model is a weighted combination of the autoregressive and moving average coefficients.

$$r_t = C + \sum_{i=1}^M \phi_i r_{t-i} + \sum_{j=1}^M \theta_j \epsilon_{t-j} \quad (2.22)$$



where,  $\phi_i$ , is the autoregressive coefficients,  $\theta_i$ , is the moving average coefficients, the residuals are,  $\epsilon_t$ , and returns  $y_t$ . The vector representation of this ARMA model is;  $\mathbf{x} = (C, \phi_1, \dots, \phi_j, \theta_1, \dots, \theta_j)$ .

Users of GARCH models generally perform a series of preestimation analysis tests on the data to be modelled. These tests can identify the optimum model specification for the data. The model is then optimised using maximum likelihood, specifically, a constrained nonlinear optimisation<sup>4</sup>. Variations on the particular negative log-likelihood objective function are used depending on the chosen distribution. With this basic setup there are a number of different possible implementations, for example see (McCullough & Renfro, 1998; Brooks et al., 2001). In this thesis, the Matlab implementation of the GARCH optimisation process is used. This uses a quasi-Newton optimisation algorithm. For details, see (MathWorks Matlab, 2002).

### 2.3.6 Meteorological Models

Meteorology is the study of weather processes in the atmosphere. Since numerical weather predictions began in the 1950s meteorologists have focused on providing density forecasts as opposed to point estimates of future weather conditions. Within the field of weather forecasting there are many sub problems such as wind, temperature or rainfall prediction. The density forecasting models designed to predict these events use a mixture of complex physical models of the atmosphere coupled with observational data to make extremely accurate estimates of future weather conditions. Data from weather stations, satellites and weather aircraft are used as inputs to these models to produce the forecasts that we hear daily on the television and radio. Examples of these meteorological prediction models include (Molteni et al., 1996; Hodur, 1997; Kurihara et al., 1997).

Due to the complexity of the models and the underlying problems, a review of the techniques used for density forecasting in meteorology is beyond the scope of this document. However, it is important to note that density forecasting is widely used in this field. For example, the work of Adrian E. Raftery (97) has many parallels with the theories described here and was the inspiration for much of the work. Indeed, much of the peripheral research on density forecasting models cited in this thesis comes from the weather forecasting field, such as density forecast model evaluation. In future work, it would be interesting to apply the techniques suggested here on meteorological models.

---

<sup>4</sup>Also known as nonlinear programming.

## 2.4 Discussion and Summary

In this review section a number of density forecasting methods have been introduced. Prediction intervals, one of the most commonly used methods of describing uncertainty, were briefly described. Although, often described as an alternative, prediction intervals are in fact a sub-field of density forecasting. Numerous different methods of estimating prediction intervals exist. However, the form of distribution used is restricted in most cases to one of a small number of possibilities e.g. Gaussian, Student t, etc. For sequential data, HMM models can be used to make accurate estimates of the conditional density function.

There exists a number of different density forecasting methods within the neural network class of models. The flexibility of neural networks as universal approximators makes them suitable for tackling the complex task of estimating conditional densities. Weigend and Nix (1994) produced the first density estimator. However, their approach was restricted to a Gaussian estimate. Bishop (1994), and contemporaneously Neuneier (1995), introduced the Mixture Density Network, a method of estimating conditional probability density forecasts with complex distributions. This flexible base model has been adapted by many researchers to produce accurate forecasts for different prediction problems. GM-RVFL models are another neural network based probability density function estimator that performs comparably well with MDNs. Indirect methods of density forecasting have also been discussed, however, these approaches are unpopular because they do not produce good models. Gaussian Processes have also been briefly introduced and described. GPs offer an alternative methodology to the other forms of density forecasting, however, as yet they are restricted to a specific type of distribution and so do not offer the same flexibility as other methods. Other possible approaches include Radial Basis Functions (Broomhead, 1988) or Generalised Regression Neural Networks (Specht, 1991).

In terms of the common machine learning approaches to density forecasting described in this review the Mixture Density Network is the most versatile and accurate model. For these reasons the MDN is used as the density forecasting model of choice in the experiments and evaluations that follow. For time-series data GARCH models are used to estimate the conditional density of the series. This method has been chosen because it is an example of a density forecasting model that is used as an industry standard. There are numerous research papers that use, extend and benchmark GARCH type models. Therefore, the

impact of the proposed approach in this thesis can be fully show-cased using GARCH type models within the important application domain of finance.



## Chapter 3

# Density Forecast Evaluation

### 3.1 Introduction

Getting a forecast right is invariably a difficult exercise. Naturally, the more ambitious the prediction, the more difficult it is to get it right. Therefore, an important aspect of the prediction process is the evaluation step where the ability of a model to predict is quantified. For density forecasts, evaluation of the predictions from a model is complicated by the fact that the prediction is not in the same form as the target. The true conditional density is never observed, instead only a single random realisation of the true conditional density function is available to the forecaster to evaluate a model.

By determining the evaluation metrics that correctly identify a good probabilistic distribution the current standard optimisation approach to density forecasting can be analysed and the issues associated with it can be highlighted. In this chapter, the general concepts of density forecast evaluation are introduced and the most widely used metrics are described.

### 3.2 Calibration and Sharpness

The motivation for predicting probability density functions over point estimates is to give the prediction user additional information regarding the probability of different events occurring. Therefore, this additional probabilistic information must be correct to justify the density forecasting model. Intuitively, a correct probabilistic estimate of  $P\%$  for an event would result in that event occurring  $P\%$  of the time in the long run (Dawid, 1984). This characteristic of density forecasts is known as *calibration*.

Although having a model that is calibrated is a high priority for any density forecaster, there are two practical problems with assessing this characteristic. First, predictions can only be assessed on a finite set of outcomes. Second, and more importantly, there is no inherent reason why a calibrated model should produce predictions that bear any relation whatsoever to the actual outcomes (DeGroot & Fienberg, 1983). In fact, a forecaster can be made to be calibrated by biasing their true opinion to correct for calibration inaccuracies. Furthermore, Murphy and Winkler (1977) and Hamill (2000) have shown that even if a forecaster is calibrated, its predictions are not necessarily accurate in all respects or useful. For example, if over the long run it was observed that Microsoft’s closing price is higher than the previous days closing price 55% of days, then a simple density forecaster that always predicted that the stock price would rise with 55% probability is calibrated but not useful. This example is known as a model that predicts the *unconditional distribution* of the target variables and represents the point at which forecasts are not predictive. In meteorology, this is known as the climatology.

The Probability Integral Transform (PIT) is a popular method of calibration evaluation across all forms of density forecasting; for example, Schervish (1989) uses it for evaluation of classification problems; Christoffersen (1998) adapts it for prediction intervals; and there is an equivalent method of evaluation for ensembles of point forecasters (Talagrand et al., 1997). PIT overcomes the problem of loss function inconsistencies across problem domains by not depending on a specific user loss function. Instead, the PIT assumes that the prediction user is attempting to estimate the true conditional density function. This is a fair assumption as the true conditional density is at least as good as, if not better than, any other possible model i.e. it weakly dominates all other models. In the case where the true conditional densities are being assessed, the set of cumulative densities at the realisations will be uniform<sup>1</sup>. By making the assumption that we are striving to find the true conditional densities we can assess the set of predicted cumulative densities at the observations for uniformity. The PIT is described as:

$$z_i = \int_{-\infty}^{t_i} p(u|x_i)du \quad (3.1)$$

For a series of length  $M$ , a correctly calibrated model should have a set of  $z$  values in the same form as would be the case for the set of true conditional densities. Rosenblatt (1952)

---

<sup>1</sup>In the case of time series or sequential data the cumulative densities should also be independently and identically distributed. An excellent introduction to the PIT for sequential and time series data is provided in (Diebold et al., 1998b).

shows that this sample will be  $U(0, 1)$  and *i.i.d.* for the true conditional densities and any correctly specified density forecasting model.

A recent paper (Gneiting et al., 2005a), analyses the concept of calibration for regression problems. Gneiting et al. (2005a) identify three different modes of calibration: probabilistic, exceedance, and marginal. They show how these different modes of calibration are independent of each other and can occur in any combination. The PIT score above is a method of evaluation of probabilistic calibration. In their paper, Gneiting et al. (2005a) identify two occurrences where probabilistic calibration suggests a model is calibrated but is not a useful forecaster. The first scenario, introduced above, is when the predicted distribution is simply the unconditional distribution/climatology. The second scenario is when the forecaster is ‘unfocused’. This discovery comes from Hamill (2000) where it is shown that it is possible to find a combination of predictions that are well calibrated but completely non-predictive.

To identify scenarios where these situations occur a second evaluation metric can be introduced. When producing density forecasts a secondary objective is to make predictions that have minimum variance. This is called sharpness and refers to the concentration of predicted densities. It is obvious that evaluating this characteristic will identify models that estimate the unconditional distribution as the more spread out the average forecast becomes the closer it gets to the unconditional distribution and the worse sharpness the prediction will have. Similarly, an unfocused forecaster will have poorer sharpness than the correct conditional density, however, the difference is subtle and the prediction user should be aware of this issue.

Combining the objective of statistical consistency of probabilistic forecasts and the objective of minimum variance of the predictions, the goals of any density forecasting model should be to *maximise sharpness subject to calibration* (Gneiting & Raftery, 2004). Since these are the goals of density forecasting models they should also be the objectives when the model is being optimised. In the following sections a number of common density forecast scoring metrics that aim to optimise models correctly in terms of sharpness and calibration are discussed.

### 3.2.1 Negative Log Predictive Density

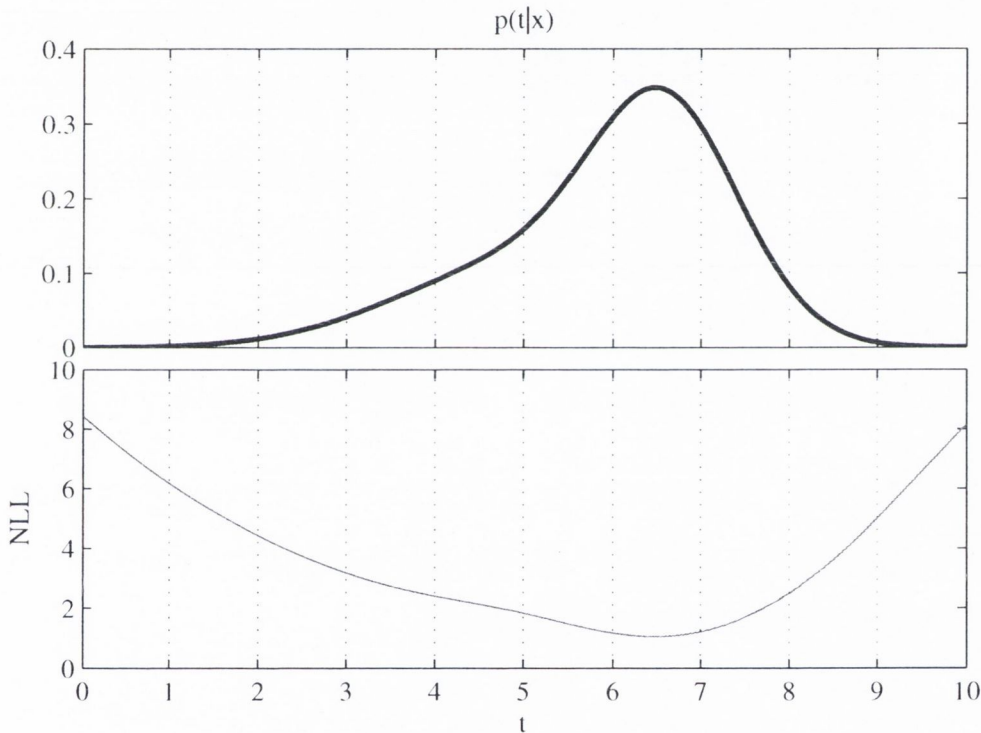
The Negative Log Predictive Density (NLPD), also known as the Ignorance Score (Good, 1952), Logarithm Score (Gneiting et al., 2005a) or the Negative Log-likelihood (NLL) in



the case of measuring error on training data (see Section 2.2.1), is a method of assessing the quality of a predicted density function. It is specified as follows:

$$NLPD_i = -\log(p(t_i|x_i)) \tag{3.2}$$

Simply put, the NLPD is the logarithm of the predictive density evaluated at the target. It can be easily and cheaply calculated and is by far the most popular error function in the density forecasting literature for this reason. The NLPD or a variation on the NLPD is almost exclusively used as the optimisation error function of density forecasting models. In Section 2.2.1, the NLL emerged as an error function from maximum likelihood (ML) theory.



**Figure 3.1:** The top plot depicts a sample predicted density. The bottom plot shows the NLPD for a target at each point on the  $t$  dimension for the given sample density. The relationship between density and contribution to error can be clearly seen.

Figure 3.1 plots the relationship between the NLPD value for a sample prediction density against all possible outcomes in the interval  $[0, 10]$ . The NLPD is a negatively oriented score, meaning the smaller the NLPD value the better. Although the NLPD attempts to simultaneously address both sharpness and calibration, the ability of the metric to assess sharpness out-weights its ability to assess calibration. In Section 5.2, this

bias for sharp models is shown by way of example. For this reason, this metric can be thought of as a sharpness score rather than a calibration score.

### 3.2.2 Continuous Rank Probability Score

The Continuous Ranked Probability Score (CRPS) (Hersbach, 2002) is a verification method for probabilistic forecasts of continuous variables. It is equivalent to the Brier Score (Brier, 1950) integrated over all possible values and is a generalisation of the Ranked Probability Score (Epstein, 1969) that is used to evaluate probabilistic predictions over ordinal variables. In essence, the CRPS measures the difference between the predicted and the occurred cumulative distributions, see Figure 3.2. In order for the score to be sensitive to distance, the squared errors are computed with respect to the cumulative probabilities of the forecast and observation. The CRPS is calculated as follows:

$$CRPS_i = \int_{-\infty}^{\infty} (p(u|x_i) - H(u, t_i))du \quad (3.3)$$

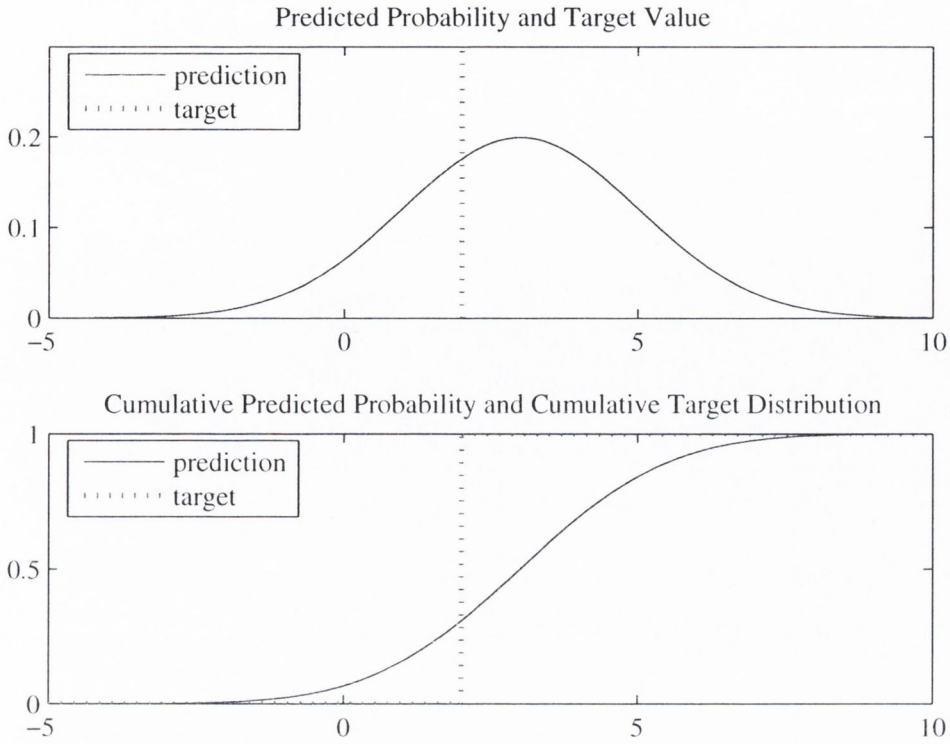
where,  $H$ , is the Heaviside function,

$$H(l, m) = \mathbf{1}\{l \geq m\}$$

Again, like the NLPD, the mean CRPS is calculated over all predictions to determine the average error.

Hersbach (2002) shows that the CRPS reduces to the Mean Absolute Error for deterministic forecasts. It is sensitive to distance i.e. it is capable of penalising predictions that are far away from the actual observation. Therefore, this evaluation technique is a means of comparing deterministic and probabilistic forecasting models and is also easy to interpret as an error measure. This interpretability is further aided by the fact that it is in the same units as the target variable.

Like the NLPD, the CRPS is a combined score addressing both sharpness and calibration. Gneiting and Raftery (2004) suggest that it is a more robust alternative to the NLPD and is therefore a better scoring metric. However, a major disadvantage of the CRPS score is that it is not of closed form. It is possible to derive a closed form version of the CRPS for normal distributions (Gneiting & Raftery, 2004); however, if the probabilistic model produces non-Gaussian solutions, such as mixture models, then determining numerical estimations of the integrals for every input pattern is a computationally intensive task. Furthermore, in Chapter 7 the CRPS is shown to be a poor estimator when data is leptokurtic performing worse than the NLPD score.



**Figure 3.2:** The top plot in this figure is the predicted probability density function for some regression task. The target value is at 2. The bottom plot shows the cumulative distributions for the target and prediction. The area between the target and predicted cumulative is the value returned by the CRPS score. The sensitivity of the CRPS to distance is linear relative to the predicted densities error. Sharpness (small spread) is rewarded if the forecast is accurate. A perfect CRPS score is 0.

### 3.2.3 Wilson Score

The Wilson Score (Wilson et al., 1999) is a means of assessing the quality of forecasts of continuous variables in terms of an acceptable range. For example, predictions of temperature may only need to be accurate between  $\pm 1^\circ\text{C}$  of the actual observation. This score evaluates predictions in terms of this range. Like the CRPS, the concept of the Wilson Score is derived from the Brier Score and Rank Probability Score. The Wilson Score determines the percentage of the forecasted probability that lies within the tolerable range of the observation/target. The equation for the Wilson Score is;

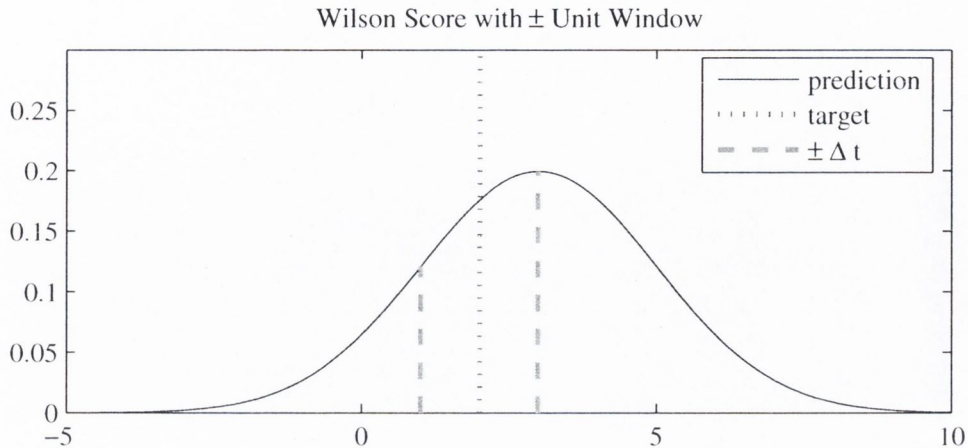
$$WS_i = \int_{t-\Delta t}^{t+\Delta t} p(u|x_i) du \quad (3.4)$$

where,  $\Delta t$  represents the threshold, or tolerated distance from  $t$ .

Figure 3.3 depicts the area contributing to the Wilson score for a sample prediction. This is a positively oriented score in the interval  $[0,1]$ . A perfect score receives a 1. The



numerical score that it produces can be understood as a probability. It is sensitive to distance and to the spread of the forecasts.



**Figure 3.3:** Shows the area of the distribution that is considered by the Wilson Score  $\pm \Delta t$ . In this example a tolerance, or window size of  $\pm 1$  units is used.

A weakness of this score is that the modality of the predicted distribution affects the results. The error score must be adapted to account for the different predicted modes. Wilson et al. (1999) describe how the score can be adapted by determining the modality of the target a priori. However, this is not a trivial task and may not even be achievable in certain scenarios. Another difficulty with this score, like the CRPS, is that it requires the calculation of an integral.

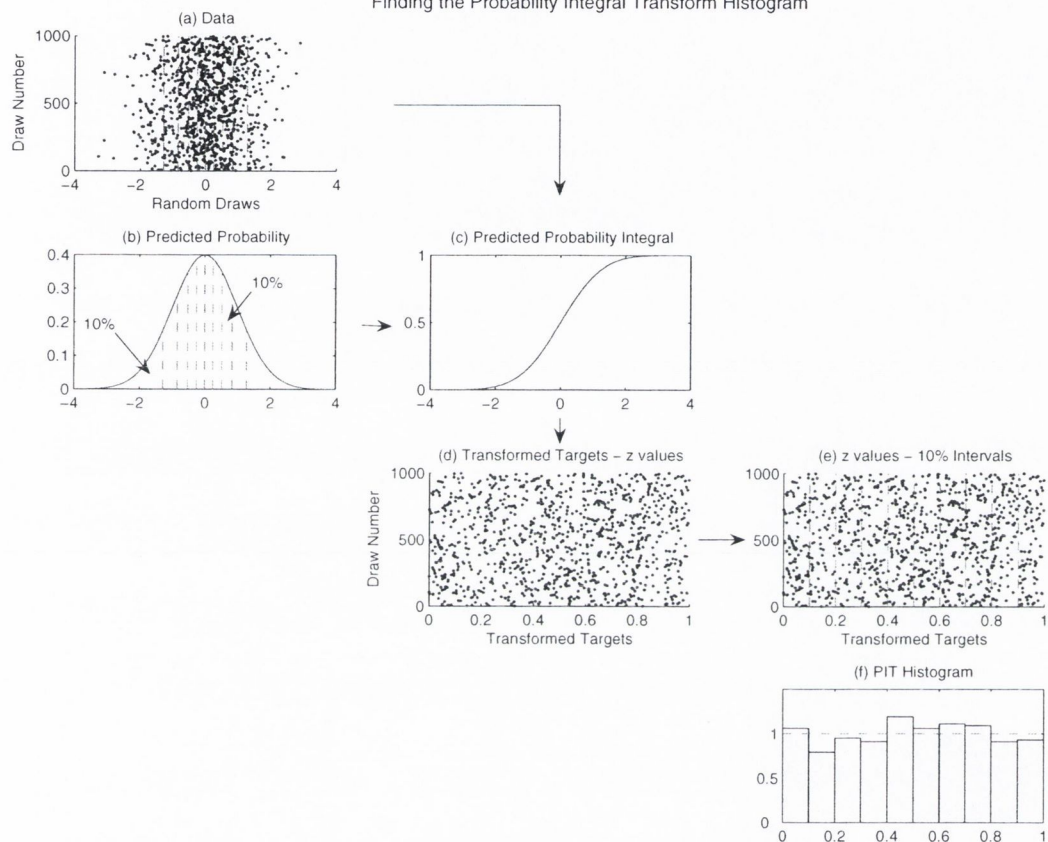
### 3.2.4 Probability Integral Transform Histogram

The PIT evaluation method is described above. As mentioned in the introduction to Section 3 the set of PIT values should be uniform in the interval (0,1). Diebold et al. (1998b) suggest that visual assessment of the set of PIT values in a histogram is a useful means of evaluating calibration. Figure 3.4 shows how the PIT  $z$  values are used to construct the PIT histogram for measuring calibration.

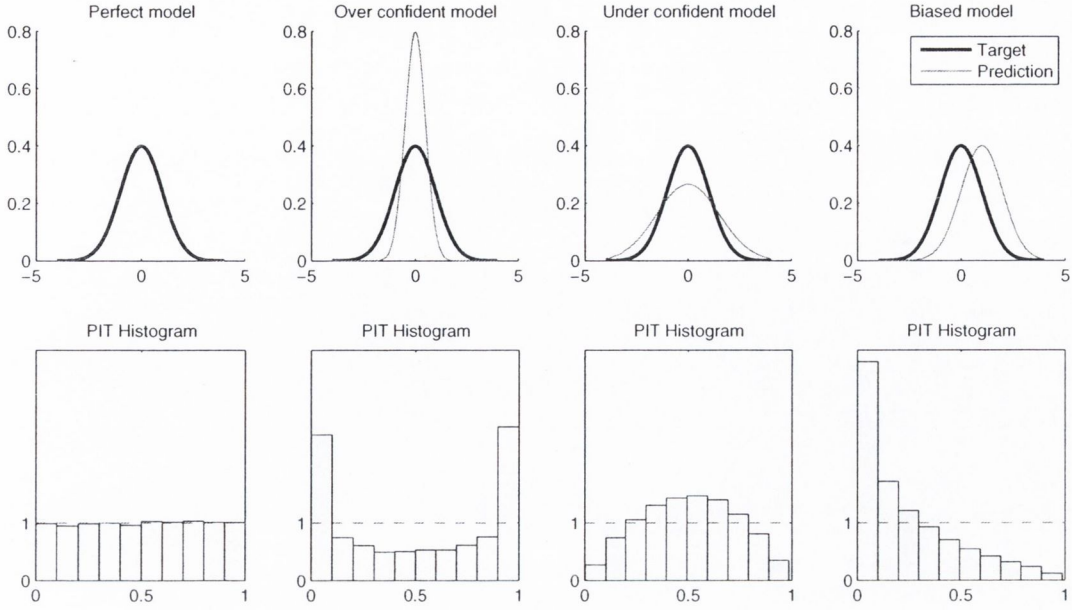
To further understand the PIT approach to determining calibration a demonstration of the effect of bias<sup>2</sup> and variance on the PIT histogram follows. To do this the problem is first simplified. A static distribution generates all data for testing, this is chosen to be a standard normal distribution,  $N(0,1)$ . Therefore, the true conditional density at every point is an  $N(0,1)$  density. Knowing the true density means the bias and variance of the

<sup>2</sup>In this experiment, bias refers to the incorrect specification of the mean of the predicted density.

Finding the Probability Integral Transform Histogram



**Figure 3.4:** This figure shows how the PIT histogram is constructed. A simple example is used. (a) 1,000 random draws from a standard normal distribution are used to represent the target variables. All predictions take the form of a standard normal distribution (b). The deciles are highlighted in both plots. (c) represents the probability integral transform of the predicted distribution. This is used for transformation of the predictions and targets to  $z$  values by Equation 3.1. (d) shows the resulting  $z$  values. (e) shows the  $z$  values and the deciles. Finally, (f) shows the resulting PIT histogram. The histogram shows the clearest picture of the calibration of the predictions. An almost flat histogram suggests a well calibrated model. See figure 3.5 for an explanation on how to interpret PIT histograms.



**Figure 3.5:** The top row of plots shows the true or target generating distribution ( $N(0,1)$ ) and the predicted distributions. From left to right the predicted distributions are,  $N(0,1)$ ,  $N(0, 0.5)$ ,  $N(0,2)$  and  $N(1,1)$ . The bottom row of plots represents the resulting PIT histogram for each prediction distribution determined by evaluation on 10,000 points generated from the target distribution. A perfect model is indicated with a dashed line.

predicted densities can be artificially simulated by making all predicted densities  $N(\mu, \sigma)$ . Bias is simulated by varying the  $\mu$  of the density and variance is simulated by varying the  $\sigma$  of the density. Figure 3.5 shows the distributions and resulting PIT histograms for different  $\mu$  and  $\sigma$  pairs. The first PIT histogram is the only correctly specified histogram because the predicted density is  $N(0,1)$ , the same as the distribution used to generate the data. Bias and variance effect the PIT histogram in different ways. Too narrow a variance forms a “U” shaped PIT histogram signifying over-confident predictions. Too wide a variance creates a hump in the middle of the PIT. This can be thought of as under-confidence. Bias causes a sloping effect and in the extreme case it creates a “J” or “L” shaped PIT histogram depending on the direction of the bias.

Although the PIT histogram identifies models that are under-confident and over-confident and these problems directly relate to sharpness defects, its strongest characteristic is that it can identify whether a model is calibrated or not. As this is not adequately achieved by the other methods described above it is particularly important that models should be assessed using this metric.



### 3.3 Discussion

In this chapter the concepts of calibration and sharpness are introduced and the reasons why they are the objectives when density forecasting are explained. The PIT histogram is described as a useful means of assessing probabilistic calibration. The interpretation of the PIT histogram is also explained.

Although the NLPD has a number of problems associated with it (see Section 5.2) it is by far the most common evaluation mechanism used in the literature. This approach is particularly popular because it fits inside the maximum likelihood framework and can be evaluated very cheaply. Although the CRPS is arguably a better metric (Gneiting et al., 2003), it is an expensive metric to calculate. When a prediction is not a Gaussian distribution there is no closed form equation for the metric and the integral in the formula must be estimated numerically. The Wilson score is a very descriptive and intuitive evaluation metric, however; it can only be used for very specific problem types. Therefore, throughout this thesis the NLPD score is primarily used. The CRPS is used where suitable for evaluation of density models.

Analysis of the CRPS and NLPD equations shows that although they reward calibrated models with low scores they do not directly assess the calibration of a model. The NLPD score penalises models based on the predictive density at the target variable, this directly relates to the variance of the predicted density and therefore the sharpness. Similarly, the CRPS rewards models based on the area between the cumulative predictive density and the cumulative target density. Again, how this relates to sharpness is clearly obvious. However, calibration again is not directly assessed. For this reason, the CRPS and NLPD scores can be thought of as sharpness metrics as the assessment of calibration is relegated to a secondary position being a fortuitous side-effect of these error scores. Throughout the rest of the thesis, NLPD and CRPS will be referred to as sharpness metrics and the PIT calibration measure will be referred to as a calibration metric.

This thesis proposes that both a sharpness and calibration metric should be used when models are being optimised.

## Chapter 4

# Review of Multi-Objective Evolutionary Algorithms

### 4.1 Introduction

In Chapter 2 two density forecasting models, MDN and GARCH, were chosen as the methods of density forecasting that would be used for evaluation. Both of these models can be fully described by a vector of real-valued parameters. In this chapter, optimisation methods for these models are reviewed. Specific attention is paid to evolutionary algorithm and multi-objective algorithm approaches. In the next chapter experiments will show how sharpness and calibration are conflicting objective functions and are candidates for multi-objective search. In preparation for this, a review of multi-objective algorithms is presented in this chapter.

The chapter is laid out as follows. Optimisation methods in general are introduced in Section 4.2 and Evolutionary Algorithms are introduced in Section 4.2.1. If the reader is already familiar with single objective optimisation and/or evolutionary algorithms these sections can be skipped. Section 4.3 introduces multi-objective evolutionary algorithms and Sections 4.4 and 4.5 review evolutionary algorithm approaches used for optimised MDN and GARCH methods. Finally, Section 4.6 summarises the chapter.

### 4.2 Optimisation Methods

Typically, in supervised learning, function optimisation takes the following form. A function  $f : \mathbb{R}^m \rightarrow \mathbb{R}$  is defined over data so that the objective when optimising is to find the

minimum  $\mathbf{w} = \operatorname{argmin} E[f(\mathbf{w}; \mathbf{x})]$ , where,  $\mathbf{w}$  are parameters to the function,  $E[\cdot]$  is the function expectation and  $\mathbf{x}$  is the input data. If the function  $f$  is differentiable with respect to the parameter values the preferred approach to optimisation is to use a *gradient method*. Within the class of gradient methods there are two sub-classes, *first-order* gradient methods and *second-order* gradient methods. First-order gradient methods use the gradient information  $g(\mathbf{w}) = \frac{\partial E[f(\mathbf{w}; \mathbf{x})]}{\partial \mathbf{w}}$  to perturb the parameter vector in the steepest downhill direction.  $g(\mathbf{w})$  is a vector function of derivatives of the scalar function  $E[f(\mathbf{w}; \mathbf{x})]$  with respect to all the parameters. The advantage of this gradient descent approach is that the gradients are cheap to compute which allows for fast parameter optimisation if the problem is well-conditioned. This form of gradient descent is popular for optimisation of multi-layer perceptrons, for example.

The second-order gradient methods use the Hessian matrix,  $H(\mathbf{w}) = \frac{\partial^2 E[f(\mathbf{w}; \mathbf{x})]}{\partial \mathbf{w} \partial \mathbf{w}^T}$  to make better and bigger steps toward the function minimum. There are a number of versions of second-order optimisation that are popular. For example, Newton's method, Quasi-Newton, Levenberg-Marquardt, and Conjugate Gradient. Each of these approaches are variations on the same idea with subtle adjustments to ameliorate different problems in the underlying method. An example of a second-order method is the scaled conjugate gradient method (Moller, 1993) used for optimisation of mixture density networks.

Gradient methods are popular because in most optimisation problems they can very quickly find good parameter estimates for the function being optimised. However, there are a number of difficulties with these methods. For example, in order for optimisation to be possible, the expectation function must be differentiable. This constraint restricts the type of solution that can be found. In very complex problems gradient methods can be slow to converge or tend to converge on a local minimum. This is particularly the case with first-order gradient methods. For example, optimisation of a mixture density network is rarely successful with first-order gradient methods due to the complexity of the local error curvature (Husmeier, 1999). The algorithm tends to converge on a very poor local minimum solution. Second-order gradient methods can prevent this from happening by taking very large movements in the beginning of the search to escape from local minima. However, if the problem is ill-conditioned or non-convex then second-order gradient approaches can be very slow or will not converge to a good solution also.

An alternative to the gradient methods described above is to use a *direct optimisation method*. These gradient free optimisation techniques use only the function value to search



the parameter space. Within this group of optimisation approaches there are many distinct algorithms, for example, simulated annealing, Monte Carlo, SPSA, Tabu search, genetic algorithms, evolutionary strategies and many more. These methods share a common approach but differ in their implementation.

There are numerous advantages to direct methods, for example, there is no need to derive or compute gradients so the expectation function does not need to be differentiable. This also means that the models can solve discrete/combinatorial problems and can address even non-formalized problems. This approach can even find solutions to non-convex search spaces. Furthermore, direct methods can take multiple objectives into consideration during optimisation.

There are disadvantages to the direct approach as well. For example, there is no sense of the appropriate direction or size of step to take, these are decided at random. Another detractor is the length of time it takes to converge, optimisation with this approach is an order of magnitude slower than gradient methods. However, there are heuristics and methods that can reduce the effect of these problems, for example, convergence times can be reduced by parallelising the execution of the algorithm and heuristics exist that can adjust the direct optimisation algorithms step size with respect to proximity to the global minimum e.g. evolutionary strategies.

Despite the problems with direct methods outlined above, they are the most suitable optimisation method for the problem at hand. Firstly, they are the only optimisation technique that can optimise non-differentiable objective functions. The calibration metric,  $A^2$ , is non-differentiable so gradient methods are ruled out for optimisation on this dimension. Secondly, for the proposed solution it is necessary for the optimisation algorithm to be capable of optimising two objectives simultaneously. This can only be achieved with a direct method such as an evolutionary algorithm as multiple criteria search is not supported by gradient optimisation methods. Thirdly, optimisation of density forecasting models is a difficult problem that has a complex error surface and EAs are well suited to this type of problem. For these reasons an evolutionary algorithm is the most suitable means of multi-objective optimisation of density forecasting models. A review of single objective EAs follows.

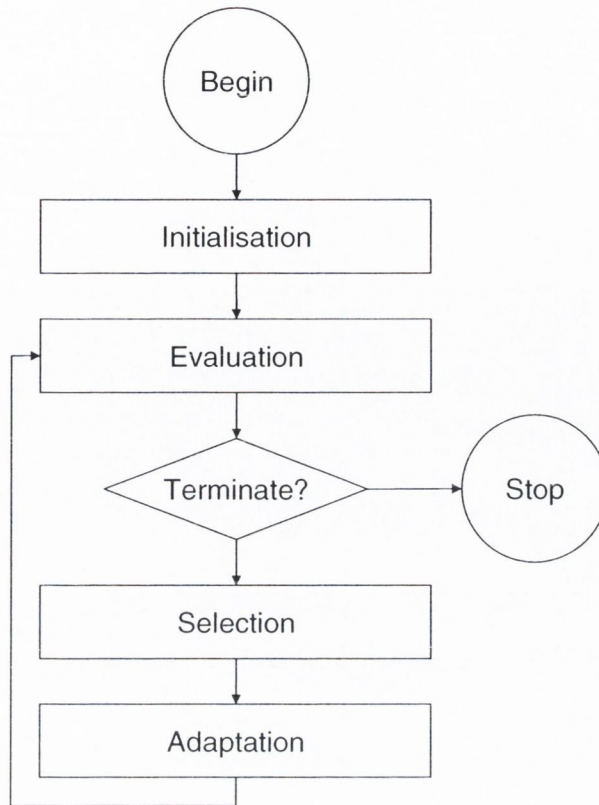
### 4.2.1 Evolutionary Algorithms

An Evolutionary Algorithm (EA) is a generic term used to indicate any population-based direct optimisation algorithm that uses mechanisms inspired by biological evolution, such as reproduction, mutation and recombination. EAs are inspired by the Darwinian principle of natural selection (survival of the fittest). An objective function is defined to describe the goals of the search problem and candidate solutions are assigned a fitness value through evaluation of their performance on that objective function. The objective function describes a fitness landscape that can be characterised in terms of the different parameters of all possible candidate solutions. Within this landscape we are seeking an optimum solution or multiple optimal solutions in terms of the parameters. EAs rely on stochastic variations in the variables to traverse the fitness landscape. To guide this search the fittest solutions are selected to survive and produce offspring. Through this process the population progressively gets fitter solutions.

There are two different evolutionary algorithm types that are generally used, Genetic Algorithms (GA) (Holland, 1975) and Evolutionary Strategies (ES) (Rechenberg, 1994). Although the basic conceptual framework of all EAs is very similar, their particular implementations differ. For example, the relative importance of mutation and crossover, the selection mechanism used and the representations of individuals vary depending on the specific algorithm. In order to understand and choose an EA for multi-objective search it is necessary to first understand how single objective EAs work. Therefore, before describing the various approaches to multi-objective optimisation this section provides a brief review of single objective EAs followed by a review of multi-objective evolutionary algorithms.

Figure 4.1 shows the basic structure of most evolutionary algorithms. The four steps to any EA are initialisation, selection, adaptation, and evaluation. A brief description of each follows:

1. *Initialisation* - Typically, initial values for each variable of the vector are randomly generated; however, domain specific knowledge can also be used to bias the search.
2. *Selection* - Selection performs two important tasks, it decides which members to remove from the population and which individuals will be used to generate offspring. Ideally, selection should identify good individuals and eliminate bad individuals. Generally, selection is based on the fitness of a solution; the better the fitness of an individual the higher the probability of that individual being selected. There are a



**Figure 4.1:** Shown is a flow diagram of a typical Evolutionary Algorithm. A population of candidate solutions (individuals) is initialised and then evolved from generation to generation by repeated applications of evaluation, selection, adaptation.

number of different selection methods, for example, tournament, roulette-wheel and ranking selection. For an account of the most popular methods see (Back, 1996).

3. *Adaptation* - The adaptation phase of an EA is the point where the selected individuals are adapted to create new individuals. The aim is for the new individual produced to have superior fitness to its parents. While this is the case, the algorithm is moving toward the optimum individual. Depending on the particular EA there are different adaptation methods. The most common methods are mutation and crossover.
4. *Evaluation* - The evaluation step of an EA is where each solution is evaluated with respect to the objective function.

The following sections describe how these steps vary for the two main types of evolutionary algorithm.



4.2.2 Genetic Algorithms

Genetic Algorithms, as their name suggests, are motivated by the principles guiding the science of genes. They simulate the fundamental ideas around natural genetics to find robust solutions in complex domains. Originally developed by Holland (1975), a vast literature has developed around GAs. This is nicely summarised in an excellent book on the theory, operation and application of GAs (Goldberg, 1989). In this section, a brief description of the most novel aspects of the GA are introduced.

A GA represents a candidate solution as a string of symbols called a chromosome (or genome). Generally, chromosomes are binary encodings of the parameters of a candidate solution. This representation enables crossover during the adaptation step of the algorithm. Crossover is a powerful and intuitive method of creating fit offspring from the population. There are two steps to crossover; first, each of the chromosomes chosen at the selection step are coupled at random for mating; second, each pair of chromosomes undergoes crossing over. Table 4.1 gives an example of crossover for a binary GA. The chromosomes come from the population selected for survival by the selection method. A dividing point is selected at random and used as the crossover point. The end sub-strings of the two chromosomes are swapped to create two new chromosomes known as offspring. These new chromosomes are evaluated and the selection process decides whether they will be kept or not. This is an example of the simplest form of crossover. Variations such as multi-point crossover, where more than one crossover point is selected are commonly used.

**Table 4.1:** This table shows an example of crossover for a binary GA. Two new offspring have been formed by choosing a crossover point at random and swapping the sub-strings of the chromosomes.

|              |                     |
|--------------|---------------------|
| Chromosome 1 | 10101 — 00100110110 |
| Chromosome 2 | 11011 — 11000011110 |
| Offspring 1  | 10101 — 11000011110 |
| Offspring 2  | 11011 — 00100110110 |

Mutation is another aspect of adaptation that is often used in genetic algorithms. The mutation of chromosomes in a binary GA simply involves the flipping of random bits in each chromosome. A probability of mutation is assigned before the algorithm commences and for each decision variable a uniform [0,1] random number is generated; if the random number is less than the probability of mutating, mutation occurs and that decision variable

bit is flipped from 0 to 1 or vice versa. Genetic Algorithms use mutation as a “background” operator to serve the purpose of preventing crossover getting stuck in a local optimum. Holland (1975) suggests that a mutation rate in the range of  $[0.001, 0.01]$  is sufficient to achieve this goal.

GAs have a number of different selection mechanisms available to them. Aside from simple rank selection, where only the best individuals are maintained, there are two other popular approaches to selection called Roulette-Wheel selection and Tournament selection.

**Roulette-wheel selection** is a proportionate selection method (Goldberg, 1989). The analogy with a roulette wheel is useful when trying to understand how this method works. The selection method is equivalent to dividing a roulette wheel up into  $N$  divisions, where  $N$  is the population size. The size of each division is proportional to the relative fitness of each population member. If the fitness of the  $i^{th}$  individual is  $f_i$  and the average fitness of each individual is  $f_{avg}$ , then the proportion of the roulette assigned to that individual is  $\frac{f_i}{Nf_{avg}}$ . The wheel is spun  $N$  times and the chromosome related to the division in which the roulette wheel stops is selected for mating.

**Tournament selection** is another popular form of selection that has better or equivalent convergence and computational time complexity properties than any other selection method that exists in the literature (Deb, 2001). It works as follows, each tournament consists of two or more randomly selected chromosomes. The fitness values of the chromosomes are compared and the chromosome with the best solution is inserted in the mating pool.  $N$  tournaments are played and if the selection process is systematic it is possible that each chromosome has competed in exactly two tournaments. This selection mechanism has a bias for fit individuals.

GAs are unique in the manner in which they represent the set of parameters to be searched; however, with this binary representation, crossover can be used successfully to optimise a parameter set quickly and efficiently. Any implementation of a GA requires decisions to be made on the crossover, mutation and selection methods, as well as careful encoding of the chromosome string. The theory behind the use of a binary alphabet for a GA is vast and the reader is referred to (Holland, 1975) and (Goldberg, 1989) for details. The main disadvantage of this approach is that it restricts the problems that can be tackled. For example, many problems require optimisation of real numbers; however, the encoding of real numbers in binary strings results in very long chromosomes. To address this issue in GAs there have been a number of adaptations that allow the chromosome



to include real number values. Herrera et al. (1998) provide a useful review of these methods. Unfortunately, some of the attractions of the GAs such as its intuitive mutation mechanism of bit flipping and crossover taken from genetics do not map directly to this more complex domain and instead a form of averaging real valued solutions called blending is used that is much less principled and elegant in comparison to the binary method.

### 4.2.3 Evolutionary Strategies

Evolutionary Strategies (ES) have emerged as a popular evolutionary algorithm from the engineering domain. Developed contemporaneously with genetic algorithms by Rechenberg, Bienenert and Schwefel in the sixties, the technique was originally created to solve problems such as the optimum bends in a pipe, or drag minimisation of a joint plate (Rechenberg, 1994). The ES process used in practical engineering examples was simple; a candidate solution was mutated to create a new offspring (e.g. the bends of a pipe were adjusted) and evaluated with respect to the objective function. If the new pipe formation was better than the parent pipe formation then it was kept, otherwise, it was discarded and its predecessor was retained.

In evolutionary algorithm terms this process can be described as follows: A parameter vector representation of the problem is decided upon. This may consist of any type of variable that can be mutated. An initial solution is created consisting of an  $m$ -dimensional parameter vector of, for example, real-valued numbers. Mutations are applied to each parameter variable, in the case of real-valued variables this mutation is generated from a normal distribution with zero mean and specified variance. The resulting offspring is evaluated and compared with the parent solution. If this new offspring is fitter than the parent it is used as the parent for the next iteration of the algorithm, if not, its predecessor is used (Back, 1996). This particular ES is called a  $(1+1)$ -ES, signifying that there is only one parent and one offspring. There are variations on this that introduce a population aspect. These are typically denoted as  $(\mu+\lambda)$ -ES,  $\mu$  signifies the number of parents and  $\lambda$  signifies the number of offspring produced. In this approach, the best individuals from the parents and offspring are selected for the next epoch of the algorithm. With this set up individuals can persist across multiple epochs of the algorithm. If this is a problem there is another version of ES called  $(\mu, \lambda)$ -ES where selection of only the best individuals from the offspring are used as parents for the next epoch. These population based variations can also use real value crossover as an adaptation mechanism, see Section 4.2.2.



A benefit of the ES that should not be overlooked is its ability to adapt to diverse problem domains. The ES is not confined by a single representation method for the problem in hand. It can deal with real, binary and discrete variables and even varying parameter vector lengths i.e. individuals can have different parameter types and a different number of parameters.

From the basic ES algorithm there have been numerous suggested amendments; however, the most important amendment is the use of additional strategy variables in the search process to aid convergence. The strategy variables usually take the form of a mutation strength vector. This is a vector of length  $m$  that determines the standard deviation of the zero mean Gaussian that is used for mutating each parameter. The desired behaviour of the mutation strength vector is to initially perform large mutations to quickly find a good region of the error space and then use smaller steps as the algorithm homes in on an optimum solution. Therefore, the optimum mutation strength must change dynamically. Rechenberg (1994) determined the best mutation strength for a set of simple problems and observed that the mutation strength should be inversely proportional to the number of variables and should reduce as the distance from the optimum solution decreases. Evolutionary Strategies that use the mutation strength vector are generally called *self-adaptive* evolutionary strategies.

There are two main types of self-adaptive ES that are of interest, isotropic and non-isotropic. Isotropic self-adaptive ES uses a single mutation strength variable for each parameter in the parameter vector. Therefore, an additional strategy parameter,  $\sigma$ , is maintained for each population member. Mutation then proceeds as follows:

$$\sigma^{t+1} = \sigma^t \exp(\tau_0 N(0, 1)) \quad (4.1)$$

$$x_i^{t+1} = x_i^t + \sigma^{t+1} N_i(0, 1) \quad (4.2)$$

where,  $x_i^t$  represents parameter  $i$  of the  $m$  dimensional parameter vector  $\mathbf{x}$  at iteration  $t$  of the optimisation algorithm.  $N(0, 1)$  and  $N_i(0, 1)$  are random draws from a standard normal distribution.  $\tau_0$  is the learning parameter and should be set to  $\tau_0 \propto \sqrt{\frac{1}{m}}$  as proposed by (Schwefel, 1987).

The non-isotropic self-adaptive ES is a variation on the isotropic mutation described above. In this version of self-adaptation there is an individual mutation strength variable for each parameter in the parameter vector. This approach is particularly useful when the different variables in the parameter vector contribute unequally to the objective function.

The update rule for this approach is:

$$\sigma_i^{t+1} = \sigma_i^t \exp(\tau' N(0, 1) + \tau N_i(0, 1)) \quad (4.3)$$

$$x_i^{t+1} = x_i^t + \sigma_i^{t+1} N_i(0, 1) \quad (4.4)$$

where,  $\tau' \propto \sqrt{\frac{1}{2n}}$  and  $\tau \propto \sqrt{\frac{1}{2\sqrt{n}}}$ . The initial values of the mutation strength vectors can be difficult to tune. Fortunately, variables in the parameter vector are generally bounded, when this is the case the convention is to chose  $\sigma_i^0 = (x_i^U - x_i^L)\sqrt{12}$ , where  $x_i^U$  is the upper bound on the parameter  $x_i$  and the superscript  $L$  signifies the lower bound (Back, 1996). In addition to this approach, there are more variations of self-adaptation that attempt to take advantage of the information generated during the optimisation process. For example, correlated self-adaptation uses additional strategy variables to take advantage of correlations within the different parameter vectors.

#### 4.2.4 Other Evolutionary Algorithms

It should be noted that there are two other forms of evolutionary algorithm, Evolutionary Programming and Genetic Programming. These methods are less commonly used in practice, therefore, they are only briefly introduced here to complete the survey of evolutionary algorithms.

Genetic Programming (GP) are another EA that are used to find computer programs that do a user defined task. Koza (1992) developed the GP to evolve efficient and meaningful programs for solving a task. However, in the context of the problems discussed in this thesis the GP is not applicable.

Evolutionary Programming (EP) (Fogel et al., 1966) is another form of EA that is very similar to ES, although both were developed separately. Again, like the ES, the representation used by the algorithm can be adapted to the problem and mutations are achieved through zero mean Gaussian adaptations. However, the selection mechanisms differ. EPs use a tournament selection mechanism to determine the population for the next epoch whereas the ES generally uses a very simple deterministic rule that keeps individuals based on their fitness ranking. EPs also differ in the mechanisms they use at the adaptation phase. EP is based on the theory of evolution within reproductive populations as opposed to being based on the individual behavioural level. Thus, no crossover is used in the EP algorithms as this is not something that is observed within reproductive populations. Aside from these subtle differences EPs and ESs are very similar.



Most variations on ESs can be reformulated as an EP and vice versa. For this reason EPs have not been given their own category in this section.

In the following section Multi-objective Evolutionary Algorithms are introduced.

### 4.3 Multi-objective Evolutionary Algorithms

There are many reasons why evolutionary algorithms are good candidates for optimisation of density forecasting models. For instance, they can find a global optimum in complex multi-modal search spaces and they can optimise on non-differentiable objective functions. In this section, the multi-objective capabilities of evolutionary algorithms are explained.

The origins of multi-objective optimisation (MOO) are embedded in economics and can be attributed to Francis Y. Edgeworth (1845-1926) and Vilfredo Pareto (1848 - 1923). Edgeworth (1881) was first to determine the optimum of a multi objective problem. He does so, in terms of two objectives  $P$  and  $\pi$ , as follows: *“It is required to find a point  $(x, y)$  such that in whatever direction we take an infinitely small step,  $P$  and  $\pi$  do not increase together but that, while one increases, the other decreases”*. Pareto (1896), used a slightly different description for the same notion: *“The optimum allocation of the resources of a society is not attained so long as it is possible to make at least one individual better off in his own estimation while keeping others as well off as before in their own estimation”*, although the result is the same<sup>1</sup>.

There are innumerable circumstances in different domains where multiple objective problems exist. For example, in finance the primary objective is to maximise returns while minimising risk, or in engineering there is a desire to maximise performance while minimising cost. The distinction between single objective and multi-objective problems is that there are two or more *conflicting* objectives. Conflicting, as Edgeworth described above, means that improving in one will cause deterioration in the others. Later, in Section 5.4 the conflicting nature of sharpness and calibration is shown.

A generic multi-objective optimisation problem can be formulated as follows. A vector function  $f$  that maps an individual, i.e. a vector of  $m$  parameter variables,  $\mathbf{x}$ , that reside in the parameter space  $X$ , to a vector of  $k$  objectives,  $\mathbf{y}$ , from the objective space  $Y$ .

---

<sup>1</sup>Translations and extracts taken from (De Wick, 2005)



$$\begin{aligned}
\min/\max \quad \mathbf{y} &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \\
\text{subject to } \mathbf{x} &= (x_1, x_2, \dots, x_m) \in X \\
\mathbf{y} &= (y_1, y_2, \dots, y_k) \in Y
\end{aligned} \tag{4.5}$$

This is the general setup of any multi-objective optimisation problem, the solution to this problem consists of all the parameter vectors whose objective function vectors do not improve in any dimension without a subsequent deterioration in another dimension (Zitzler & Thiele, 1999). The set of all these solutions is known as the Pareto optimal set. This concept can be expressed mathematically as follows: Assume, without loss of generality, a minimisation problem and consider two parameter vectors  $\mathbf{a}, \mathbf{b} \in \Omega$ . Then,  $\mathbf{a}$  is said to dominate  $\mathbf{b}$  (also written as  $\mathbf{a} \succ \mathbf{b}$ ) iff:

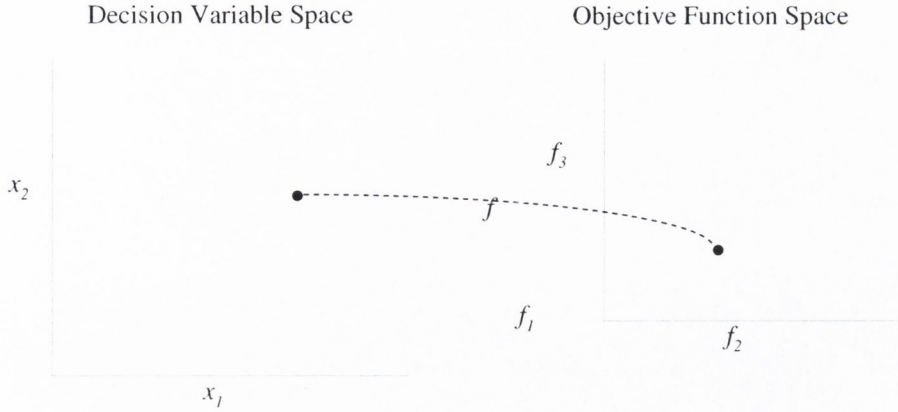
$$\begin{aligned}
&\forall i \in \{1, \dots, k\} : f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \quad \wedge \\
&\exists j \in \{1, \dots, k\} : f_j(\mathbf{a}) < f_j(\mathbf{b})
\end{aligned}$$

where,  $f_i(\mathbf{x})$ , returns the fitness for parameter vector  $\mathbf{x}$  on the  $i^{th}$  objective function. All parameter vectors which are not dominated by any other parameter vector of a given set are called *non-dominated* with respect to the set. The parameter vectors that are non-dominated within the entire search space are denoted as *Pareto optimal* and constitute the so called Pareto-optimal set or Pareto-optimal front (Zitzler & Thiele, 1999). The goal of multi-objective optimisation is to find a diverse Pareto-optimal set.

Figure 4.2 shows how a parameter vector with two variables ( $x_1, x_2$ ) is mapped to an objective space. In this example there are three objectives being evaluated resulting in a 3-dimensional objective space ( $f_1, f_2, f_3$ ). When there are three or less objectives it is possible to visually assess the quality of each individual model in terms of the objectives by plotting them in a scatter plot.

### 4.3.1 A priori method

The classic approach to multi-objective optimisation is the *a priori* method. In its simplest form this approach applies a linear combination to the multiple objectives to create a single composite objective before optimisation commences, hence the name *a priori* method. Creating a composite objective function allows single objective optimisation methods to



**Figure 4.2:** On the left is a single decision/parameter vector (dot) in decision/parameter variable space. The right plot shows the position of this decision vector in objective space.

be used. This approach can be formulated as follows:

$$\min/\max : f^* = \sum_{i=1}^k \alpha_i f_i(\mathbf{x}) \quad (4.6)$$

where,  $\alpha_i$  is a positive weight applied to each objective function and  $\sum_{i=1}^k \alpha_i = 1$ . There are a number of variations on this approach, for example, the multiplicative technique where objectives are combined through multiplication (Horn & Nafpliotis, 1993), or the target vector technique that can be thought of as determining the distance to the target from all objectives as the fitness metric (Whitely, 1996). For a comprehensive review of other *a priori* methods see (Coello, 1998).

The difficulty with this method is that, as shown above, unlike in single objective optimisation, true multi objective optimisation problems never have a single optimum solution. Instead, because the objective functions are conflicting, there exists a set of optima that relate to different trade-offs in objective function values. The *a priori* approach will return a maximum of one of these optima. Therefore, depending on the objective weights selected, a different, sometimes very different, solution will result. Determining the objective weights prior to optimisation requires a combination of qualitative and quantitative analysis and is rarely a trivial problem. One solution to this particular problem is to use various settings of the weights to find a set of diverse solutions. Although this seems like a sensible approach it also has its drawbacks. First, there is the practical problem of selecting different suitable weights, equal spacing of weights rarely results in an equally spaced Pareto front. This issue is more evident when the number of objectives is large. The second, more serious issue, is that if the set of optimal solutions is non-convex, it is

not possible for this approach to find points from all parts of the Pareto set. Das and Dennis (1997) provide a geometrical argument as to why this is the case.

In summary, the *a priori* approach to multi-objective optimisation is a simple method of finding solutions from the Pareto optimal set and easy to implement. However, a good, evenly spread out, and diverse set of trade-off solutions is difficult to find using this method. Furthermore, in certain scenarios it is impossible to obtain Pareto optimal solutions with this approach.

#### 4.3.2 A posteriori method

The *a posteriori* method to multi-objective optimisation is a process that attempts to find the true Pareto optimal set of solutions. No prior commitment is made about the trade-off between objective functions. This approach does not require objective function weights, therefore, it can be considered a true multi-objective search process rather than just a single objective search over an aggregated objective function. The optimisation process returns a Pareto set of individuals from which the modeller can select a trade-off solution that minimises their loss.

The most common approach to *a posteriori* optimisation is known as *Pareto Sampling*. This approach maintains an archive of non-dominated individuals throughout the search process. Periodically this archive is updated with new models that are non-dominated and individuals that have become dominated are removed. The implementation of this type of multi-objective optimisation method is straightforward within an evolutionary algorithm optimisation process. A flow diagram of the basic algorithm is depicted in Figure 4.3. The diagram is very similar to the standard EA shown in Figure 4.1. However, the details of each step in the flow diagram are quite different in the multi-objective setting. A description of each of these steps follows.

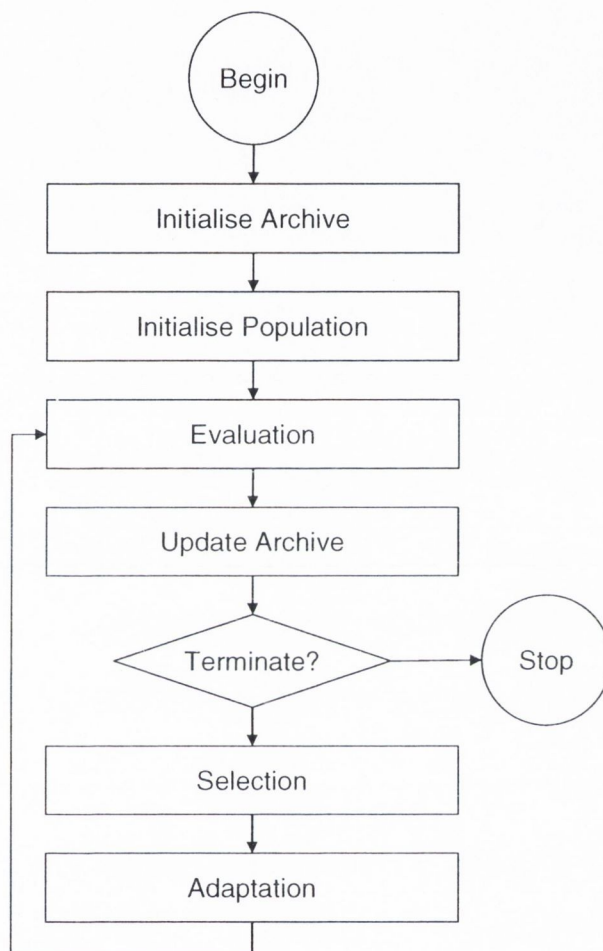
##### Initialise Archive

The archive of an MOEA is a store of non-dominated individuals. This is just a placeholder for the non-dominated individuals found during the search process.

##### Initialise Population

Prior to this algorithm commencing, a representation of the model to be optimised that is compatible with the chosen EA must be selected. In the case of evolutionary pro-





**Figure 4.3:** Shown is a flow diagram of a typical Multi-objective Evolutionary Algorithm. Like the standard EA a population of candidate solutions (individuals) is initialised and then evolved from generation to generation by repeated applications of evaluation, selection, and adaptation. The main distinction between this and Figure 4.1, aside from multiple objectives, is that a separate archive of non-dominated solutions is initialised and maintained throughout optimisation.

gramming and evolutionary strategies this is usually straightforward, however, if the base evolutionary algorithm is a genetic algorithm then it is typically the case that suitable representation of the problem must be decided upon. This initialisation step then generates parameter vectors with the specified structure using a random number generator.

### Evaluation

The evaluation step of a multi-objective evolutionary algorithm is the point where the population of individuals are evaluated over the two or more objective functions. Each individual will then have an associated vector of fitness values.

### Update Archive

After evaluation the non-dominated individuals of the current population are inserted in the archive if they are not dominated by any individuals in the archive. Any archive individuals that are dominated by the new individuals inserted are removed. As the archive size grows this process becomes increasingly more and more time consuming. There are a number of possible algorithms that can be used to improve the time taken to update the archive. For example, Deb (2001) suggests two possible update procedures other than a naive approach of comparing all models for dominance. Zitzler and Thiele (1999) suggest clustering solutions into a predefined number of individuals periodically during the optimisation process. Then only one member of each cluster is kept and the other solutions are discarded. Fieldsend et al. (2003) suggest changing the structure of the way the archive is stored from being a list of fitnesses values to being a dynamic tree that allows dominance checks to be completed quickly. However, the archive update algorithm that is used is dependent on the modeller's sensitivity to optimisation time and the number of iterations generally required to produce a sufficiently diverse set of models.

### Selection

The selection process for multi-objective search is quite different to the selection mechanisms that are used in a single objective EA. The objective of an MOEA is to return a diverse set of Pareto optimal solutions. One commonly used solution is to select a member uniformly at random from the archive set. However, this approach artificially concentrates the search on densely populated regions of the front. It would be better to select disparate individuals. To achieve this there are two commonly used algorithms. First, the selection

method introduced by Zitzler and Thiele (1999) for their Strength Pareto Evolutionary Algorithm. This uses a clustering algorithm to create different clusters along the Pareto front and maintains a diverse set of individuals allowing uniform random selection. However, this also leads to a constrained archive where possibly useful parents are systematically discarded.

Instead, the Partition Quasi Random Selection method or PQRS is used in this thesis (Fieldsend et al., 2003). The PQRS works as follows, one of the objective functions is selected and the values of that objective function are separated into  $(N - 1)$  bins, where  $N$  is set as some minimum number of bins or the number of individuals that need to be selected, whichever, is larger. Individuals are then selected at random from the bins. At the next iteration of the loop a different objective function is used for determining the individuals selected in a round robin style. By equally spacing the bins the problem of bias towards clusters of similar individuals is avoided. Fieldsend et al. (2003) also recommend using pinning i.e. giving the most extreme individual on the objective function dimension a bin to itself, as the goal of multi-objective search is to return a diverse set of individuals and pinning encourages diversity.

## Adaptation

Typically, the adaptation step of an MOEA is the same as the methods used in the single objective EA.

This multi-objective evolutionary algorithm above is an introduction to multi-objective optimisation that provides the basis of the proposed optimisation method that is the focus of this thesis. Implementation details differ depending on the model being optimised and the objective functions used, however, the steps described above generally remain the same.

## 4.4 Evolutionary Neural Networks

In the next chapters a multi-objective evolutionary mixture density network will be introduced. As the mixture density network is a sub class of neural networks, a brief review of evolutionary techniques for optimisation of neural networks is provided here. Yao (1999) provides a comprehensive review of single objective evolutionary algorithm optimisation techniques for neural networks. This class of network is called an Evolutionary Artificial



Neural Network (EANN). EANNs can optimise connection weights, network architectures and even network learning rules. One of the most popular approaches to EANN optimisation is to take a hybrid approach, combining the speed of a gradient method with the flexibility of an evolutionary algorithm. The EPNet (Yao & Liu, 1998), optimises based on gradient information, however, periodically, the search is stopped and the network weights and architecture are evolved to find the optimum network design.

The first multi-objective neural networks used simple EANNs with the *a priori* approach, for example (Wang & Wahl, 1997). Abbass (2001) successfully developed a multi-objective optimisation approach for neural networks that predicted whether breast cancer was present or absent in mammograms. This was the first multi-objective neural network that used an *a posteriori* method and produced a Pareto set of solutions. Subsequently, other multi-objective neural networks were developed by (Fieldsend & Singh, 2002; Gonzalez et al., 2003). Finally, a general framework for multi-objective optimisation of neural networks was suggested by Fieldsend and Singh (2005). Although much work has been carried out on general evolutionary neural networks there has been no prior work in the field of evolutionary mixture density networks.

## 4.5 Evolutionary GARCH

Goffe et al. (1994) successfully apply simulated annealing to a number of different econometric models and show how the robust nature of an evolutionary algorithm means that it is less likely to fail and more likely to find the global optimum than conventional algorithms. Jerrell and Campione (2001) develop this work by comparing different evolutionary algorithms using a number of different econometric models including GARCH. They concluded that both simulated annealing and evolutionary strategy optimisation algorithms found the best results in general. However, the evolutionary strategy uses less processor time.

## 4.6 Discussion and Summary

In Section 2.2.1 the relationship between maximum likelihood and density forecasting was developed. This approach to density forecasting provides an already developed theoretical foundation for the estimators, making numerical estimation straightforward using standard gradient methods. Unfortunately, this will not necessarily return the best set of solutions

as this approach can be sensitive to anomalies and may fail. For example, Cramer (1986) lists a number of ‘unpleasant possibilities’: the algorithm may not converge in a reasonable number of steps, parameter values may tend toward infinity, or the algorithm may continuously loop through the same points. Also, the algorithm may have difficulties with ridges and plateaus. When faced with such difficulties, the researcher is often reduced to trying different starting values (Cramer, 1986; Finch et al., 1989). In fact, even if the algorithm converges, there is no assurance that it will converge to a global, rather than local, optimum since conventional algorithms cannot distinguish between the two (Goffe et al., 1994). Finally, the maximum likelihood framework does not provide a means of optimising on more than one objective.

Evolutionary Algorithms have the potential to solve these problems. EAs explore the function’s entire surface and try to optimise the function while taking both large and small steps. Thus, they are largely independent of the starting values, often a critical input in conventional algorithms. Further, EAs can escape local optima and go on to find the global optimum through their varied step size. Less stringent assumptions are made regarding the function than a conventional algorithm e.g. it need not be continuous, this allows the algorithm to more easily deal with ridges and plateaus (Goffe et al., 1994). Also, it can optimise functions that are not defined for some parameter values and optimise on multiple objectives simultaneously.

Although numerous EAs exist the two main methods used are GA and ES. The question is which approach is most suitable for density forecast optimisation. In the experiments that follow we use evolutionary strategies as opposed to genetic algorithms. The primary reasons for choosing ESs over GAs are:

- The vector representation of the parameters of a density forecast model do not have to be adapted or transformed.
- Values of the parameter vectors can be optimised at different rates.
- ESs have already been proven to optimise density forecast models effectively (Jerrell & Campione, 2001).
- Adaptation from single to multi-objective optimisation is possible.
- ES performs well when the search problem is complex.

In this chapter, different methods of optimisation were reviewed and discussed. Within the problem space of density forecasting, evolutionary strategies emerged as the most suitable optimisation algorithm. This chapter provides the theoretical basis behind the algorithms that are described in the next chapter.



## Chapter 5

# Theory and Contributions

### 5.1 Introduction

The central message in this thesis is that the calibration of a density forecasting model is as important as the sharpness of a model. In this section, the most common density forecast quality metrics, Continuous Rank Probability Score (CRPS) and Negative Log Predictive Density (NLPD), are analysed and their weaknesses in terms of calibration are highlighted. Particular attention is paid to the NLPD because it is the most common optimisation error score (Engle, 1982; Bishop, 1994).

Optimising a model using NLPD is an attempt to find the set of model parameters that maximise the probability density at the observation. This is a sensible approach to conditional density function estimation when the data generating function is well understood and the assumptions underlying the predictive model can be justified. Unfortunately, this is not always the case. For many prediction tasks the data generating function is unknown and the underlying conditional density functions are complex and irregular. In these circumstances, more care needs to be taken at the model design and optimisation stage to make sure that the predictions from the resulting models are useful to the prediction user. Other common modelling problems, such as over-fitting, can also occur. In the following section some of the more common weaknesses experienced when using density forecasting models are highlighted.

Once the main problems with NLPD are recognised, a new method of ranking calibration models that uses the Anderson-Darling score is introduced (Anderson & Darling, 1954). Experiments show that it is an accurate estimator of calibration performance and

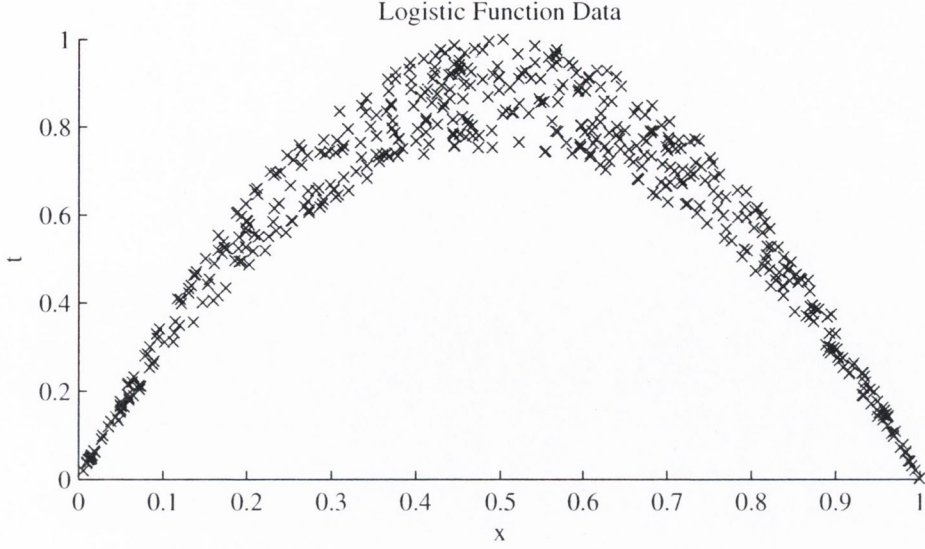
robust even with small amounts of data. This provides the missing link between optimisation using the NLPD score that biases sharpness and optimising on both calibration and sharpness simultaneously. However, before proceeding on this path, it is necessary to confirm that sharpness and calibration are conflicting objectives. In Section 5.4 it is demonstrated that this is the case. Finally, a framework is proposed that combines these findings with the multi-objective evolutionary algorithms reviewed in the previous chapter to optimise density forecasting models on both sharpness and calibration.

## 5.2 Common Density Forecast Optimisation Problems

Density forecast modelling has the ambitious goal of producing a probability density function estimate of the true conditional density function. Naturally, there are a number of challenges that are still outstanding in the density forecast domain that have not yet been addressed. In this section, three common problems when optimising density forecasts are identified and explained. As mentioned in Chapter 3 the popular metrics for assessing probability density estimates, CRPS and NLPD, are biased toward models with good sharpness rather than good calibration. Section 5.2.1 illustrates this argument. In Section 5.2.2 the propensity of density forecast models to over-fit is shown and the consequences are demonstrated. Finally, the NLPD score’s sensitivity to outliers, a common issue identified in the density forecasting literature, is also demonstrated. Synthetic data sets are used in the following experiments to clearly illustrate the problems. However, in Chapters 7 and 8 real data is used for evaluation of the proposed technique and the same problems can be seen arising.

### 5.2.1 Error Function Sharpness

The objective of the NLPD score when used within the maximum likelihood framework is to attempt to find the set of model parameters that maximise the probability density at the observation. Similarly, the CRPS score attempts to minimise the difference between the predicted cumulative density and the cumulative density of the observation. Unlike the Probability Integral Transform (PIT) score that directly assesses the empirical frequency of the probabilistic predictions to evaluate the calibration of the model, neither the CRPS nor NLPD score directly assess this objective. The effect of this can be illustrated by analysing the error scores of a density forecasting model during training.



**Figure 5.1:** Sample of logistic function data.

In this experiment the logistic function data set is used. The logistic function is a simple arc shaped function with heteroskedastic variance. It is based on a simple synthetic data set described by Husmeier (1999). Data points are generated by the function:

$$t_i = \alpha_i x_i (1 - x_i) \quad (5.1)$$

where,

$$P(\alpha_i) = \begin{cases} 1 & \text{if } \alpha_i \in [3, 4] \\ 0 & \text{otherwise} \end{cases} \quad (5.2)$$

The characteristics of this data set make it particularly interesting for experimenting with density forecasting models. The variance is largest for  $x_i = 0.5$ , and then decreases monotonically to zero as  $x_i \mapsto 0, 1$ . Husmeier (1999) shows that the true conditional variance at any point can be determined by:

$$\sigma_i = \frac{1}{\sqrt{12}} x_i (1 - x_i) \quad (5.3)$$

Figure 5.1 is a sample training set generated from the logistic function, where  $x$  values are random draws from a Uniform distribution.

A three hidden unit and one Gaussian output Mixture Density Network (MDN) model is trained for 1000 epochs on 200 randomly drawn instances of the logistic function data set. The Scaled Conjugate Gradient algorithm with the NLPD score is used for training of the model. However, at each epoch four error measures were recorded. First, the

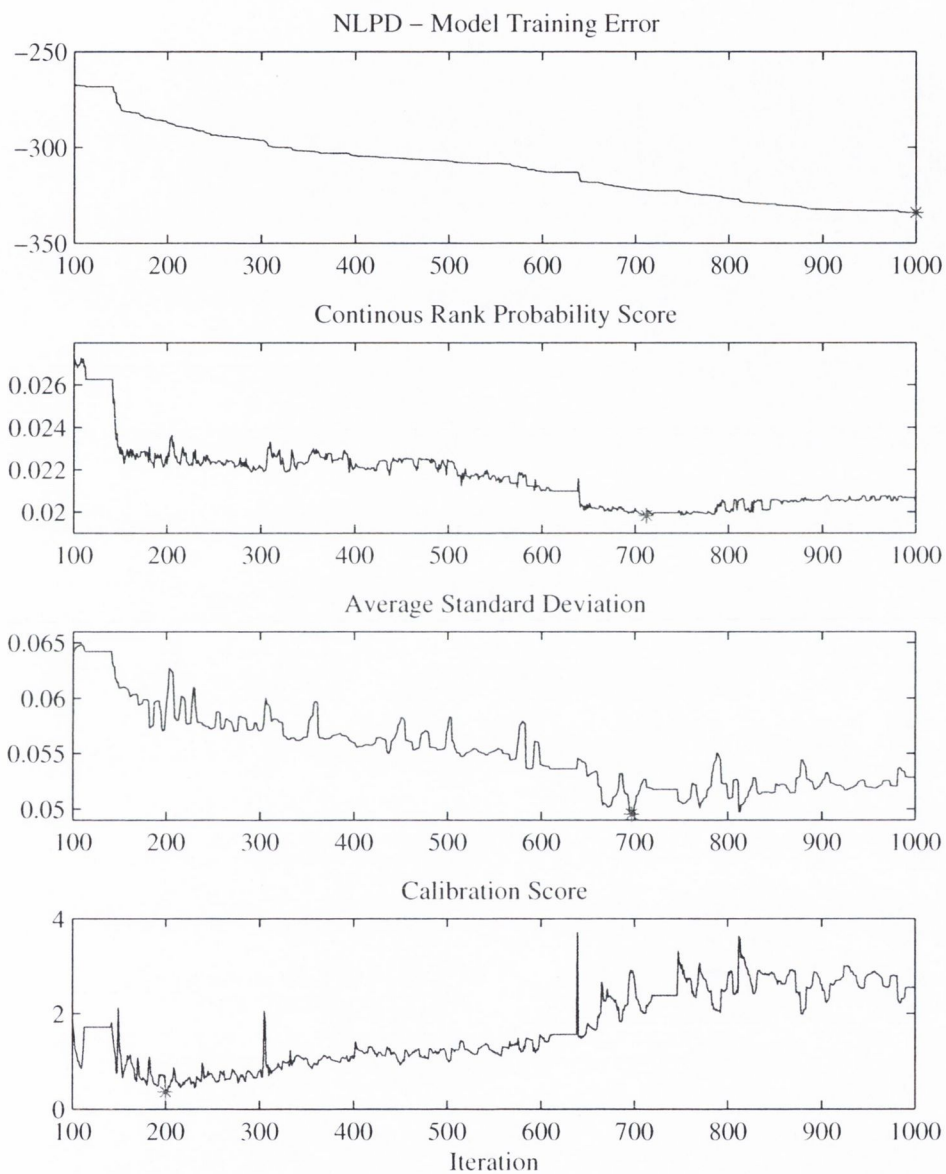


NLPD was calculated. Second, the CRPS score was calculated. Fortunately, because the predicted density was Gaussian, this could be done in reasonable time. Third, as sharpness refers to the concentration of density forecasts and because this is a very simple function to estimate, the average standard deviation of the predicted densities could be calculated as a true measure of sharpness. Finally, a new calibration score based on the PIT and described in detail in Section 5.3 is used to directly assess calibration. Figure 5.2 shows the resulting error plots.

The first 100 iterations are not included in Figure 5.2 because during this period of the search process the model error scores are an order of magnitude higher than the last 900 iterations of the algorithm and due to the restriction in vertical height on the page a comparison of methods would be difficult. The top plot of Figure 5.2 shows the actual error progression of the MDN model during training. It is a continuously decreasing plot with the minimum value occurring at the end of the series. The second plot evaluates the same models in terms of CRPS. A similar plot is shown; however, after approximately 700 iterations the error values begin to increase. The third plot represents the actual sharpness of each model during training, again a similar decreasing plot is shown; however, as identified by the CRPS score, the actual sharpness begins to increase at around 700 iterations, but the increase is marginal. Finally, the fourth plot shows the calibration score for each model. This tells a very different story. The minimum calibration score is at approximately the 200<sup>th</sup> iteration with calibration deteriorating rapidly after that point.

These results confirm the suggestion made previously that CRPS and NLPD are mainly sharpness error functions. The NLPD error function has clearly taken a substantial decrease in the calibration of the model being optimised in favour of improving the sharpness of the model. The CRPS score suggests stopping earlier, however, at this point much of the deterioration in calibration has already occurred. As Gneiting and Raftery (2004) suggest, the CRPS score seems to be slightly more robust than the NLPD score in terms of its sharpness evaluation as its minimum point is much closer to the actual sharpness minimum.

There are two other interesting results from this experiment. First, there is evidence to suggest that calibration and sharpness are not complimentary characteristics, in fact the model parameters conflict near the global optimum in one or other objective function dimension. This proposition is confirmed in Section 5.4. Second, there is high volatility



**Figure 5.2:** Four plots are shown. The top plot shows the actual error progression in terms of NLPD of the model during training. The minimum point on each plot is marked with an \*.

of the calibration score as the model training progresses. This suggests that calibration is very sensitive to model parameter values. Therefore, taking care at the model training phase should lead to large improvements in calibration while only marginally negatively effecting the model's NLPD score.

### 5.2.2 Over-fitting

Over-fitting is a common problem amongst point estimators. This generally occurs when the model used to describe the data is too complex. A sufficiently flexible model may fit the training data perfectly but may produce extremely poor forecasts on test data sets. The difficulty of the task of predicting a conditional probability density function rather than a single point, results in density forecasting models being typically more complex than equivalent point prediction models. Due to this added complexity they are generally in violation of Ockham's razor<sup>1</sup> and so are prone to over-fitting. To make matters worse, a density forecasting model can over-fit on both mean and variance dimensions.

Over-fitting on the conditional mean is hard to diagnose. Similar to point estimation problems this problem is generally not evident until the model is tested on out-of-sample data. Over-fitting of the conditional variance, however, can often be identified by plotting a PIT histogram of the predictions of the training data.

Figure 5.3 shows the PIT histogram for the model trained in the previous section on the logistic function data set. A further 500 randomly drawn instances of the data set are generated for testing. A distinct "U" shaped histogram is seen in the training data. As explained earlier a "U" shaped histogram signifies over-confident predictions and under-dispersion of variance estimates. This error carries over to the test data where a similar malformed histogram occurs.

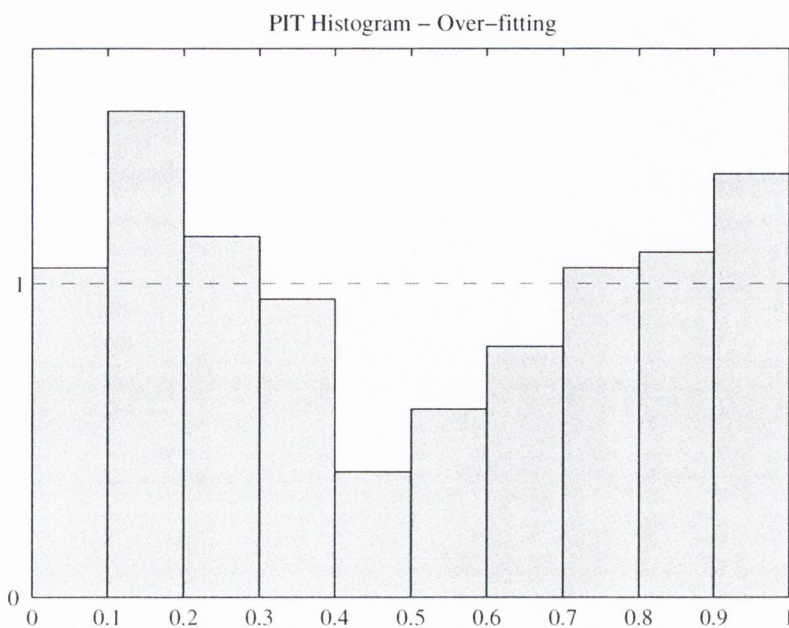
### 5.2.3 NLPD Sensitivity to Outliers

The NLPD error score penalises inaccuracies on an exponential scale. This results in the score being sensitive to rare events. This problem has been cited by many practitioners of the NLPD. For example, Weigend and Shi (2000) state that extreme outliers can effect the results of models and suggest using a trimmed mean of NLPD values when evaluating them. This solution is less than optimal as it simply chooses to discard the effect of the

---

<sup>1</sup>Ockham's razor, 'entities should not be multiplied beyond necessity' i.e. one should strive for parsimony and simplicity in scientific theories.





**Figure 5.3:** Shown is the test data PIT histogram for an MDN that has over-fit the training data.

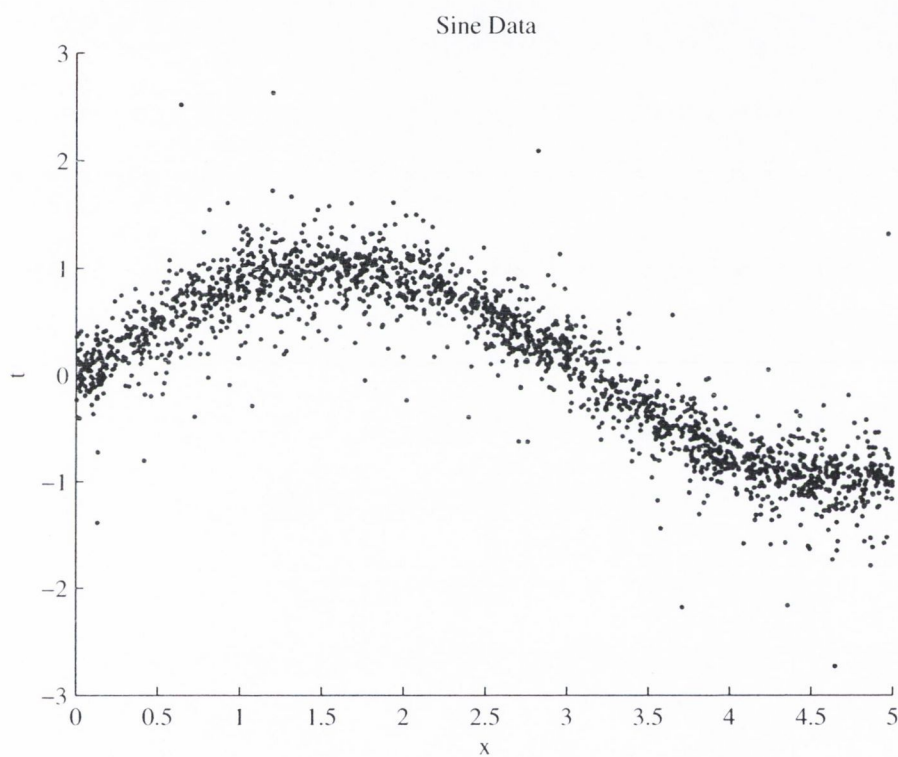
least frequent occurrences. In most scenarios, these occurrences are very important to the prediction user as they indicate the bounds of the risks they are exposed to so this solution is unsatisfactory. Gneiting and Raftery (2004) and Candela et al. (2006) attribute this sensitivity to the exponential nature of the NLPD score. As Figure 3.1 shows, the contribution to error of observations on the extreme tails of the predicted distribution is considerable in comparison to errors in higher density regions of the predicted density. The effect of this bias in the error function is that the NLPD often produces poor models in data sets that contain more observations of extreme outliers than would be expected by the distribution assumed by the prediction model. This may occur because the selected sample of data for training has coincidentally more occurrences of rare events than is typically seen in that problem domain or because the underlying data generating distribution is more leptokurtic than the density forecasting model permits its predicted density function to be.

It is possible to demonstrate this problem by contriving a data set that contains more outliers than would be expected by the prediction distribution. To achieve this, a simple synthetic data set based on the sine function with added noise drawn from either a Normal

distribution with zero mean and a standard deviation of 0.2 or a Normal distribution with 0 mean and standard deviation of 3, where the probability of noise being generated from the first distribution is 99% and the probability of noise coming from the distribution with broad variance is 1%. Inputs for the data are uniformly drawn from the interval [0,5]. The target values are generated according to,

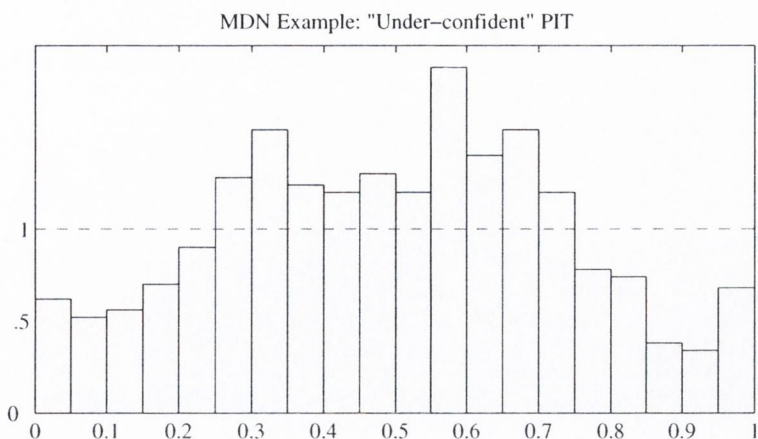
$$t(x) = \sin(x) + \epsilon \tag{5.4}$$

where,  $\epsilon$ , is the noise. Figure 5.4 depicts the data set used. A Mixture Density Network



**Figure 5.4:** Sine wave data with outliers.

(MDN) with 5 hidden units and 1 Gaussian output was trained for 2,000 epochs on 1,000 input/output pairs of the sine wave data. Figure 5.5 depicts the PIT histogram from the resulting model. The histogram shows that the model has produced a set of forecasts where the observations occur too often in the area around the first standard deviation of the predicted distribution. The extreme outliers cause the variance estimates to be higher resulting in the model having very conservative estimates. This is called an under-confident model i.e. it has too broad a variance around the conditional mean.



**Figure 5.5:** Sample PIT histogram from under confident MDN.

In many circumstances outliers signify an error in the readings, an exogenous event temporarily upsetting the generating distribution, or it can be the case that the generating distribution has fat tails, i.e. is leptokurtic, as is common in financial data. The effect of using these poorly calibrated predictions is demonstrated in Chapter 7 within a risk management application. The CRPS score is not analysed here, however, in Chapter 7 the CRPS is shown to perform worse than the NLPD score when the data has leptokurtosis.

The effect of outliers demonstrated above in the context of MDN models is also evident in GARCH models. In practice, it is observed that the standardized errors of a GARCH model will be leptokurtic, albeit less than in the underlying financial data e.g. see (Bollerslev, 1987; Baillie & Bollerslev, 1989; Hsieh, 1989; Park, 2002). Franses and Ghijsels (1999) suggest that this is sufficient evidence to conclude that *aberrant observations* exist in the returns that are not captured by the general GARCH models. They go on to say that neglecting to take this into account leads to biased parameter estimates for the conditional density functions. This problem persists in the out-of-sample forecasts. Ledolter (1989) concurs with this suggestion and empirical examples in Jorion (1995); Andersen and Bollerslev (1988) show how outliers detrimentally affect out-of-sample performance when optimised using negative log-likelihood objective functions.

#### 5.2.4 Discussion

The three examples above show how sharpness metrics are poor error functions when used for estimating density forecasting models in certain common scenarios. Other problems



can also occur, for example, it is often the case that the data generating function in prediction tasks has a skewed conditional distribution. In the area of weather prediction many of the target variables have a minimum value of zero e.g. rainfall prediction has minimum value of 0 millimetres and therefore, rain prediction data is generally skewed in the direction of zero. Or stock prices have a tendency to increase over time and so there is a skew in the positive direction. These important aspects of the data generating process are neglected by this technique of optimisation. The optimisation method proposed in this thesis attempts to address these problems.

### 5.3 A New Calibration Score

In Section 3.2 the probability integral transform was introduced. Diebold et al. (1998b) suggest plotting a histogram of the PIT values and comparing this to a perfect  $U(0,1)$  distribution as a method of discerning the degree of calibration of a model. A test for calibration relates directly to a test for whether the PIT values for the predictions follow a uniform distribution. Therefore, a number of possible goodness-of-fit tests can be carried out to determine calibration. Diebold et al. (1998b) argue that using the PIT values in this way is uninformative because, *“Such tests [...] are not likely to be of much value in practical applications, because they are not constructive; that is, when rejection occurs, the tests generally provide no guidance as to why.”* He carries on by suggesting that a useful method of discerning the calibration of a model is to plot a histogram of the PIT values and visually assess the histogram’s uniformity, or alternatively, to plot the PIT values as an empirical stepwise distribution and compare them to the cumulative uniform distribution. Diebold’s argument makes sense in situations when the calibration of a single or a small number of models in isolation is being assessed. However, it is often the case that there are a large number of models that require ranking, for example in the case of model selection. Furthermore, drawing histograms also creates a problem when there are insufficient available test points to create a meaningful plot. In these scenarios the reason *“why”* the model is not calibrated is unimportant. Fortunately, there are a number of pieces of literature that suggest different goodness-of-fit tests for assessing the uniformity of PIT values.

Crnkovic and Drachman (1997) suggest using the Kuiper statistic for measuring uniformity. However, this is not a robust test for uniformity and requires a very large number of

data points before it is consistent in its estimates. Berkowitz (2001) suggests a Likelihood Ratio test for evaluating PIT values and developed a rigorous framework for evaluation of different aspects of the density forecast, even when evaluation data is sparse. Specifically, he proposes that the PIT values be transformed into a normally distributed series,  $N(0,1)$ , via the inverse normal cumulative density function transform. This is proposed because tests for normality are more powerful than tests for uniformity. Using this transformed series he suggests a one-degree-of-freedom test of independence against a first-order autoregressive structure and a three-degree-of-freedom test of zero mean, unit variance and independence. This approach is arguably the most well developed and principled method of evaluation of PIT values to date; however, it is also the most time consuming and computationally intensive. Wallis (2003) suggests using an adaptation of Pearson's chi-squared goodness-of-fit test for density forecasting evaluation. His suggested adaptation provides a means of extracting information from the PIT values that can diagnose more precisely where the predictions fall down e.g. location, scale or skewness. Somewhere between the suggestions of Crnkovic and Drachman (1997), and Berkowitz and Wallis (Berkowitz, 2001; Wallis, 2003), lies the research carried out by Noceti et al. (2003). In their paper, the Kolmogorov-Smirnov, Kuiper, Cramér-von Mises, Watson and the Anderson-Darling goodness-of-fit tests were compared. Through their analysis they concluded that the Anderson-Darling test was the most robust metric for this task (Anderson & Darling, 1954). Independently, we tested the Chi-square and Shapiro-Wilks tests for uniformity. The Anderson-Darling test performed better than these two other metrics also.

The Anderson-Darling is known as a quadratic test because it is based upon a weighted square of the vertical distance between the empirical stepwise density function and target cumulative density function. It differs from the well known Kolmogorov-Smirnov test, which finds the maximum vertical distance between the empirical and target density. Similar approaches, such as the Cramer-von Mises statistic and the Watson statistic, use different weighting schemes. The Anderson-Darling is defined as follows,

$$A^2 = -m - \frac{1}{m} \sum_{j=1}^m (2j-1) [\log(z_j) + \log(1 - z_{m-j+1})] \quad (5.5)$$

Where,  $m$ , is the number of  $z$  values, and the  $z$  values are sorted in ascending order.

$A^2$  is not a proper scoring rule in the sense that it cannot assess the statistical consistency of a single prediction - it can only be determined for a set of predictions. However,



Anderson and Darling (1954) suggest that this statistic produces assessments of uniformity that are valid for sample sizes larger than 40. Therefore, this quality score should be used when the dataset size is greater than 40. Naturally, larger sample sizes give more accurate scores.

### 5.3.1 Empirical Comparison: $A^2$ and Statistical Consistency

From the value  $A^2$  (Equation 5.5) it is possible to determine rejection or acceptance of the null hypothesis, however, because there is a direct relationship between uniformity and the  $A^2$  value it is reasonable to use the equation above as an error score. To demonstrate the direct relationship between calibration and the  $A^2$  error function the following experiment was performed.

If calibration is understood to be the correct specification of a probability forecast it is possible to assess the performance of a model on specific quantiles over a large test set by determining the quantile accuracy. Intuitively, over a large set of predictions, 5% of target variables should be below the 5<sup>th</sup> percentile of the predicted densities. This value is called the *HitRate* (Engle & Manganelli, 1999).

$$HitRate = \frac{\sum_{i=1}^N 1 \{ \varphi_i^{-1}(q) > t_i \}}{M} \tag{5.6}$$

where,  $q$  is the specified percentile,  $\varphi^{-1}$  is the inverse cumulative of the predicted density and  $M$  is the size of the data set.

100 samples of 6,000 simulated observations were created using a Gaussian GARCH(1,1) process with parameters (0.3 0.05 0.9) (see Section 2.3.5 for more information on GARCH) on the first 3,000 observations of each data set. For each sample, both the  $A^2$  and the *HitRate* of the 0.001, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99, and 0.999 percentiles were calculated<sup>2</sup> for the remaining 3,000 observations. The average absolute value of the *HitRate* of the 0.001, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99, and 0.999 percentiles were calculated<sup>2</sup> for the remaining 3,000 observations. The average absolute value of the *HitRate* minus its quantile for all the percentiles were averaged to determine a quantile calibration score for the density forecasting model. Formally this is,

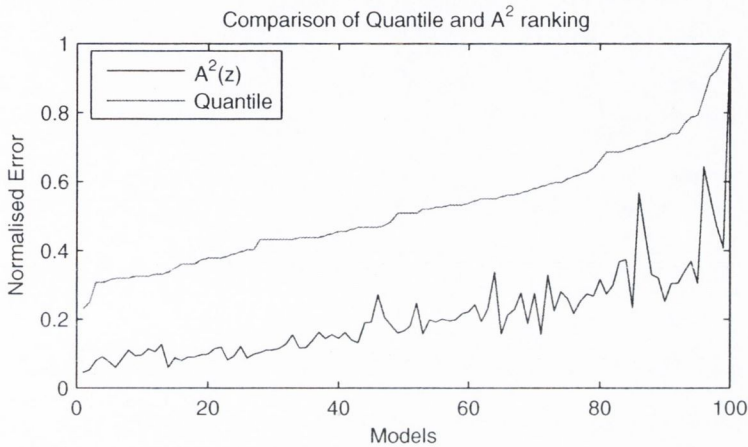
$$Quantile = \frac{1}{Q} \sum_{i=1}^Q |HitRate_i - q_i| \tag{5.7}$$

---

<sup>2</sup>These percentiles were chosen to focus on the tails of the distribution because our application, Value-at-Risk, is most interested in this part of the distribution.



where,  $Q$  is the number of quantiles that are checked. Figure 5.6. plots the normalized *Quantile* and the normalized  $A^2$  score for each model in ascending *Quantile* order. It is clear that there is a strong correlation between the scoring mechanisms. This verifies the legitimacy of the  $A^2$  error function as a means of ranking models in terms of statistical consistency. In Chapter 6 the  $A^2$  is assessed further and shown to perform very well as an estimator of calibration when compared against the Kullback-Leibler divergence and *Quantile*.

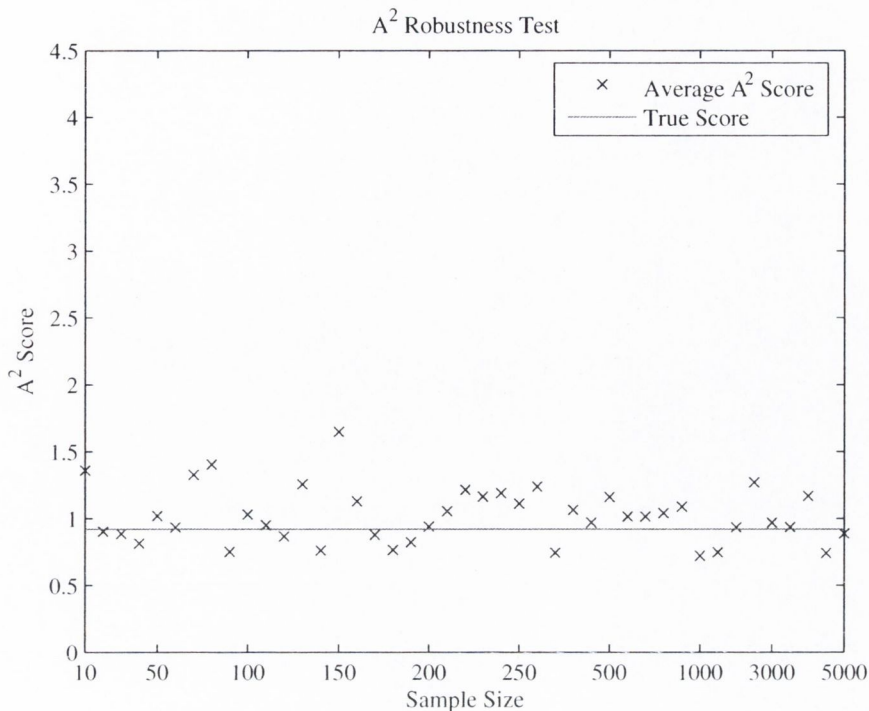


**Figure 5.6:** Comparison of quantile ranking and  $A^2$  ranking on simulated data. The correlation coefficient is 0.874. The quantiles used were 0.001, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.99 and 0.999.

### 5.3.2 Test of $A^2$ Robustness

One of the major obstacles when assessing calibration is that there needs to be a sufficient number of observations to get an indication of the empirical consistency of the probabilistic estimates. However, a good calibration ranking score should be robust enough to perform well even when the observations are scarce. In the following experiment the ability of the  $A^2$  score to provide robust estimates of calibration regardless of the quantity of data is explored.

This experiment is set up as follows. Parameters for a GARCH(1,1) model were chosen at random, the resulting model represents the data generating function and is used to generate samples of data. A sample with 10,000 instances is generated and a new GARCH model is fitted to this data set. The  $A^2$  score is calculated for the fitted model on the data. This result should be a highly accurate estimate of the calibration of the fitted model given the large amount of data used to estimate the score. This value is labelled the *True Score*



**Figure 5.7:** Shown are the average  $A^2$  scores for the same model but with different random samples and sample sizes generated from the same GARCH process. The figure shows that the  $A^2$  score is robust. There is a small variance in the  $A^2$  score with respect to the True Score across the wide range of sample sizes tested.

in Figure 5.7. Using the same model that was fitted to the 10,000 instance data set, the  $A^2$  score is calculated for twenty random samples for a number of different sample sizes ranging from 10 instances to 5000 instances. The average of the twenty scores for each sample size is then reported in Figure 5.7. As can be seen from the figure, the  $A^2$  score is largely unaffected by data size and can give accurate estimates of calibration in problems where the amount of data available is as low as 10. This demonstrates the ability of the  $A^2$  score to report robust estimates of calibration.

## 5.4 Conflicting Objectives

As already mentioned, good sharpness and calibration are the objectives of any density forecast. Therefore, the density forecasting problem is naturally suited to a multi-objective optimisation solution. However, in order for multi-objective search to be legitimately used it is necessary to demonstrate that the objectives being optimised are conflicting i.e. objectives do not converge to the same global minimum in parameter space. This

fact has already been preliminarily demonstrated in the over-fitting example presented in Section 5.2. In this section, an experiment is described that provides further evidence of how sharpness and calibration are different objectives that do not converge to the same solution.

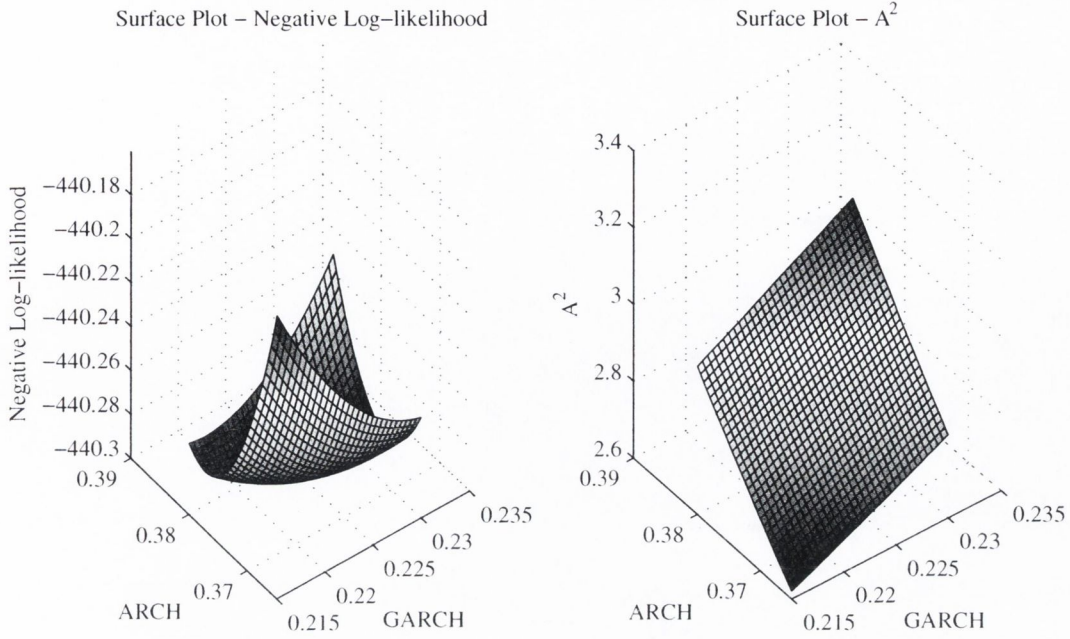
One method of demonstrating that calibration and sharpness are conflicting is to analyse their relationship in terms of the parameter space of a model. This is possible by constructing an experiment using a simple GARCH(1,1) model (see GARCH Section 2.3.5). A synthetic data set is used so that it is possible to correctly specify the other parameters of the GARCH model *a priori* i.e. the constant mean and long-run average variance parameters.

The GARCH(1,1) model assumes the data is generated using a Gaussian distribution, however; to show more clearly the disagreement in the two objective functions the test data is deliberately generated from an EGARCH(1,1) model that assumes a *Student-t* distribution (Nelson, 1991). 1,000 points were simulated from this EGARCH(1,1) model to create a data series. The two free parameters of interest are: the weight applied to the new information for this period that is captured by the most recent squared residual, or the ARCH term; and the weight applied to the variance predicted for this period or GARCH term. To find the global minimum in the context of sharpness for the ARCH and GARCH terms of GARCH(1,1) model on this data set we use the standard NLPD GARCH(1,1) optimisation algorithm.

Since the model is restricted to only two free parameters, it is possible to accurately plot the error functions with respect to parameter space. The resulting function will be a 3 dimensional surface above the 2-dimensional parameter space. Figure 5.8 shows a plot of the error function surfaces in terms of the two parameters (ARCH and GARCH) of the model around the *NLPD* global minimum. It is clear from the surface plots that they are completely different functions and the minima of the two error functions are located in different regions of the parameter space. Therefore, the two objective functions have conflicting goals and are suitable for multi-objective optimisation.

This experiment shows clearly how the  $A^2$  and *NLPD* objectives and hence calibration and sharpness are conflicting. Further evidence of the fact that sharpness and calibration are conflicting is implicit in the evaluation section of this thesis by the presence of a Pareto front after multi-objective optimisation.



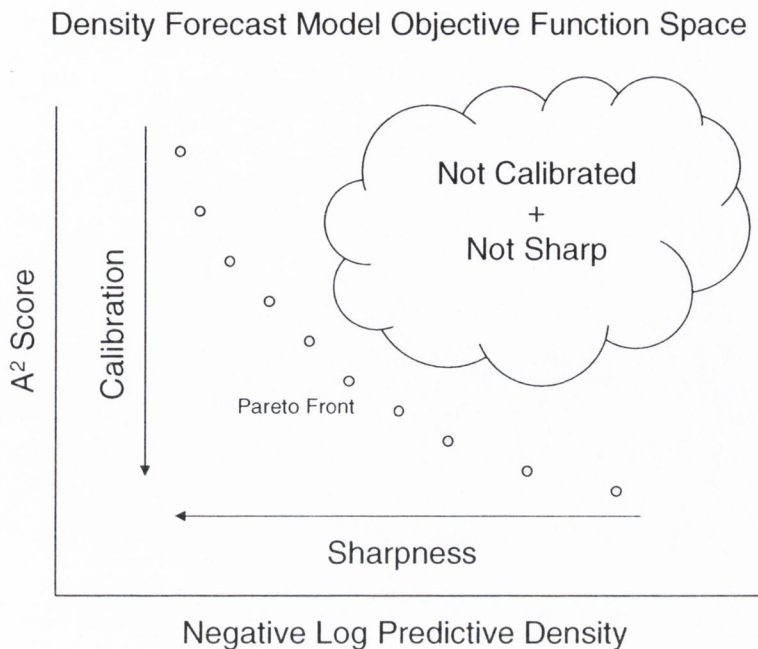


**Figure 5.8:** Comparison of the *NLPD* and  $A^2$  error surfaces in the region of the *NLPD* minimum. The two surfaces clearly have their minimum values in different areas of the parameter space and therefore are conflicting objectives.

## 5.5 A Multi-objective Methodology for Density Forecast Optimisation

Sharpness and calibration are equally important in any density forecasting model. As shown at the beginning of this chapter, optimisation using a sharpness metric alone will often fail to produce calibrated models. Optimisation on calibration will not result in a useful model either as a model can be calibrated but not predictive e.g. by simply predicting the unconditional distribution. Therefore, without a degree of success in both goals, a density forecasting model is useless. In this chapter, the tools and theory required to optimise density forecasting models on both calibration and sharpness are developed. The methodology for optimisation proposed by this thesis is based on a multi-objective optimisation algorithm.

By its nature, density forecasting takes the form of a multi-objective optimisation task. In this section, a general framework for a density forecasting optimisation technique that can find the optimal model in terms of both sharpness and calibration is assembled. In Chapter 4 multi-objective search is introduced, evolutionary algorithms are explained and the *a posteriori* multi-objective evolutionary algorithm technique, that is used by the



**Figure 5.9:** The figure depicts conceptually what the Pareto front of a multi-objective density forecasting model would look like.

density forecasting framework, is motivated. In Section 5.4, experimentation shows that calibration and sharpness are conflicting objectives, confirming the multi-objective nature of the problem. In the next chapter, two algorithms are described, *Pareto-MDN* and *Pareto-GARCH*, and sufficient details are given so the interested reader can implement these algorithms.

In the context of density forecasting the Pareto front will contain two objective functions, one for optimising sharpness and another for optimising calibration. Figure 5.9 is an example of what a density forecast Pareto front should look like after optimisation. Both the sharpness score ( $NLPD$ ) and the calibration score ( $A^2$ ) are negatively oriented resulting in a Pareto arc that is concave in a north west direction.

## 5.6 Conclusions

In this chapter, the most common challenges facing a density forecast modeller were presented. It was demonstrated that the most popular quality scores used are biased toward sharpness rather than calibration. This results in poorly calibrated models in a number of scenarios. As calibration is equally as important as sharpness for any density forecaster this problem presents a serious challenge for the density forecasting field. To attempt

to address the calibration imbalance a new quality score based on the Anderson-Darling test for uniformity called  $A^2$  was presented. It was shown to be robust and suitable for ranking density forecasting models in terms of calibration. Experimentation with this new quality score and the NLPD showed that these objectives were conflicting. This provides a very strong motivation for the use of multi-objective optimisation algorithms. Finally, a description of the proposed solution method for density forecast optimisation was described.



# Chapter 6

## Implementation

### 6.1 Introduction

In this chapter, three algorithms are presented. First the algorithm for Pareto optimisation of density forecasting models is presented. Then, the second algorithm, a variation on the first algorithm that can produce models with better generalisation by utilising a validation set during training is explained. Finally, an algorithm for single objective optimisation of density forecasting models with an evolutionary strategy is described.

To maintain coherency, Table 6.1 is a reference for the various variables used in the following algorithms.

|              |                                                                                                            |
|--------------|------------------------------------------------------------------------------------------------------------|
| $\lambda$    | Number of offspring generated at each iteration.                                                           |
| $\mu$        | Size of parent population.                                                                                 |
| $D$          | Data to be modelled, $[D_{tr} + D_v]$ .                                                                    |
| $D_{tr}$     | Training data.                                                                                             |
| $D_v$        | Validation data.                                                                                           |
| $F_i$        | Non-dominated set of models found up to iteration $i$ based on evaluation on $D$ or $D_{tr}$ i.e. archive. |
| $L$          | Size of initial random population of solutions.                                                            |
| $M$          | Density forecasting model type.                                                                            |
| $T$          | Maximum number of iterations.                                                                              |
| $V_i$        | Non-dominated set of models found up to iteration $i$ based on evaluation with $D_v$ .                     |
| $\mathbf{x}$ | Vector representation of $M$ .                                                                             |
| $\mathbf{s}$ | Strength vector used for guiding ES search.                                                                |

**Table 6.1:** Nomenclature for optimisation algorithms

## 6.2 A Multi-objective Evolutionary Algorithm for Optimisation of Density Forecasting Models

In this section, the theory presented in the previous chapter is developed and presented in algorithmic form. A new algorithm for optimisation of density forecasting models on both sharpness and calibration is proposed and two implementations of the algorithm using different base models are fully described.

---

### Algorithm 1: Pareto Optimisation of Density Forecasting Model

---

*Inputs:*  $D$  data,  $T$  number of iterations,  $L$  initial population size and  $M$  density forecasting model type.

*Outputs:*  $F_T$  non-dominated set of models that are an estimate of the true Pareto front defined by the data generation process.

---

*Algorithm:*

- Step 1: Determine  $\mathbf{x}$ , a vector representation of  $M$ .
  - Step 2: Generate a random population of  $L$  of these vectors.
  - Step 3: For each vector in the population generate a random mutation strength vector,  $\mathbf{s}$ .
  - Step 4: Evaluate population on Sharpness and Calibration scores.
  - Step 5: Create the frontal set  $F_0$  with the non-dominated solutions.
  - Step 6: Initialise iteration counter to  $t = 0$ .
  - Step 7: Select a model,  $\mathbf{x}_t$ , from  $F_t$  to use as population for this iteration.
  - Step 8: Mutate  $\mathbf{x}_t$  using mutation strength vector  $\mathbf{s}_t$  to produce  $\mathbf{x}_t^*$  and  $\mathbf{s}_t^*$ .
  - Step 9: Evaluate  $\mathbf{x}_t^*$  with respect to the error measures. If  $\mathbf{x}_t^*$  is not dominated by any member in  $F_t$  go to 10, otherwise delete  $\mathbf{x}_t^*$  and go to 11.
  - Step 10: Insert  $\mathbf{x}_t^*$  into  $F_t$ . Remove individuals in  $F_t$  that are dominated by  $\mathbf{x}_t^*$ .
  - Step 11: Iterate counter,  $t = t + 1$ . If  $t = T$  then go to 12, else go to 7.
  - Step 12: Terminate the algorithm and save members of  $F_T$ .
- 

Algorithm 1 presents an *a posteriori* multi-objective evolutionary algorithm for optimisation of density forecasting models. Steps 1 - 6 of the algorithm initialise parameters for the optimisation loop, Steps 7 - 11. A set of non-dominated density forecasting models optimised on both sharpness and calibration is returned when the algorithm terminates.

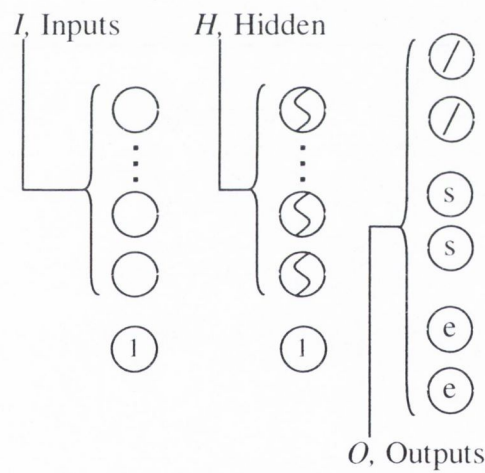
In the next two sections a complete description of how this general algorithm can be implemented to optimise a Mixture Density Network (MDN) and a GARCH type model is

presented. Unless otherwise stated the sharpness and calibration scores used are  $NLPD$  and  $A^2$  respectively.

### 6.2.1 Implementation: Pareto MDN

The first step of Algorithm 1. above requires the density forecasting model to be represented as a vector of parameters. To achieve this for an MDN, all network weights and biases must be collated into a single vector of floating point values. This vector representation must also be reversible so that the vector values can be used to reconstruct the original MDN. Using the methodology introduced in the EANN literature (Yao, 1999), it is possible to vectorize an MDN. For an MDN with  $I$  input features,  $H$  hidden units and  $O$  output units, the length of the vector representation is  $S$ , where;

$$S = (I + 1) \cdot H + (H + 1) \cdot O + 2 \cdot (I + H + O) \tag{6.1}$$



**Figure 6.1:** Shown is a sample MDN network. The locations of the  $I$  input,  $H$  hidden,  $O$  output nodes and bias nodes are marked. The weights for the network are shown in light gray.

$(I + 1) \cdot H + (H + 1) \cdot O$  represents the weight parameters for the MDN, including bias terms. The first  $I + H + O$  represents the mutation strength vector for ES mutation, and the second  $I + H + O$  are binary values that signify the presence or absence of a weight in the network. This binary mask allows the search algorithm to optimise the network architecture during training.



Initial parameter values for all the parameters of the vector representation must be generated. The initial weights of the MDN can be generated as normal using random numbers generated from a Uniform distribution. However, through experimentation, I have found that the best practice is to use a model that is near the global minimum when starting the search. Therefore, by first training a model using the standard gradient method with just NLPD and using this as the starting point of the search the search algorithm is given a point on the Pareto front at the outset. Then the search algorithm needs to spread the Pareto front out as wide as possible from that point. The mutation strength vector values can be generated from a Gaussian random number generator. The binary mask can be initially set to all ones.

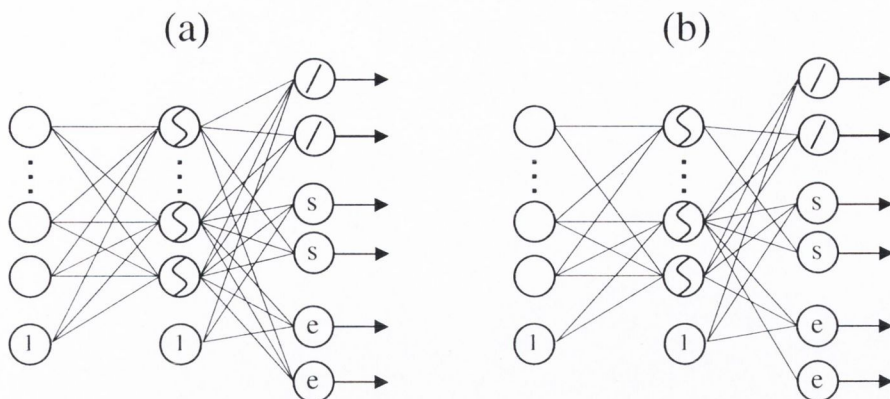
The evaluation metrics used for the Pareto MDN can be chosen by the modeller. From the results of the experiments carried out for this thesis the combination of the NLPD for estimating sharpness and the  $A^2$  for estimating calibration have been found to be the best. However, other combinations could be used, for example, the NLPD could be substituted for the CRPS score.

### 6.2.2 Selection

An unconstrained archive is maintained throughout training and the PQRS method described in Section 4.3 is used for member selection (Fieldsend et al., 2003).

### 6.2.3 Adaptation

Three methods of adaptation are used for optimising the Pareto MDN, weight addition/deletion, node addition/deletion and weight mutation. The Pareto MDN optimisation process includes network architecture optimisation. For this reason, when deciding on an initial architecture for the network it is best to choose a structure that includes all input features, and a maximum number of hidden and output units. The optimisation algorithm will then find the optimal topography during training through weight addition/deletion, and node addition/deletion. Great care has to be taken when selecting an adaptation method as this is the only means of searching the error space (i.e. gradient methods don't work). This is particularly given that multiple objectives require that more of the error space needs to be investigated.



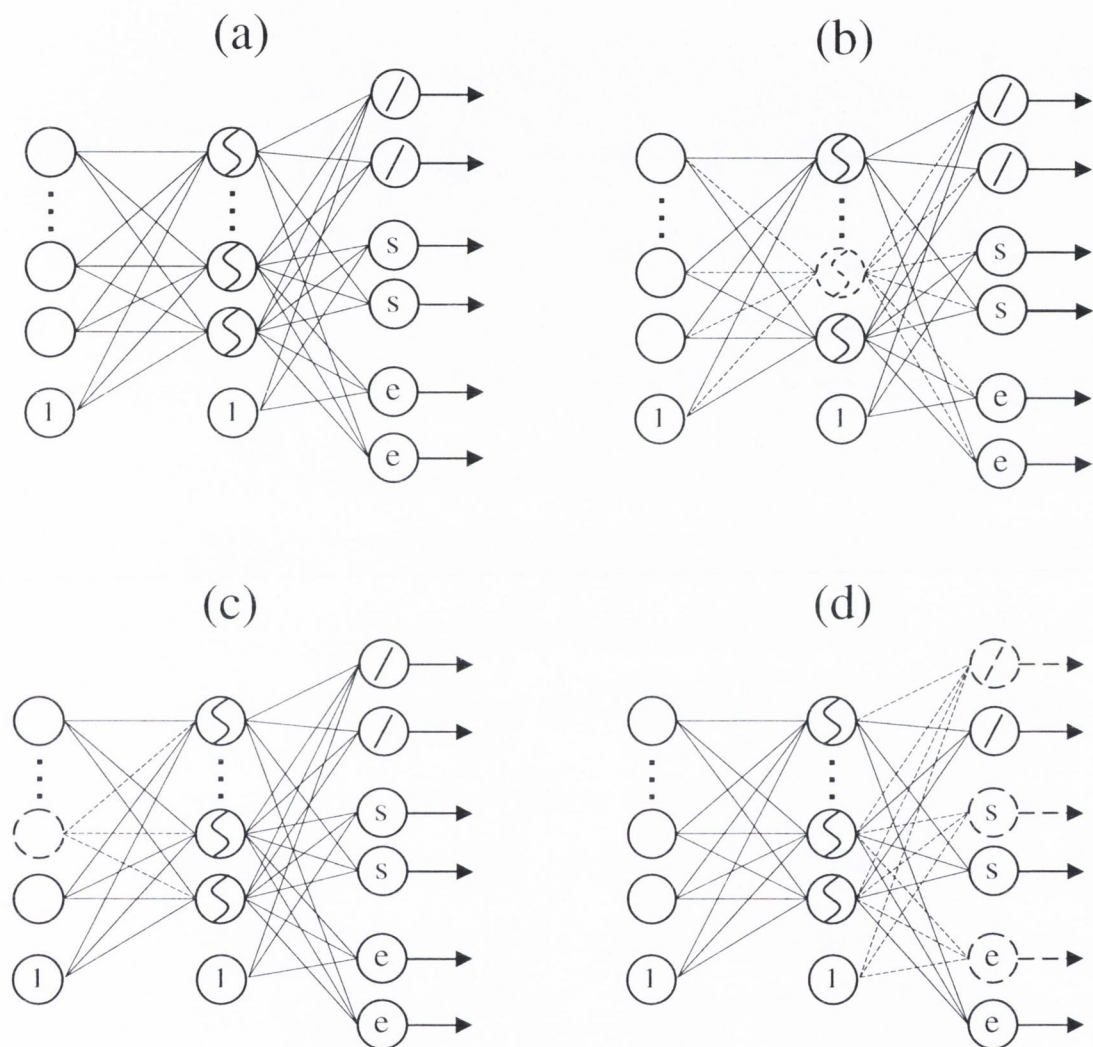
**Figure 6.2:** Shown (a) is a sample MDN network, (b) is the same network after a number of iterations of the evolutionary training algorithm. A number of the weights connecting nodes have been removed.

### Weight Addition/Deletion

The vector representation of the network described above includes a binary mask for each parameter weight in the network. Initially the mask contains all 1s signifying that the network is fully connected. However, during the adaptation phase of the algorithm the binary mask values are randomly flipped from 1 to 0, or vice versa, with a specified probability. When the mask value for a weight is zero, that weight is not included in the structure of the network. This allows the network architecture to evolve. Figure 6.2 shows how the network changes with weight mutation.

### Node Addition/Deletion

Node addition and deletion adapt the number of nodes in the network. At the input layer this has the effect of feature selection. When an input node is dropped the feature associated with that node no longer affects the result of the network. At the hidden layer, node adaptations have the effect of changing the network architecture, either by reducing or increasing the complexity of the network. Typically, output nodes would not be deleted or added during optimisation in regression tasks. However, in this case, addition and deletion of output nodes allows the search algorithm to determine the best number of Gaussian components in the Gaussian Mixture Model output. Rather than single node adaptation, output node adaptations remove the three nodes associated with a Gaussian component at a time. At each iteration of the algorithm, every node has a small probability of addition or deletion. Figure 6.3 shows how nodes from different layers can be deleted



**Figure 6.3:** Demonstration of MDN node deletion. (a) represents the maximum network architecture (b) shows the effect of a hidden node deletion (c) shows the effect of input node deletion and (d) shows the effect of output node deletion. Once nodes have been deleted they can be added back into the network later in the search process.



from a network.

### Weight Mutation

The standard mutation mechanism used for a non-isotropic self-adaptive ES is used for adaptation of the network weights (see Section 4.2.3). At each epoch every weight parameter is mutated with a certain probability.

#### 6.2.4 Implementation: Pareto GARCH

Vector parametrisation of a GARCH model is straightforward given that there are generally quite few parameters and the number remains static throughout optimisation. For example, the equation for a GARCH(P,Q) model is:

$$\sigma_t^2 = \kappa + \sum_{i=1}^P \alpha_i \sigma_{t-i}^2 + \sum_{j=1}^Q \beta_j \epsilon_{t-j}^2 \tag{6.2}$$

The parameters of interest in the equation for the GARCH(P,Q) model above are  $\kappa$ , the weightings  $\alpha$  applied to the  $P$  previous forecasts  $\sigma_{t-1}^2 \dots \sigma_{t-P}^2$  and the  $\beta$  weightings for the last  $Q$  periods' squared disturbances  $\epsilon_{t-1}^2 \dots \epsilon_{t-Q}^2$ . This type of model is commonly accompanied by a constant mean model,  $y_t = C + \epsilon_t$ , that contains the final parameter  $C$  for the parameter vector. The vector representation of a GARCH(P,Q), therefore, takes the form  $\mathbf{x} = (C, \kappa, \alpha_1, \dots, \alpha_P, \beta_1, \dots, \beta_Q)$ .

### Handling Constraints

Initial parameter values are chosen at random from a Gaussian distribution. However, the parameters of GARCH model have various constraints ensuring that the GARCH process generated is correct. Some constraints, such as a parameter must be a positive real number, that can be easily handled by taking the absolute value of that parameter. However, other constraints, for example, that the sum of the GARCH and ARCH terms should not exceed 1, ensure that the process generated remains stationary and need to be handled during the search process so that a valid model is produced. When the initial population is being generated new parameter vectors are created and checked for constraint violation until  $L$  correctly specified individuals are found. This ensures that the search algorithm has the correct population size of valid models at the beginning of the search.

Inevitably, during the search process, the Evolutionary Strategy (ES) mutations will produce parameter vectors that violate the various constraints of the GARCH model. To

provide for this occurrence, new individuals are checked for constraint violations. Rather than repeating the mutation process, individuals that have constraint violations are given a maximum error score across all objectives. This signifies that the model is incorrect and the algorithm will systematically discard these erroneous models through natural selection.

### **Selection and Adaptation**

The selection mechanism used for the Pareto GARCH model is the same as the Pareto MDN method, PQRS. Only one form of adaptation is carried out, weight mutation. The multi-objective algorithm used is based on a non-isotropic self-adaptive ES(1+1) without crossover mutation. Again, this uses a mutation strength vector to guide the search process. Random mutations are generated by Gaussian and Cauchy distributions for fast convergence. No crossover, or parameter addition or deletion is carried out.

## **6.3 A Multi-objective Evolutionary Algorithm for Generalised Optimisation of Density Forecasting Models**

An important issue when optimising any forecasting model is the problem of generalisation error. In order to aid generalisation a common approach is to divide the available data into training, validation and test sets. During training of the model the forecast error on the validation set is monitored and when it begins to increase training is stopped. The goal of this approach is to prevent over-fitting of the training set. This simple method of improving generalisation performance is effective in the single objective space. However, in the multi-objective space it is not possible to identify as clearly when validation set performance begins to deteriorate because the tendency with multiple objectives is for one objective to improve while the others deteriorate. Hence, a different method of improving generalisation performance is required. Fieldsend and Singh (2005) developed a variation of the following algorithm for the optimisation of well generalised neural networks.

---

**Algorithm 2: Pareto Optimisation with Validation**


---

*Inputs:*  $D_{tr}$  training data,  $D_v$  validation data,  $T$  number of iterations,  $L$  initial population size and  $M$  density forecasting model.

*Outputs:*  $V_T$  non-dominated set of models that are an estimate of the true Pareto front defined by the data generation process.

---

*Algorithm:*

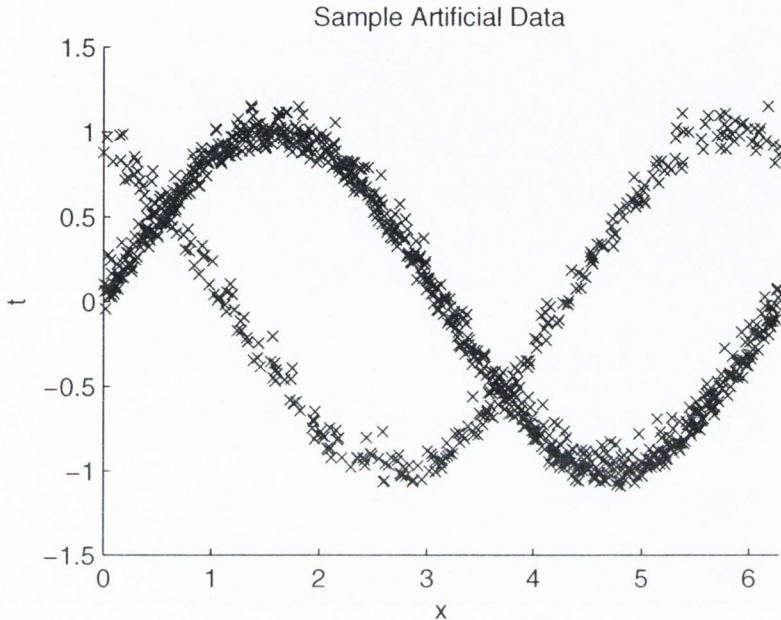
- Step 1: Determine  $\mathbf{x}$ , a vector representation of  $M$ .
  - Step 2: Generate a random population of  $L$  of these vectors.
  - Step 3: For each vector in the population generate a random mutation strength vector,  $\mathbf{s}$ .
  - Step 4: Evaluate population on  $NLPD$  and  $A^2$  with  $D_{tr}$ .
  - Step 5: Create the archive set  $F_0$  with the non-dominated solutions.
  - Step 6: Evaluate each member of  $F_0$  on both error measures with  $D_v$ .
  - Step 5: Create the archive set  $V_0$  with the non-dominated solutions.
  - Step 7: Initialise iteration counter to  $t = 0$ .
  - Step 8: Select a model,  $\mathbf{x}_t$ , from  $V_t$  or  $[F_t \cup V_t]$  to use as the population for this iteration.
  - Step 9: Mutate  $\mathbf{x}_t$  using mutation strength vector  $\mathbf{s}_t$  to produce  $\mathbf{x}_t^*$  and  $\mathbf{s}_t^*$ .
  - Step 10: Evaluate  $\mathbf{x}_t^*$  with respect to the error measures and  $D_{tr}$ . If  $\mathbf{x}_t^*$  is not dominated by any member in  $F_t$  go to 11, otherwise delete  $\mathbf{x}_t^*$  and go to 14.
  - Step 11: Insert  $\mathbf{x}_t^*$  into  $F_t$ . Remove individuals in  $F_t$  that are dominated by  $\mathbf{x}_t^*$ .
  - Step 12: Evaluate  $\mathbf{x}_t^*$  on  $D_v$ .
  - Step 13: Insert  $\mathbf{x}_t^*$  into  $V_t$ . Remove individuals in  $V_t$  that are dominated by  $\mathbf{x}_t^*$ .
  - Step 14: Iterate counter,  $t = t + 1$ . If  $t = T$  then go to 15, else go to 8.
  - Step 15: Terminate the algorithm and save members of  $F_T$ .
- 

This algorithm is very similar to Algorithm 1 with the exception of a validation archive. The implementation process is identical in every detail with the exception of the added evaluations on manipulation of the extra non-dominated set (Steps 5, 8, 12 and 13). After training of the algorithm has terminated the set of non-dominated individuals on the validation set is returned. This set should contain individuals that will generalise well to a test set. Fieldsend and Singh (2005) suggest another method for improving generalisation performance on multi-objective algorithms that involves taking random bootstrap samples from the training data to reduce over-fitting. This methodology is similar to ensembling



and the reader is referred to the Fieldsend paper for further details of this method.

## 6.4 Analysis of the Pareto MDN on Artificial Data



**Figure 6.4:** Sample plot of artificial data set.

In this section, using a synthetic dataset, the proposed calibrated density estimating model is demonstrated. The advantage of using a synthetic dataset is that it is possible to develop a data generating process where the conditional distribution of the data at every point on the function is known. For the purpose of this experiment an artificial data source that has both higher moment influences and bimodality influences is chosen. The data takes the form of a random phase shift sinusoid adapted from (Lo & Bassu, 2002). It is fully described by,

$$f(x) = \sin(x + c) + \epsilon \quad (6.3)$$

where  $x$  is in the interval  $[0, 2\pi]$ ,  $c$  is equal to 0 with probability 0.75 and 2 with probability 0.25; and

$$\epsilon \sim \frac{(W(\lambda, k) - median)}{5} \quad (6.4)$$

where  $W$  is the Weibull distribution, with  $\lambda = 2$  and  $k = 2$  and *median* is the median of the distribution. The median offsets the errors on the sinusoids and the denominator scales the errors. Datasets can be created by drawing values at random from the interval

$[0, 2\pi]$  and evaluating the corresponding function values. 2000 data pairs for training, 1000 for validation and 2000 for testing were generated for each experiment run. Figure 6.8 is a sample plot of a training set generated from Equation (6.3).

#### 6.4.1 Experiment Details

**Table 6.2:** The parameters of the Pareto MDN used on the artificial data sets.

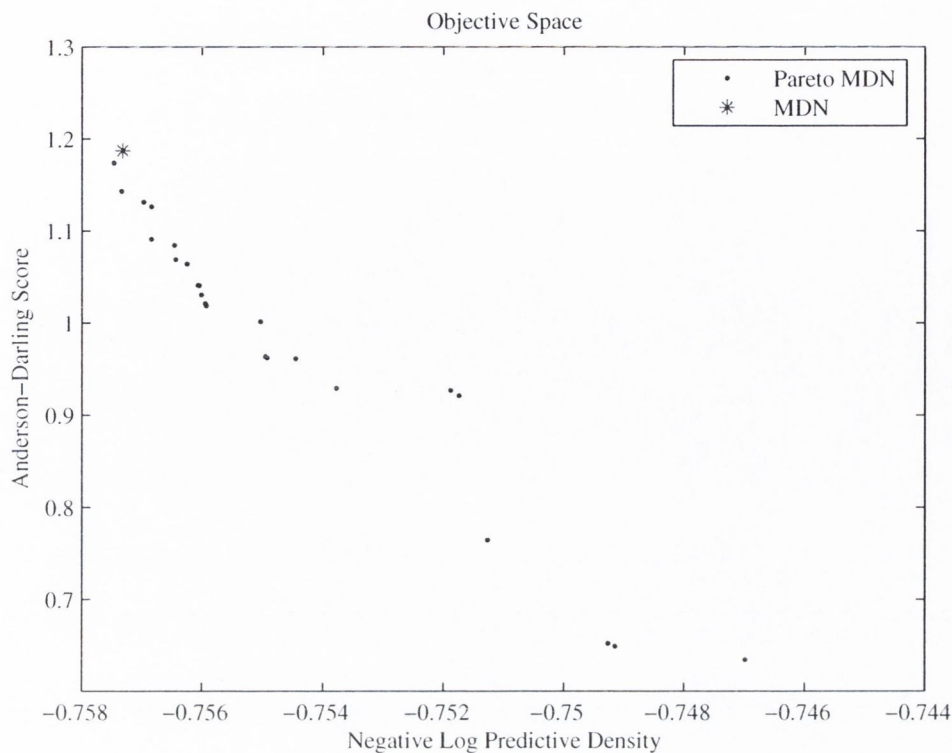
|                                       |                      |
|---------------------------------------|----------------------|
| Initial weights                       | $N(0, 1)$            |
| Probability of weight perturbation    | 0.2                  |
| Perturbation                          | $N(0, 1) \times 0.1$ |
| Probability of node addition/deletion | 0.02                 |
| Probability of weight flip            | 0.02                 |

For each experiment run, a single MDN was trained using a Scaled Conjugate Gradient (SCG) optimisation algorithm for 5,000 epochs. A record of the NLPD score on the validation set is kept for each epoch and the epoch with the lowest error on the validation set determines the weights for the network. This is the base network that is used for benchmarking the calibrated model. It is denoted as ‘Standard MDN’ in the figures and tables. In this experiment the MDNs topology has 5 hidden nodes and 12 outputs, which equates to a 4 Gaussian mixture model conditional density estimate.

Algorithm 2 was used for multi objective optimisation of the MDN. Before commencing training, the archive and validation archive of the multi-objective search are populated by carrying out 200 mutations on the initial solution determined by the Standard MDN. The parameters for the optimisation process are summarised in Table 6.2. Training was carried out for 25,000 epochs. The experiment was repeated 20 times using different initialization vectors, different random seeds and different training, validation and test data sets drawn from the generating function.

#### 6.4.2 Experiment Results

The expectation of the Pareto MDN models is that training will result in a set of models ranging from accurate but poorly calibrated, to less accurate and well calibrated. Figure 6.5 is a plot of the archive of models produced during training on one run of the experiment. The characteristic shape of the non-dominated set can be clearly seen and the high degree of sharpness of the initial Standard MDN solution is present. It is intended that the



**Figure 6.5:** This is a plot of the objective space after training a Pareto MDN on a single run of the artificial data experiment. The initial MDN produced a single very sharp but poorly calibrated model. Using this as an initial solution, the Pareto MDN creates a set of non-dominated or trade-off solutions varying in calibration and sharpness. The user can then pick the model that best suits their objectives.

decision maker selects the model that optimises their objectives in the way that best suits their needs. Therefore, to evaluate the effectiveness of the Pareto model we can suppose that there are two different decision makers (User A and User B). User A wishes to maximise the accuracy/sharpness of their predictions. The other, User B, wishes to maximise the calibration of their predictions. Table 6.3 summarises the performance of the selected models over 20 runs of the experiment, assuming the users choose the model that optimised their goals the most on the validation set. Results are then compared to the Standard MDN.

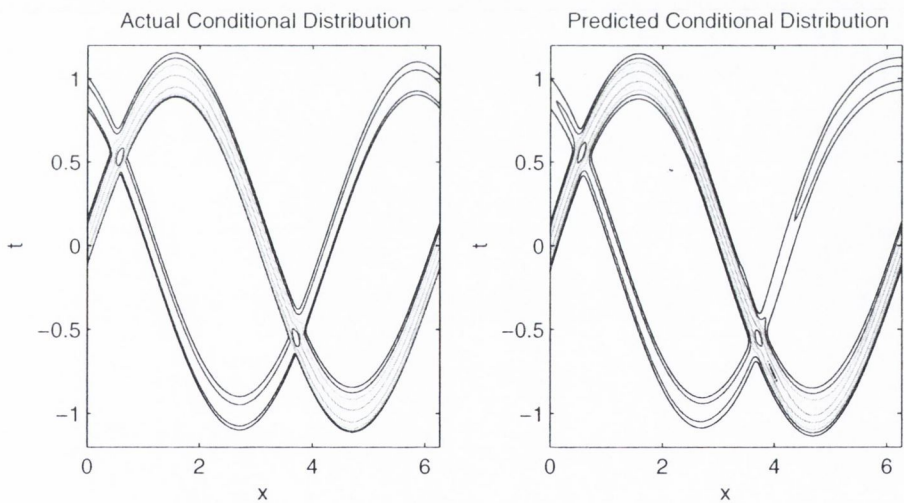
This straightforward evaluation shows that there is an improvement in the calibration score. Figure 6.6 is a surface plot of the predicted conditional densities from a member of a User B validation archive set compared with the real conditional distribution of the data. The synthetic data has been captured well by the MDN. This figure is useful to



**Table 6.3:** The results show the average error and standard deviation over 20 runs of our model on the artificial data. User A and User B both improve their respective objectives. User B obtains a statistically significant improvement over the standard MDN in the calibration metric.

|                     | <i>NLPD</i>             | $A^2$                  |
|---------------------|-------------------------|------------------------|
| Standard MDN        | -0.736<br>±0.057        | 2.942<br>±1.598        |
| User A (Pareto MDN) | <b>-0.737</b><br>±0.056 | 2.432<br>±1.110        |
| User B (Pareto MDN) | -0.734<br>±0.057        | <b>1.391</b><br>±1.057 |

show how well an MDN can predict a conditional density function, however; it is not of high enough resolution to visually evaluate the quality of the predictions in a meaningful way. Instead, to determine whether the  $A^2$  score has improved prediction results, two tests of calibration that leverage the fact that the data is artificially generated are carried out below.



**Figure 6.6:** The plot on the left is the actual conditional distribution of the data and the plot on the right is the predicted distribution. Simple visual comparison shows that the predicted densities accurately capture the dynamics of the data.

### 6.4.3 Analysis of the Calibration Score

To determine whether the  $A^2$  objective function affects the actual calibration of the Pareto MDN predictions the following tests have been carried out. The goal of the calibration

score is to improve the probability estimates of the models optimised thereby improving the accuracy of the predicted density. In the following two tests these two attributes are tested. Firstly, an empirical test on the set of percentiles of the predicted distributions is carried out. The aim of this test is to show whether the calibration score improves probability estimates. Secondly, the performance of models preferred by the calibration score are tested by using the Kullback-Leibler divergence to calculate the average predicted density divergence from the actual conditional density as a true measure of prediction accuracy.

### Empirical Validity - Percentile Test

In Section 5.3 the *QuantileScore* is introduced. In this experiment every 5<sup>th</sup> percentile between 5% and 95% was assessed in terms of *QuantileScore*. Figure 6.7 shows the relationship between the *QuantileScore* and the  $A^2$  score. Models that achieved a good *QuantileScore* also received a good  $A^2$  score and models with a poor hit rate received a high  $A^2$  score. The results, that the  $A^2$  score rewards good probability predictions and penalises poor probabilistic predictions, presented in Section 5.3 are confirmed in the context of the Pareto MDN model.

### Kullback-Leibler Results

The Kullback-Leibler divergence (Kullback & Leibler, 1951) is used to determine the divergence of an estimated distribution  $p$  from the true distribution  $p^{true}$ . If  $p$  and  $p^{true}$  are identical the Kullback-Leibler divergence is zero. The formula is:

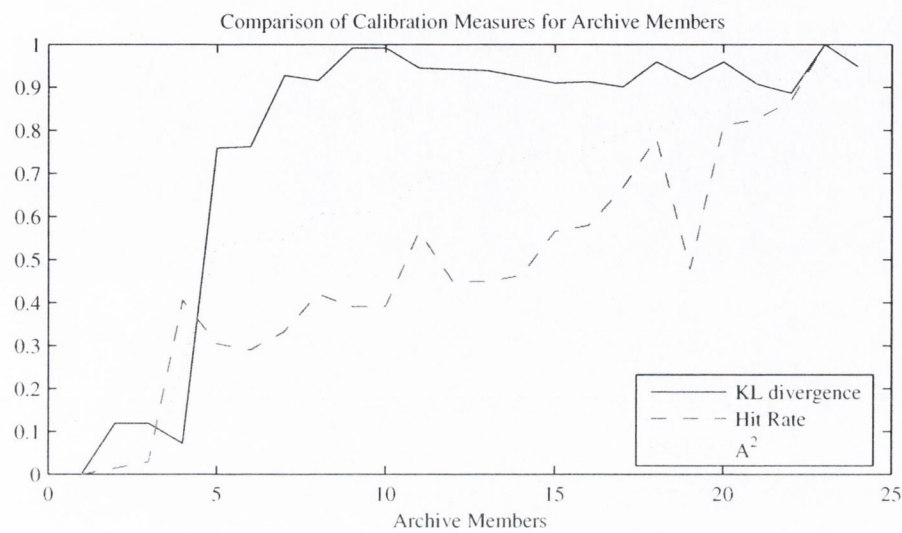
$$KL(p, p^{true}) = \int_{-\infty}^{\infty} p(y) \log \left( \frac{p(y)}{p^{true}(y)} \right) dy \quad (6.5)$$

However, the classic Kullback-Leibler formula, in this form, is asymmetric i.e.  $KL(p, p^{true}) \neq KL(p^{true}, p)$ . Although the  $KL$  divergence is often called a distance metric this is in fact a misnomer. The asymmetric nature of the  $KL$  means that the ‘distance’ from  $p^{true}$  to  $p$  is not the same as the ‘distance’ from  $p$  to  $p^{true}$ . This fact means that the  $KL$  is not a true distance metric. This can be corrected by adjusting the formula to the following, (Siegler et al., 1997):

$$\bar{KL}(p, p^{true}) = KL(p, p^{true}) + KL(p^{true}, p) \quad (6.6)$$

Figure 6.7 has a plot of the normalised average  $\bar{KL}$  divergence against the normalised  $A^2$  for the set of non-dominated models produced after one run of training. The plot shows

a strong correlation between the  $A^2$  and the  $K\bar{L}$  divergence. This is compelling evidence confirming the conjecture that it is possible to optimise an MDN’s calibration and that doing so improves conditional density estimate performance by finding densities that are closer to the true generating density.



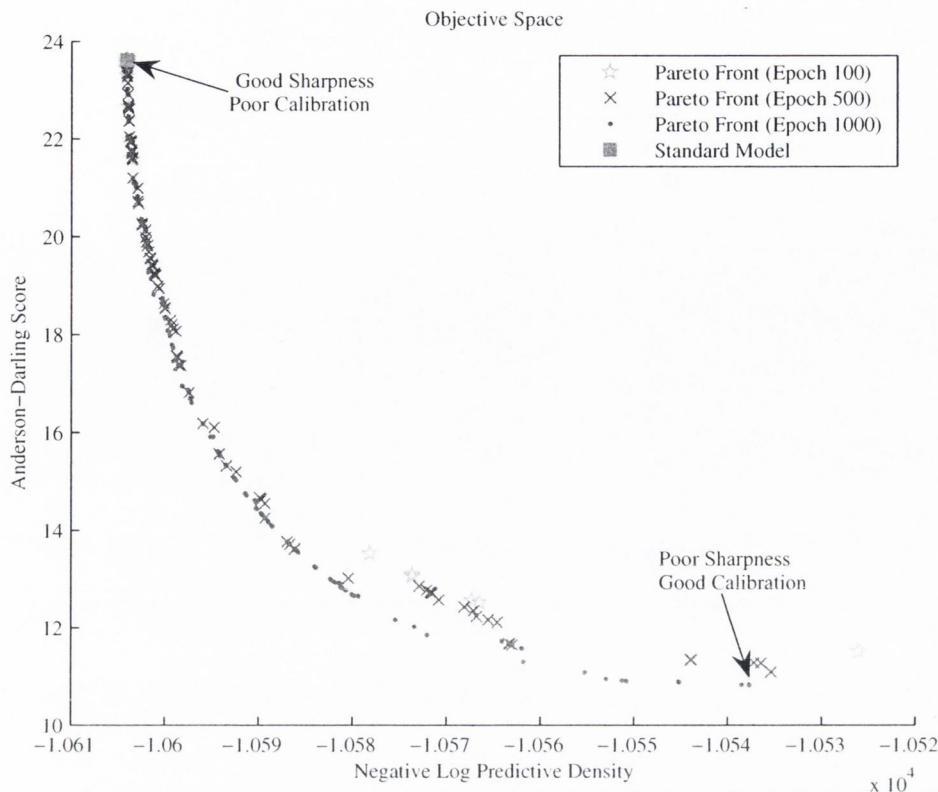
**Figure 6.7:** Typical results returned during a test of the calibration score. The correlation coefficient between the KL divergence and the Calibration score is 0.9 in this example. Similarly, the correlation coefficient between the Hit Rate and the Calibration score is also 0.9.

### 6.5 Sample Pareto GARCH Result

Algorithms 1 and 2 provide a detailed description of how a Pareto GARCH or Pareto MDN model should be constructed. In this section, a brief description of the output of one of these models is provided. In this example, a GARCH(1,1) model with constant mean and the assumption of a Gaussian distribution is trained on historical daily log returns for the General Motors share price. The training data ranges from 1 January, 1986 to 31 December, 1995, a ten year period. A Pareto GARCH(1,1) model using Algorithm 1 is used for training. First a standard GARCH model is trained using a constrained gradient method to find a good starting point for the Pareto GARCH algorithm. Then 100 initial mutations are created to populate the non-dominated archive. Then 1,000 iterations of Algorithm 1 are carried out and a set of non-dominated individuals is returned.

Figure 6.8 shows the output from the Pareto GARCH algorithm. Also included in the





**Figure 6.8:** Sample Pareto GARCH(1,1) output.

figure are the position of the models in the Pareto front archive after 100 iterations of the algorithm and after 500 iterations of the algorithm and the position of the Standard GARCH Model that was used for the initial model in the algorithm. This result is typical for the Pareto Algorithms described above producing a well dispersed Pareto front with a selection of different models ranging from well-calibrated to poorly calibrated. As the algorithm progresses the Pareto front fills in and better results begin to appear. The GARCH model found using the standard gradient method is present in the final Pareto front returned by the Pareto algorithm. This shows how the gradient method is excellent at finding the optimum solution in terms of NLPD. However, the wide Pareto front shows that calibration can be greatly improved for this model by allowing a small deterioration in the level of sharpness.

## 6.6 An Evolutionary Algorithm for Optimising Density Forecasting Models

The central argument in this thesis is that multi-objective optimisation of density forecasting models in terms of sharpness and calibration is superior to single objective optimisation methods. However, in order to prove this hypothesis it is useful to be able to compare directly single and multi-objective methods. In this section, a single objective algorithm for optimisation of density forecasting models is presented and implementation details for MDN and GARCH are provided.

---

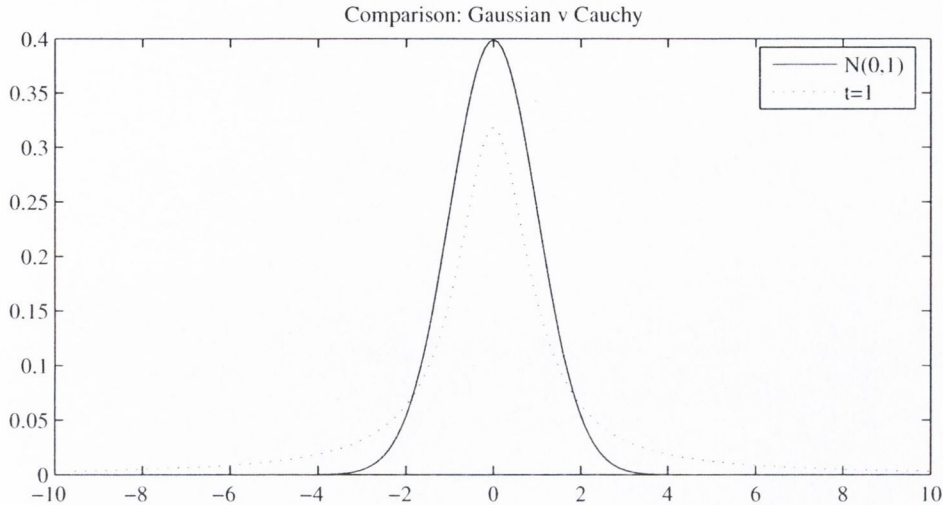
|                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Algorithm 3: ES Optimisation of Density Forecasting Model</b>                                                                                          |
| <i>Inputs:</i> $D$ data, $T$ maximum number of iterations, $\mu$ number of offspring, $\lambda$ parent population size and $M$ density forecasting model. |
| <i>Outputs:</i> $\mathbf{x}_{T, \text{fittest}}$ parameters of fittest model after $T$ iterations of the algorithm.                                       |
| Step 1: Initialise the $\mu$ model parameter vectors $\mathbf{x}$ and strength vectors $\mathbf{s}$ .                                                     |
| Step 2: Set maximum number of iterations, $N$ .                                                                                                           |
| Step 3: Initialise iteration counter, $t = 1$ .                                                                                                           |
| Step 4: Generate $\lambda$ offspring from $\mu$ parents.                                                                                                  |
| Step 5: Evaluate offspring.                                                                                                                               |
| Step 6: Select new $\mu$ members for population.                                                                                                          |
| Step 7: $t = t + 1$ .                                                                                                                                     |
| Step 8: If $t = T$ terminate algorithm, otherwise go to 4.                                                                                                |

---

Algorithm 3 shows the 8 simple steps in an ES for single-objective optimisation of density forecasting models. The main loop of the algorithm is between Step 4 and Step 8, where the three main parts of any EA (select, mutate and evaluate) reside. The following two sub sections describe how this algorithm can optimise an MDN and a GARCH model.

### 6.6.1 Implementation: EMDN

Much of the implementation details for the single objective EA are similar to the multi-objective EA described above. Specifically, the vector representation of the MDN and the adaptation methods are identical. However, the selection method is different as there is only one objective. Tournament selection is used at each iteration to determine which population members will progress to the next iteration of the algorithm. Again, a non-



**Figure 6.9:** Comparison of Cauchy and Gaussian Distributions

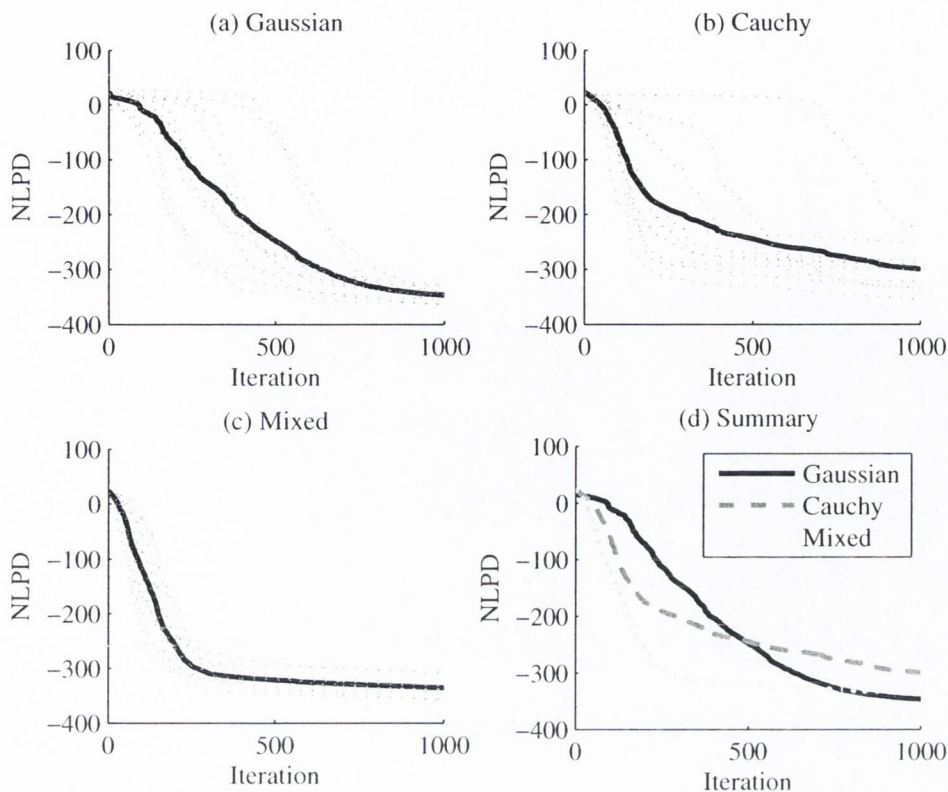
isotropic self-adaptive ES is used. However, rather than the  $ES(1 + 1)$  used in multi-objective optimisation an  $ES(20 + 20)$  is used, as in (Jerrell & Campione, 2001).

Because this is single-objective optimisation the default error function used is *NLPD*, however, other error functions can be used also e.g. *CRPS*. Variations on Algorithm 3 could also be implemented that would lead to faster convergence times. For example, Yao (1999) presents a hybrid method to optimisation that combines gradient optimisation and evolutionary algorithms. However, this method would not be possible within the multi-objective optimisation algorithms because gradient algorithms can only optimise one objective at a time.

## 6.7 Reducing Optimisation Time with the Cauchy Distribution

Yao and Liu (1997), suggest that convergence times can be reduced by using random numbers generated from a distribution other than the standard Gaussian. Specifically, they recommend using a distribution with fatter tails than the Gaussian, such as the Cauchy distribution. The fatter tails in this distribution mean that it is more likely that the mutation will make larger steps around the search space than a Gaussian distribution. This reduces the time to convergence by finding good locations to search more quickly. Figure 6.9 shows the difference between the Cauchy and Gaussian distribution. The much higher proportion of density in the tails of the Cauchy can be clearly seen.





**Figure 6.10:** Shown are the error rates of ten models trained on the logistic function data set for 1,000 epochs using random mutations from a (a) Gaussian distribution (b) Cauchy distribution and (c) Mixed Gaussian and Cauchy distribution. The heavy line in each plot represents the mean of the ten runs. (d) compares the mean error plots across the three methods.

It is possible to demonstrate the effect that this fatter tailed density has on convergence rates by means of a comparative analysis. The following experiment describes three different mutation strategies: random Gaussian mutations, random Cauchy mutations and mixed random mutations, half Gaussian and half Cauchy. The *logistic* data set was chosen as the data generating process (see Section 5.2.1) and the model chosen for optimisation was an EMDN with 3 hidden units and a single Gaussian GMM as output<sup>1</sup>. 250 data points were generated from the logistic data set for training the models. Each model type was trained 10 times on the data. Figure 6.10 shows the results of these experiments.

The increased possibility of big jumps in the search space that the Cauchy distribution produces results in faster optimisation at the beginning of the search process, however,

<sup>1</sup>The simplicity of dealing with only one objective function means that experiments with different optimisation parameters are much easier to compare. For this reason, this single objective density forecasting optimisation algorithm is used in some of the following experiments.

these large steps slow the optimisation process as the parameters near the global minimum. The reverse effect is seen using a Gaussian distribution, with poor performance at the beginning of the search. However, this distribution homes in on the global minimum much faster when the parameters are close to the optimum.

Overall, the mixed approach performs best. As expected, Cauchy produces bigger improvements early on in the search process whereas Gaussian is slower but ultimately produces a better solution. Due to the fact that a combined approach performs better on average than using only random draws from a Gaussian or Cauchy distribution the mixed optimisation method is used for random number generation in all optimisations in this thesis.

## 6.8 Conclusions

The algorithms and experiments above show how evolutionary algorithms can be used to optimise two particular density forecasting models. The algorithm presented is sufficiently flexible so that it could be applied to other density forecasting models other than GARCH and MDN.

These new evolutionary techniques have been made possible by the relative inexpense of computer cycles resulting in an upsurge in the number of new applications for these methods. This provides a means of optimising on two or more objectives simultaneously. Moreover, evolutionary algorithms are now being considered as viable alternatives to gradient approaches in finance and other application domains where the ability of an algorithm to optimise in almost real-time is paramount.

## Chapter 7

# Finance: Value-at-Risk

### 7.1 Introduction

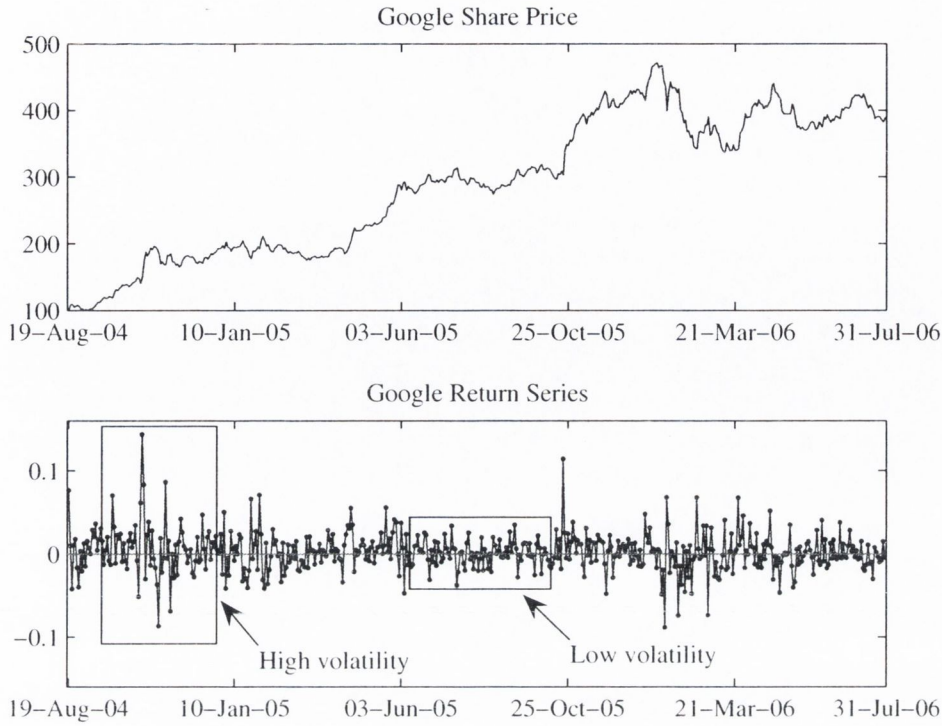
The finance world revolves around predictions and estimates of what will happen in the future. Will interest rates be higher in six months time than they are now? Or, will my stocks in Elan go through the roof or through the floor? Success in the financial world is dependent on ones ability to predict what will happen in the future. Therefore, regression is widely used throughout finance; applied econometricians typically use a sum-of-squares type model to determine the relationship between different variables. Unfortunately for the econometricians, predicting tomorrow's stock price with any regression model is not straightforward. There are many theories on what structure financial data takes but no general consensus e.g. Malkiel (1973) suggests a random walk and Mandelbrot (2004) argues the distribution of returns follows a fractal distribution.

Although predicting the conditional mean accurately has remained elusive to most econometricians aside from the very lucky (Taleb, 2001), the accurate estimation of the conditional variance has become an area of fruitful interest. The reason for focusing on anything other than the first moment is mainly down to the recent increased interest in quantifying risks and exposures<sup>1</sup>. In finance, the conditional variance of a financial instrument is called its volatility. This is a measure of fluctuation in price. Traditionally, it refers to the annual standard deviation of returns, but, more recently, volatility is calculated weekly, daily and even hourly. Fortunately, the volatility of financial securities tends to be well structured, with certain patterns reappearing in most time series of

---

<sup>1</sup>There are other applications such as derivative pricing that also require estimates of higher order moments.





**Figure 7.1:** This figure shows a sample daily price series (top) and corresponding return series (bottom). Areas of high and low volatility are highlighted on the return series.

historical prices. The most prolific of these patterns is called “*volatility clustering*”. This is a pattern of autocorrelations in the volatility of financial series i.e. periods of low volatility are followed by more low volatility and periods of high volatility are followed by periods of high volatility. Figure 7.1 shows a time series and demonstrates the areas of volatility clustering.

In the example in the figure the bottom plot is the stock price series converted to a return series. In finance, it is common practice to convert a price series into a set of relative prices that represent a price change as relative to some initial price. The converted price is then known as a return. The main reason for doing this is that returns have more attractive statistical properties than prices. For example, volatility clusterings can be clearly seen. The equation for converting a price to a log return is<sup>2</sup>;

$$r_i = \log \left( \frac{p_i + 1}{p_i} \right) \quad (7.1)$$

<sup>2</sup>There are other possible forms of returns e.g. absolute return, relative price return etc. However, the convention is to convert to log returns.

where,  $p_i$ , is the price at day  $i$ . The notation here varies from that used in other regression problems as the data is chronologically ordered in a time-series. The dependent variable is the return for the next time step,  $r_{t+1}$  and the independent variable is typically the value of the return for the current time step  $r_t$ . In this chapter all price data is converted to a returns series using the formula above.

Some of the possible applications of density forecasts in finance include estimating Value-at-Risk (Manganelli & Engle, 2001), options pricing (Christoffersen & Jacobs, 2004) and high-frequency trading (Diebold et al., 1998a). How to produce, use, and evaluate this type of forecast has become a substantial sub-field of the literature. A popular density forecasting model used in the finance community is GARCH (see Section 2.3.5). This simple model can successfully estimate the serial dependence of the volatility (conditional variance) of financial data with surprising accuracy. In this chapter, the performance of standard negative log-likelihood optimised GARCH is contrasted with the Pareto GARCH optimised models on a variety of data sets and results are compared with respect to a risk management application. A comparison and analysis of the performance of Mixture Density Network (MDN) and Pareto MDN models is also included. The following section describes the application area Value-at-Risk.

## 7.2 Value-at-Risk

Measuring risks associated with being a participant in the financial markets has become the focus of intense study by banks, corporations, investment managers and regulators (Riskmetrics, 1996). Accurate estimation of risk requires a forecast of the probability of future events. Diebold et al. (1998b) go so far as to say, *“The booming area of financial risk management is effectively dedicated to providing density forecasts of portfolio values”*. There are a number of possible catalysts for this recent boom, such as;

**Financial Disasters** In recent years financial disasters have emphasised the need for risk management e.g. 1987 crash, Nick Leeson and Baring Plc, Long Term Capital Management, etc.

**Regulatory Environment** The Securities and Exchange Commission and the Basel Committee on Banking Supervision require the risk exposure of banks to be reported.

**Performance** There has been a shift from a focus solely on the returns of an investment

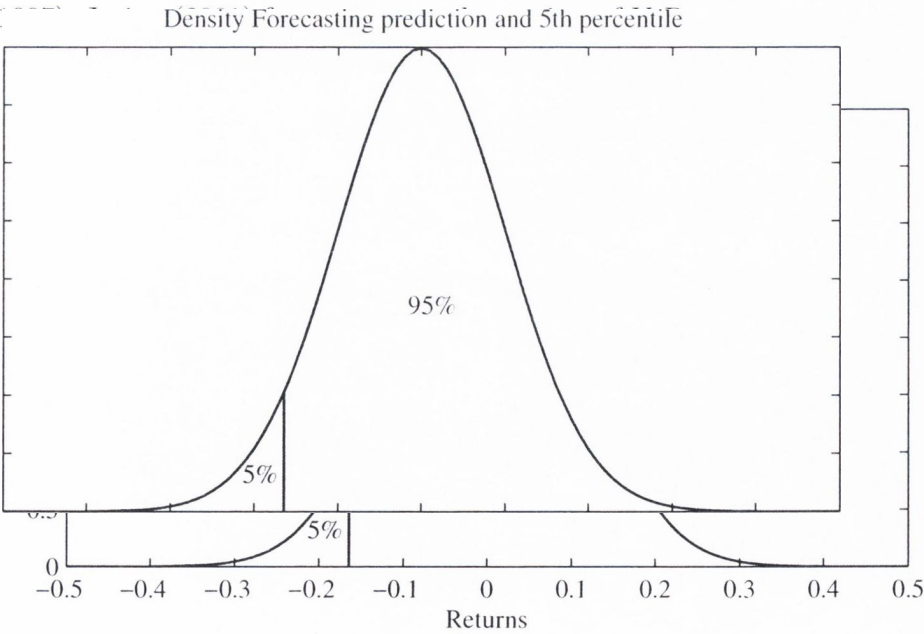
to return relative risk.

**Securitization** Global securities markets are expanding and derivatives markets are now major components highlighting the potential volatility of earnings.

One approach to risk management known as Value-at-Risk (VaR) has gained increasing acceptance in the last ten years. Value-at-Risk answers the question, “*Over a given period of time with a given probability, how much could the value of the equity / fx / index / portfolio decline?*”. VaR is a metric, a way of interpretation, rather than a measure. In essence, it is the expected loss, measured typically in dollar terms, on a given portfolio of assets in a given time frame. The operationalising of this measure requires that an estimate of the likelihood of a given percentage change in the returns on which the loss is to be measured. For example, an  $\alpha = 5\%$  1-day VaR of \$5 million can tell us that on one out of 20 days, we could expect to realise a loss of at least \$5 million. Alternatively, the maximum loss we would reasonably expect from our model on 19 out of 20 days is \$5 million. In other words, VaR is defined as the maximum loss over a given time horizon at a given confidence level. Formally, the Value-at-Risk for the next time step,  $t$ , is;

$$VaR = - \left( \int_{-\infty}^{\alpha} p(r_t|r_{t-1}...r_1) \right)^{-1} \tag{7.2}$$

The VaR function is the inverse cumulative of the predicted distribution. See Duffie and Pan (1997) for more details.



**Figure 7.2:** Shown is a sample predicted density forecast, highlighted is the 5th percentile. The -VaR is -0.1645 at the 5th percentile.



Figure 7.2 is a sample plot of a predicted probability density forecast. The 5th percentile is highlighted. The inverse cumulative value of the predicted distribution at this point is -0.1645. Therefore, the 5% VaR for this sample is 0.1645 per share in the portfolio. This conceptual simplicity of VaR makes it an appealing measure. However, a standard VaR based risk management system has never been agreed upon. This is due to several factors, Simons (2000) identifies a few;

- There is a lack of generally accepted standards.
- Risk-management systems developed for trading are expensive to implement and beyond the budget of smaller institutional investors.
- Different VaR approaches are needed to cater for different trading operations.

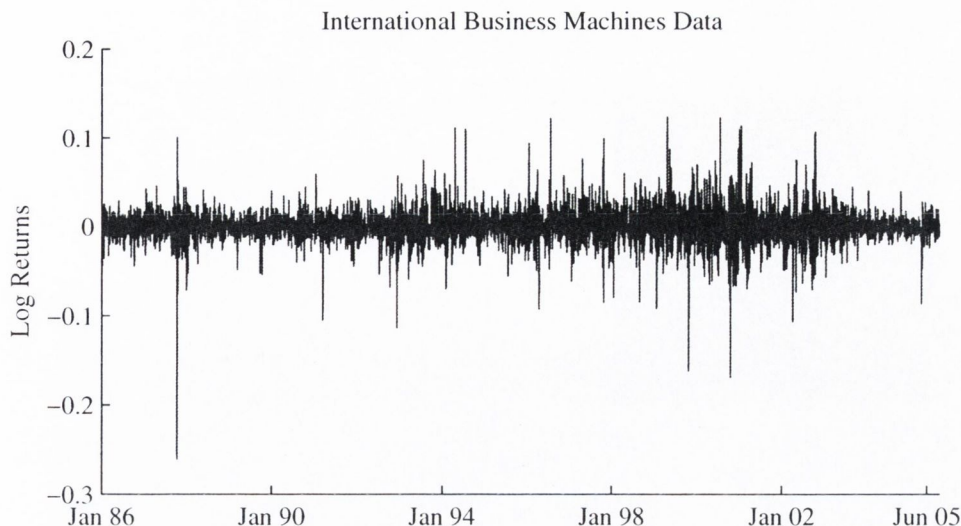
Therefore, VaR calculation is an unsolved problem and is an interesting application to compare the results of our optimisation approach with other approaches. Furthermore, the objective of VaR to report an accurate estimate of a particular percentile makes the objective of calibration paramount.

### 7.3 Setup of Experiments

In this chapter, three experiments are presented that compare the optimisation performance of standard negative log-likelihood optimised models with that of models optimised using the framework discussed in Chapter 6. Specifically, improvements that can be achieved using Pareto GARCH and Pareto MDN over the standard GARCH and MDN models are focused on. To demonstrate the practical implications of improvements in model performance, models are compared in terms of accuracy at predicting VaR. In each experiment the general set up is the same. The data is first transformed to returns. Figure 7.3 shows a sample return series plot. The training and test periods are then selected. The training data is used to optimise the models that are to be compared and results are presented and discussed for both in-sample and out-of-sample performance.

### 7.4 Experiment 1: Standard GARCH vs Pareto GARCH

In this section, a complete and rigorous evaluation of the proposed Pareto optimisation approach using historical data is presented. The data used is a sample of 5,145 daily



**Figure 7.3:** Log return series of IBM data. Shaded area represents out-of-sample data set.

closing prices for International Business Machines (IBM) from Datascope. These prices represent the contiguous period of trading from 1 January, 1986 to 20 September, 2005. For convenience, this price series was transformed into daily log returns. The data was divided into a training set consisting of the first 3,145 observations and a test set consisting of the last 2,000 observations. Figure 7.3 shows the IBM log return series.

The density forecasting model that was chosen to estimate this series was an  $\text{ARMA}(1,1)/\text{GJR}(1,1)$  model with a Gaussian distribution, resulting in a model consisting of 7 parameters to be optimised. This model was then optimised using three different optimisation algorithms, the constrained non-linear quasi-Newton optimisation algorithm implemented by Matlab, the Pareto multi-objective approach presented in Chapter 6 and a single objective evolutionary algorithm approach that minimises *CRPS*. The resulting models will be referred to as the Standard Model, Pareto Model and *CRPS* Model respectively.

The Pareto Model was optimised using Algorithm 1. The algorithm ran for 10,000 epochs. This very long period of training was chosen deliberately so that the most diverse set of solutions possible was found on the Pareto front. This gives a near complete view of the Pareto frontier facilitating clear contrasting of results from the different optimisations and model selections. Typically, it is sufficient to train the algorithm for approximately 1,000 epochs to get a diverse set of solutions.

As described above the Continuous Ranked Probability Score is another metric used for assessing the quality of a density forecasting model. Gneiting et al. (2005b) suggest that

the *CRPS* is in fact a more robust error function than *NLPD* when optimising a density forecasting model and they give examples of where this is the case in meteorology. To test this hypothesis on financial data a simple model has been developed that will facilitate a comparison of the models optimised on the *CRPS* score to the models produced by the Pareto approach and the standard non-linear quasi-Newton optimisation approach. As the ARMA(1,1)/GJR(1,1) model has a Gaussian distribution it is possible to optimise the *CRPS* score using its simplified Gaussian derivation, see (Gneiting & Raftery, 2004).

$$\text{crps}(N(\mu, \sigma^2), x) = \sigma \left( \frac{x - \mu}{\sigma} \left( 2\Phi \left( \frac{x - \mu}{\sigma} \right) - 1 \right) + 2\phi \frac{x - \mu}{\sigma} - \frac{1}{\sqrt{\pi}} \right) \quad (7.3)$$

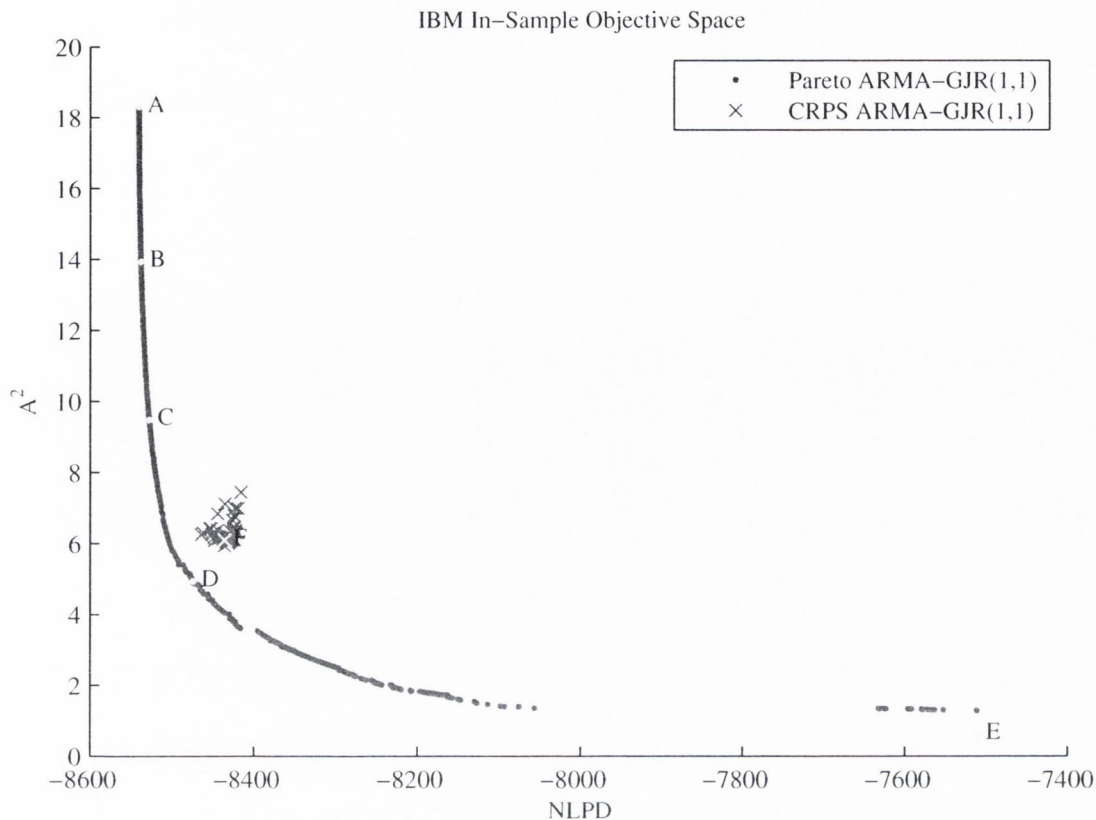
where,  $\phi$  and  $\Phi$  denote the probability density function and the cumulative probability density function of the standard normal distribution. This approach is not possible with models that have more complex distributions. Algorithm 3, the single objective evolutionary algorithm combined with the CRPS error score was used for optimisation. With this set up, fifty ARMA(1,1)/GJR(1,1) models were optimised using an ES(20+20) and CRPS for 1,000 epochs.

## Experiment Results

Figure 7.4 shows all models produced by the three optimisation approaches described above on the IBM training data. The Pareto front consists of 1,813 non-dominated individuals that represent a very diverse set of ARMA(1,1)/GJR(1,1) models. All of the fifty *CRPS* models are dominated by the Pareto front. Given that the *CRPS* models are not directly optimised on either the *NLPD* or the  $A^2$  score this result is understandable. To evaluate the results, 6 models were selected for comparison. Model A represents the model with the best *NLPD* score on the training data. This model is both a member of the Pareto front and the Standard Model. Models B, C, and D are three approximately equally spaced models chosen from within the Pareto front and Model E is the ARMA(1,1)/GJR(1,1) model that achieved the lowest  $A^2$  score on the training data. Finally, Model F is the *CRPS* model that had the lowest *CRPS* score on the training data.

An interesting aspect of the shape of the Pareto front is that near Model A it is possible to make big improvements in calibration without losing a lot of sharpness in predictions. However, as you move further from Model A towards Model E the improvements in calibration require larger losses in *NLPD* until there is a large gap in the Pareto front in terms of *NLPD*. It is at this point that the ARMA(1,1)/GJR(1,1) models change from attempt-

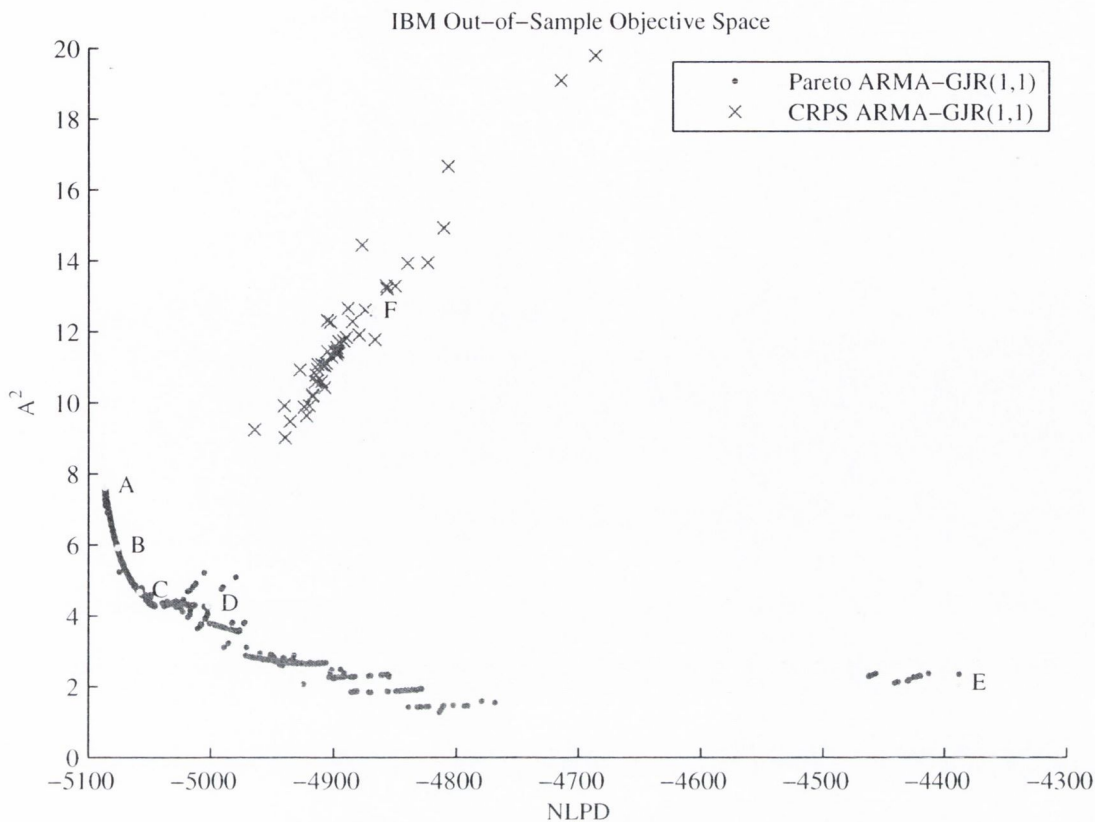




**Figure 7.4:** Shown above are the models produced by the 3 optimisation approaches described in the *NLPD* versus  $A^2$  objective space. Models A, B, C, D, E and F are highlighted.

ing to predict the conditional distribution to producing static estimates of approximately the unconditional distribution for each prediction. This observation is confirmed in Figure 7.7 where the sample autocorrelation of predicted densities shows strong persistent correlations.

Figure 7.5 shows the same models as in Figure 7.4 but evaluated on the out-of-sample data. The basic shape of the Pareto front is maintained over the out-of-sample data and the models have a similar ranking. In fact, the Spearman rank correlation of models between in-sample *NLPD* and out-of-sample *NLPD* is 0.995 and between in and out-of-sample the  $A^2$  score is 0.9834. This suggests that little over-fitting of the error functions occurred during training. Figure 7.5 also shows how the *CRPS* models are once again dominated by the Pareto front, this time, however, the difference in results is far greater. This is quite surprising as the *CRPS* is over-fitting on measures it is not trying to optimise.



**Figure 7.5:** This figure shows the same models as figure 7.4 above evaluated on the out-of-sample data.

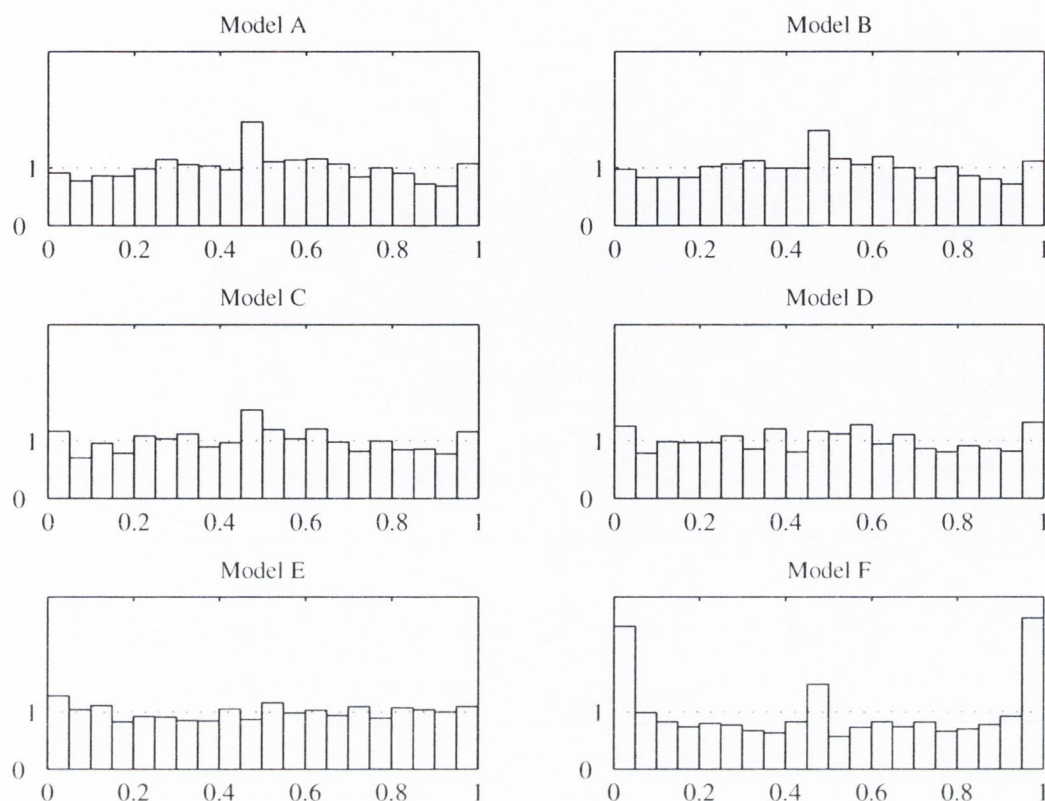
**Table 7.1:** The  $NLPD$ ,  $A^2$  and  $CRPS$  scores achieved by the 6 models on the in and out-of sample data.

| Model | In Sample |       |          | Out-of-Sample |       |          |
|-------|-----------|-------|----------|---------------|-------|----------|
|       | $NLPD$    | $A^2$ | $CRPS$   | $NLPD$        | $A^2$ | $CRPS$   |
| A     | -8541     | 18.30 | 0.008744 | -5086         | 7.62  | 0.010821 |
| B     | -8538     | 13.93 | 0.008730 | -5076         | 5.92  | 0.010822 |
| C     | -8528     | 9.49  | 0.008715 | -5058         | 4.67  | 0.010823 |
| D     | -8474     | 4.94  | 0.008814 | -5002         | 4.29  | 0.011048 |
| E     | -7510     | 0.61  | 0.012855 | -4388         | 2.05  | 0.016639 |
| F     | -8436     | 6.10  | 0.008671 | -4870         | 12.53 | 0.010867 |

### 7.4.1 Model Evaluations

Table 7.1 is a summary of the highlighted models' scores on the Pareto front. Shown are the  $NLPD$ ,  $A^2$  and  $CRPS$  scores achieved by the models on the in and out-of sample data. The  $NLPD$  and  $A^2$  scores are already displayed in the objective space figures above, however, of notable interest are the  $CRPS$  scores obtained by each model. Models A, B and C out-perform Model F, the  $CRPS$  optimised model, in the out-of-sample data on all metrics including  $CRPS$ . This suggests that the  $CRPS$  score, as an objective function for

financial applications, is not as effective as the *NLPD* or the *NLPD* and  $A^2$  optimised models.

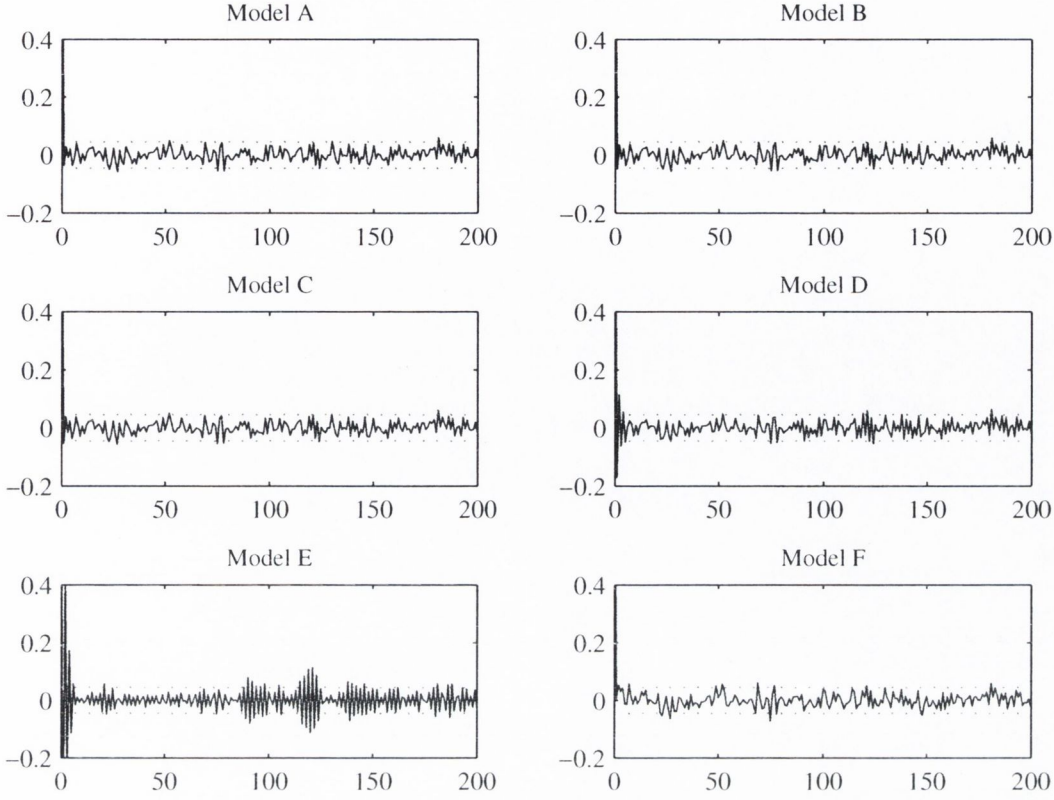


**Figure 7.6:** PIT histograms for selected models. The plot shows how models with a low  $A^2$  score have better uniformity. Also, the PIT histogram of Model F shows that the *CRPS* score has resulted in an over-confident model.

Figure 7.6 depicts the Probability Integral Transform histograms produced for each of the six models on the out-of-sample data. The plots show how the models with low  $A^2$  score (Models C, D and E) have a much more uniform series of PIT values. Also, Model F, the *CRPS* model, produces over-confident predictions resulting in a U shaped PIT histogram.

An important aspect of the evaluation of density forecasts in the context of sequential data, such as the time series used here, is the independence of the conditional density estimates. This requirement manifests itself in the fact that PIT values should not only be uniform but also should be *i.i.d.*. There is an argument for this attribute to be included as a further objective in the search process. However, capturing the various possible forms of dependencies in the predictions would result in a number of added objectives rather than just one. Fortunately, our experiments suggest that this is not necessary. Instead,

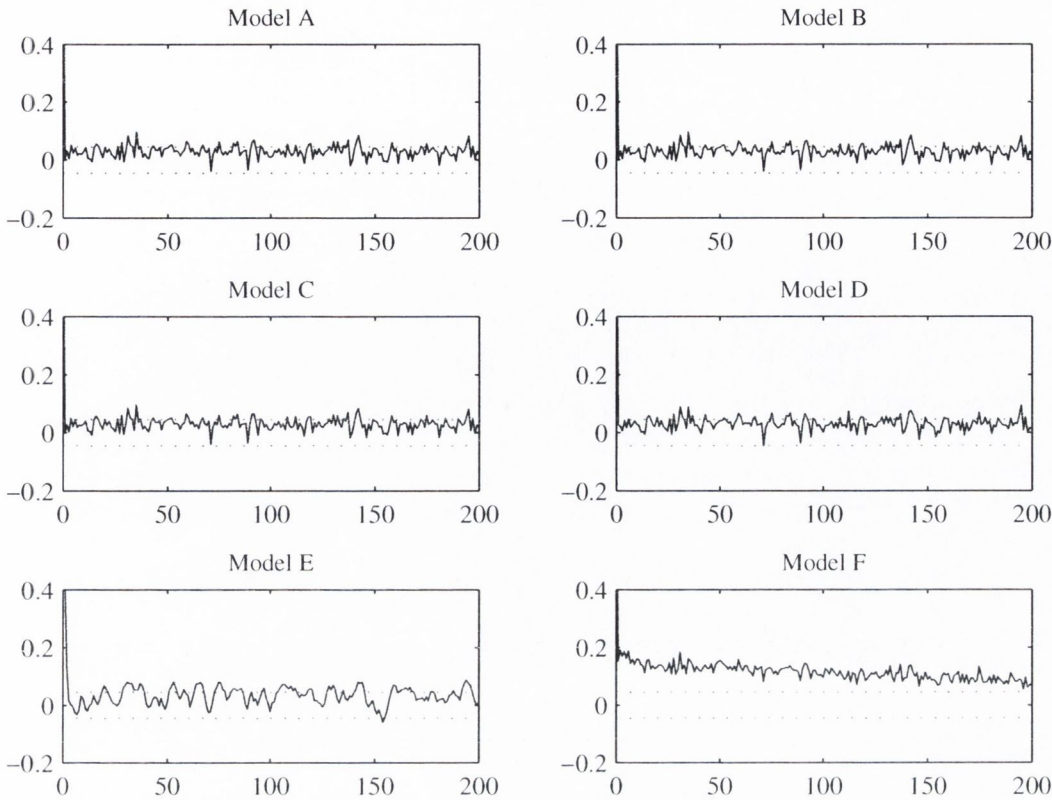




**Figure 7.7:** The autocorrelation function series over 200 lags for  $z - \bar{z}$  for the 6 selected models. The horizontal axis for each sub plot shows the lag and vertical axis relates to the sample autocorrelation.

a test for the independence of the members of the Pareto front post training is sufficient. Only very large changes in the *NLPD* score of the models results in a significant change in the autocorrelation function of the models. To demonstrate this Figure 7.7 plots the autocorrelation function sequences with confidence bounds for  $z - \bar{z}$ . Where  $z$  represents the PIT values (see equation 3.1) and  $\bar{z}$  is the mean of the PIT values. As the figure shows, Models A, B, C and F have comparable correlograms with little or no obvious correlations. Marginally worse than these models is the correlogram of Model D with some correlations in the first few lags. Model E has very strong correlations. This highlights the fact that this model is, in effect, constantly estimating the unconditional distribution.

Figure 7.8 shows the  $(z - \bar{z})^2$  autocorrelation function series for the 6 selected models. Models A, B, C and D perform comparably. There appears to be some weak serial correlations in this series. However, all models perform comparably. The autocorrelation series produced by Model E and F, however, are considerably worse showing persistent correlations. This indicates that these models have omitted some strong conditional volatility



**Figure 7.8:** The autocorrelation function series over 200 lags for  $(z - \bar{z})^2$  for the 6 selected models.

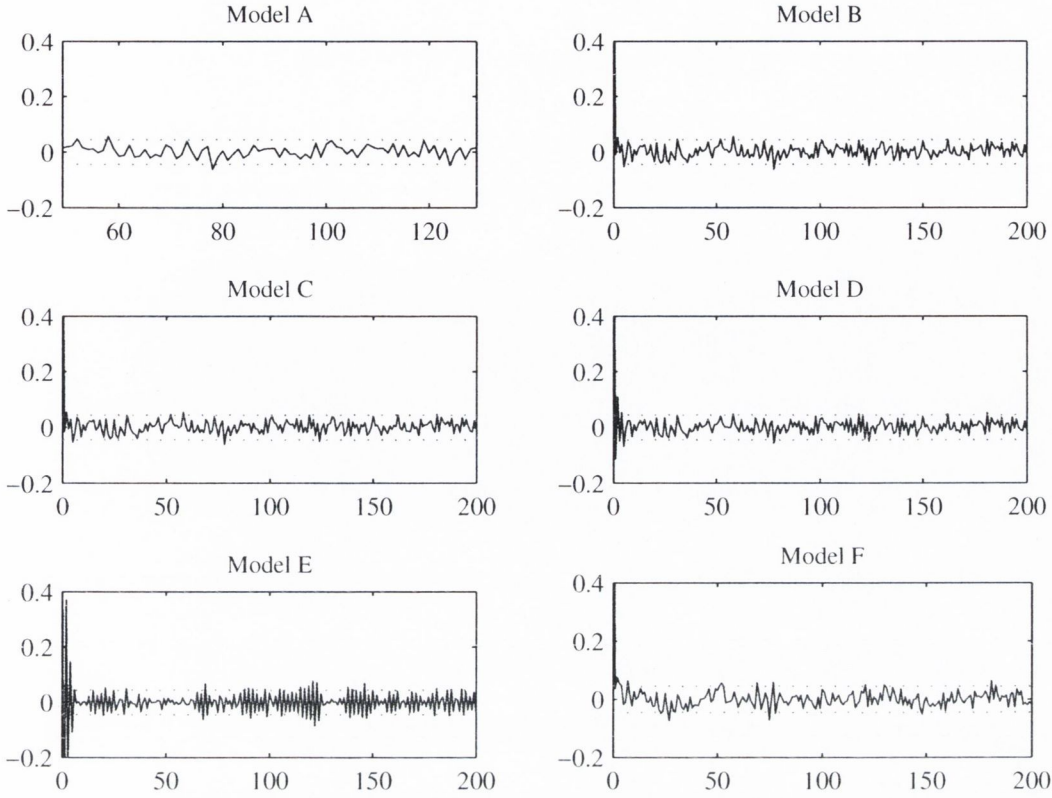
**Table 7.2:** Quantile exceedances for the selected models in and out-of-sample.

| Model | In Sample |       |        | Out-of-Sample |       |        |
|-------|-----------|-------|--------|---------------|-------|--------|
|       | 1.00%     | 5.00% | 10.00% | 1.00%         | 5.00% | 10.00% |
| A     | 1.30%     | 3.66% | 7.12%  | 1.02%         | 2.89% | 5.34%  |
| B     | 1.34%     | 4.04% | 7.69%  | 1.02%         | 3.08% | 5.72%  |
| C     | 1.56%     | 4.45% | 8.08%  | 1.30%         | 3.69% | 5.91%  |
| D     | 1.78%     | 4.96% | 9.06%  | 1.46%         | 3.97% | 6.45%  |
| E     | 1.40%     | 5.31% | 9.92%  | 1.18%         | 4.07% | 7.38%  |
| F     | 4.36%     | 9.03% | 13.90% | 4.42%         | 7.92% | 11.07% |

dynamics. The autocorrelation function series for  $(z - \bar{z})^3$  is shown in Figure 7.9. The results are similar to the  $(z - \bar{z})$  series. Figure 7.10 shows the  $(z - \bar{z})^4$  series, again Models A, B, C and D have few or no correlations and Models E and F contain persistent correlations.

#### 7.4.2 Value-at-Risk Evaluation

Finally, we look at the performance of these models on the chosen application Value-at-Risk. Table 7.2 shows the quantile exceedance for the 1st, 5th and 10th percentiles for the



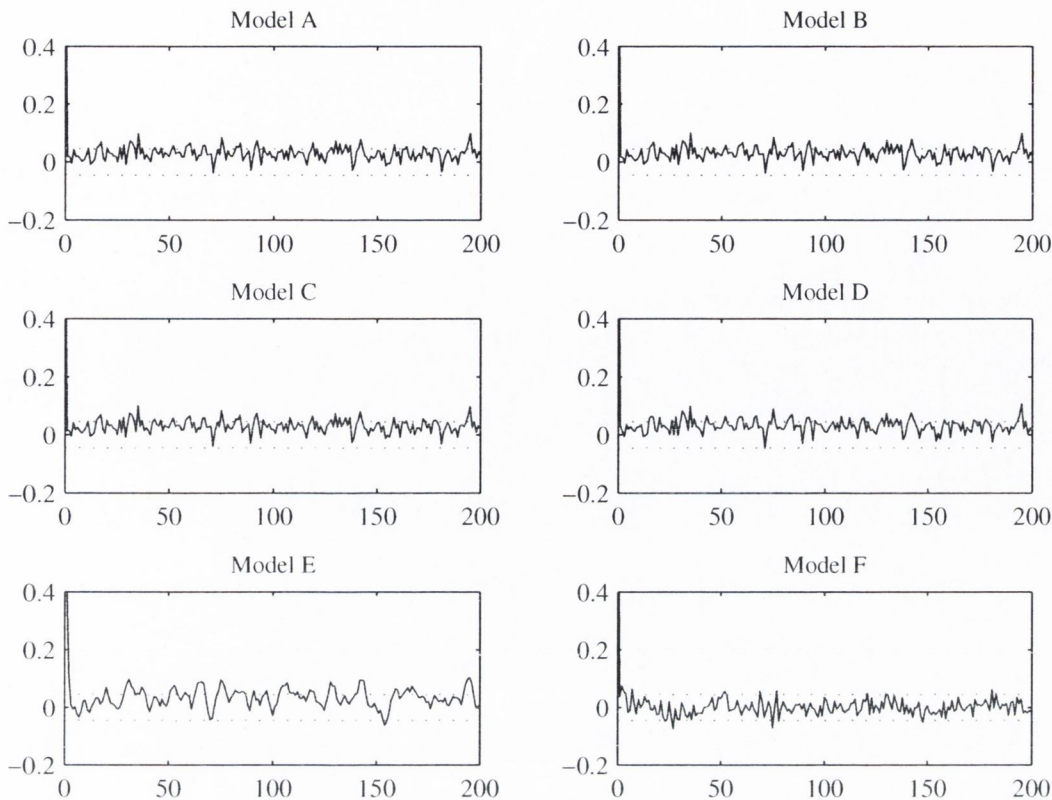
**Figure 7.9:** The autocorrelation function series over 200 lags for  $(z - \bar{z})^3$  for the 6 selected models.

6 selected models on the in and out-of-sample data. On average the best calibrated model across the chosen percentiles is Model E. This is to be expected given that Model E has the lowest  $A^2$  score. Also, it can be seen that there is a direct correlation between the  $A^2$  score of a model (position on the Pareto front) and the accuracy at estimating the percentiles of the model. Furthermore, the ranking of models on in-sample data is the same as the out-of-sample in terms of quantile accuracy. Given that Model E does not have an *i.i.d.* set of PIT values, the next best model would typically be chosen. In this case it would be either Model C or D. However, in this experiment, models were selected in order to clearly demonstrate the differences between models on the Pareto front rather than to optimise Value-at-Risk. A better model selection mechanism that selected models based on their accuracy estimating the percentiles of interest would result in better Value-at-Risk results.

### 7.4.3 Discussion

The results above show how the Pareto optimisation method presented in this thesis is capable of achieving its goal of optimisation on both sharpness and calibration. The





**Figure 7.10:** The autocorrelation function series over 200 lags for  $(z - \bar{z})^4$  for the 6 selected models.

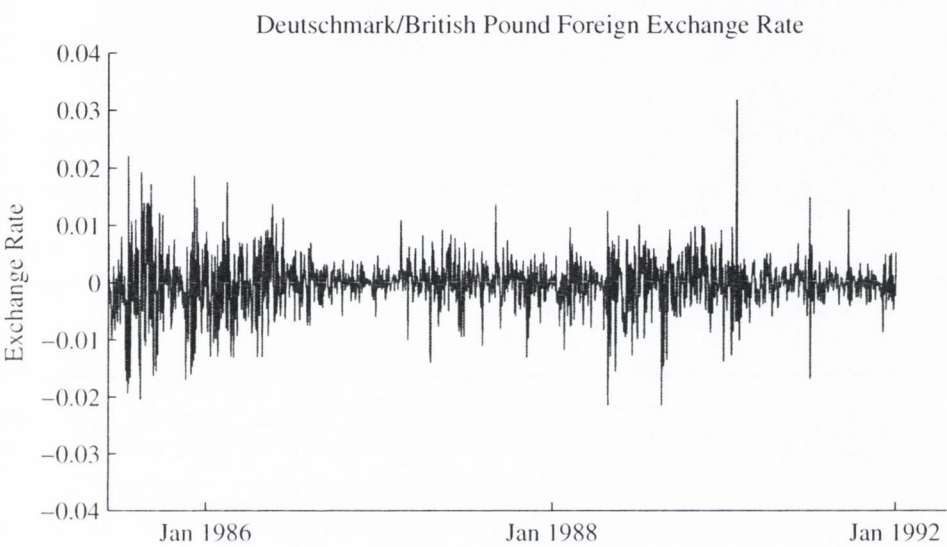
conflicting nature of  $NLPD$  and  $A^2$  suggested in Chapter 5 is shown and confirmed by virtue of the fact that a Pareto front is found. The Standard Model is optimal in terms of  $NLPD$ , however, the Pareto front shows that taking a small decrease in  $NLPD$  can result in big improvements in  $A^2$  and hence calibration. This is an important finding as the purpose of density forecasting is to produce accurate probabilistic estimates.

The *CRPS* Model, included in this evaluation performs poorly in comparison to the Standard Model and the selected Pareto Models. The *CRPS* Model produces estimates that are extremely over-confident i.e. the model has greatly underestimated the number of extreme events. The reason for this over-confidence being present in the *CRPS* models and not the  $NLPD$  model is most likely a consequence of the fact that the *CRPS* score does not penalize outliers as harshly as the  $NLPD$  score. Hence, the characteristically fat-tailed distribution of financial data causes the *CRPS* to underestimate the number of returns in the tails of the predicted density and results in a U shaped *PIT* histogram. Perhaps the *CRPS* model would perform better if a T distribution were predicted; however, a closed form expression for the *CRPS* for a T distribution is not available and numerical

estimation of the *CRPS* during optimisation is prohibitively slow.

Analysis of the autocorrelation of the *PIT* values suggests that significant gains can be made in terms of calibration without creating dependencies in the series. Awareness of the fact that well calibrated models can in fact be simply estimating the unconditional density is important; however, the *NLPD* score of a model should provide sufficient indication when this occurs.

## 7.5 Experiment 2: Pareto GARCH vs Pareto MDN

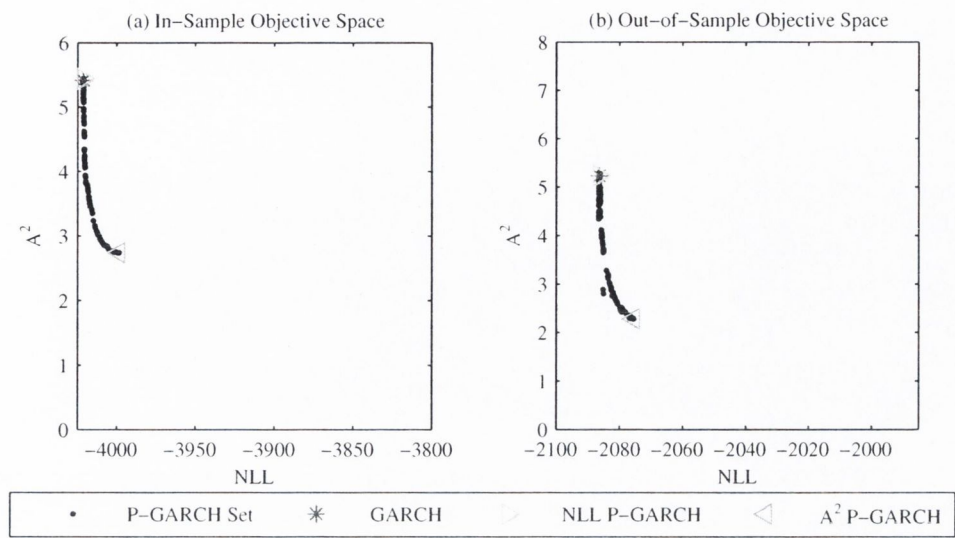


**Figure 7.11:** DEM/GBP exchange rate over the period 10 April 1985 to 1 January 1992. Test set highlighted in gray.

The aim of this experiment is to compare the results of four optimisation methods Standard GARCH, Pareto GARCH, Standard MDN and Pareto MDN. To do this we have chosen a foreign exchange series. Foreign exchange rates are difficult to predict due to the innumerable influencing factors that cause the exchange rate to fluctuate between two currencies. In this experiment, 1,501 daily Deutsche Mark (DEM)/British Pound (GBP) exchange rate closing prices for working days from the 10 April, 1985 to 1 January, 1992 are used <sup>3</sup>. This is a particularly volatile period for these two currencies that includes the major events of the 1987 stock market crash and the absorption of the East German Ostmark into the Deutsche Mark on 3 October, 1990. The return series was separated into

<sup>3</sup>This data is included with the *Mathworks<sup>TM</sup> Matlab<sup>TM</sup> Garch Toolbox* or at [www.oanda.com/convert/fxhistory](http://www.oanda.com/convert/fxhistory)

a training set of the first 1,000 days and a test set comprised of the last 500 observations. Figure 7.11 is a plot of this series.



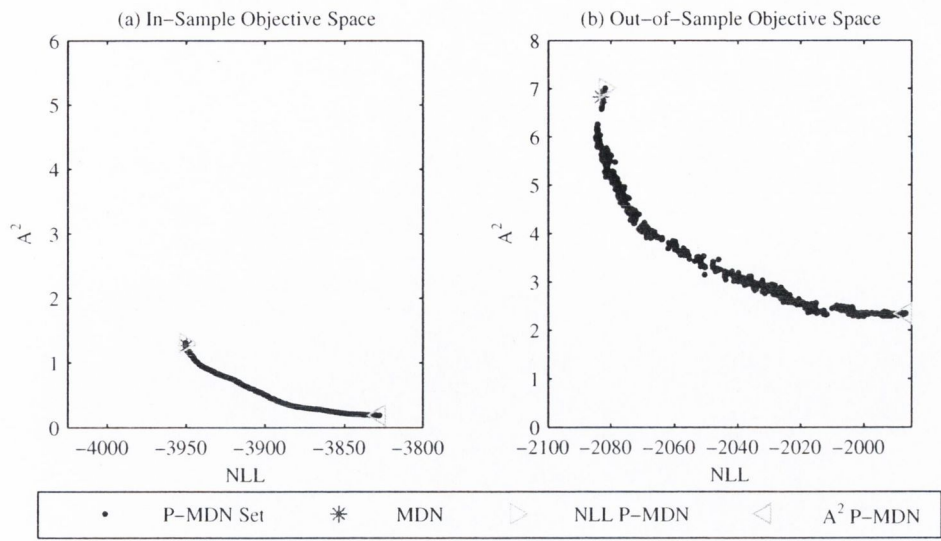
**Figure 7.12:** (a) In-sample and (b) out-of-sample objective spaces after 10,000 iterations of Pareto-GARCH

A Standard GARCH(1.1) model assuming a normal distribution was trained. The Pareto-GARCH optimisation procedure was then run for 1,000 iterations resulting in 316 non-dominated individuals. Figure 7.12 shows the in-sample and out-of-sample objective spaces for each of the models in the Pareto set. We have highlighted some models of interest, the *GARCH* model represents the initial solution trained using the standard optimisation procedure. *NLPD P-GARCH* is the model that has the best *NLPD* score and *A<sup>2</sup> P-GARCH* is the model with best *A<sup>2</sup>* score from the Pareto front.

For comparison, we also trained an MDN model on the same data. The MDN was given 6 hidden units and outputs were represented as a 2 component GMM. These parameters were chosen after a number of parameter selection experiments were carried out on the training data. An initial model was trained on the data using Scaled Conjugate Gradient method (Moller, 1993). This model was used as an initial starting position in parameter space for the evolutionary algorithm. The resulting objective spaces, on both training and test data, after 10,000 epochs of training are shown in Figure 7.13. The Pareto front has 736 individuals.

We have determined the Spearman rank correlation coefficients between the in-sample and out-of-sample models on each objective function. Both for the GARCH Pareto set and for the MDN Pareto set the model rankings are strongly correlated suggesting little





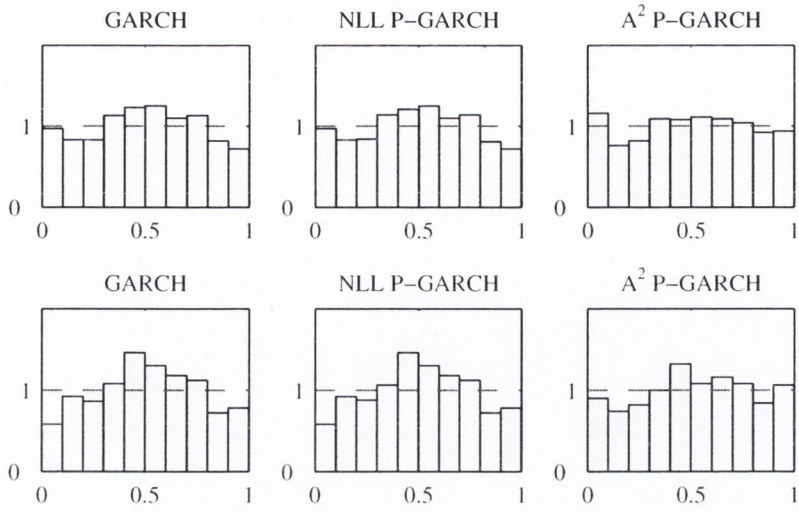
**Figure 7.13:** (a) In-sample and (b) out-of-sample objective spaces after 10,000 iterations of Pareto-MDN

**Table 7.3:** Spearman rank correlation coefficients between the in and out-of-sample objective function values for each model.

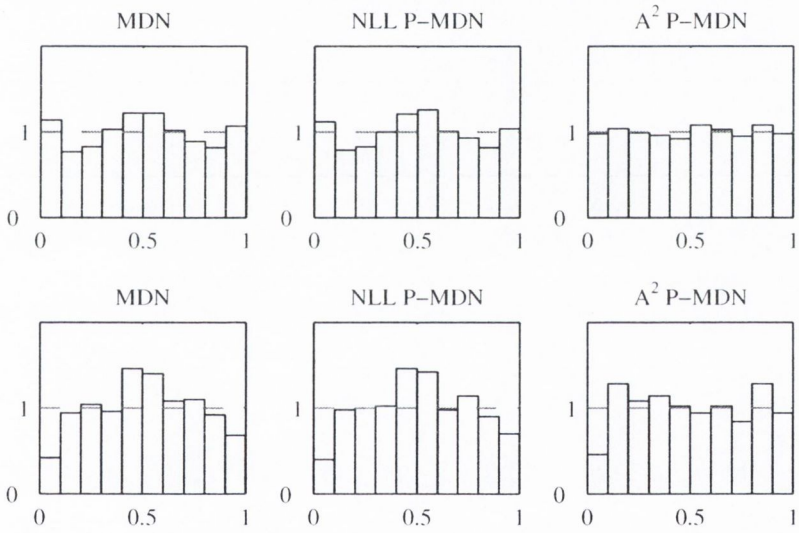
|                | <i>NLPD</i> | $A^2$  |
|----------------|-------------|--------|
| <i>P-GARCH</i> | 0.9967      | 0.9943 |
| <i>P-MDN</i>   | 0.8589      | 0.9848 |

over-fitting of the models to the data on either objective function. Table 7.3 shows the correlations.

The in-sample and out-of-sample MDN objective spaces (Figure 7.13) are scaled so that they can be compared easily with the GARCH objective spaces (Figure 7.12). Also, as in the GARCH objective space, the MDN models with best *NLPD* and  $A^2$  scores are highlighted in the objective space by the same marking scheme. The in-sample objective space figures suggest that the MDN produces a far better calibrated set of solutions on the in-sample data. This is confirmed by the PIT histograms shown in Figure 7.15 and Figure 7.14, where the calibration of the selected models can be seen for the in-sample data. The  $A^2$  *P-MDN* model shows almost perfect calibration. Again, this shows us that the multi-objective optimisation has achieved a degree of success on the in-sample data at least. Fortunately, this success is also present in the test data where the same effect can be seen. Interestingly, however, the simpler GARCH models outperform the MDN models considerably in the out-of-sample data. Again, from the PIT histograms it can be confirmed that the GARCH models, although suffering slightly from under-confidence due to the assumption of normality, produce superior models on both the *NLPD* and



**Figure 7.14:** GARCH pit histograms for the in-sample data (top row) and out-of-sample data (bottom row). The dashed line represents the Uniform distribution.



**Figure 7.15:** MDN pit histograms for the in-sample data (top row) and out-of-sample data (bottom row).

$A^2$  scores. Analysis of the dominance of solutions shows that in-sample the MDN and GARCH models produce solutions that do not dominate each other. However, out-of-sample there are 70 dominant solutions, out of the possible 1,052, 69 of these solutions are GARCH models.

**Table 7.4:** DEM/GBP quantile estimates. Comparison of quantile accuracies. Percentages represent the percentage of times the quantile was actually exceeded. Best results are underlined.

| Quantiles                          | 0.10%        | 1.00%        | 5.00%        | 10.00%       |
|------------------------------------|--------------|--------------|--------------|--------------|
| In Sample                          |              |              |              |              |
| GARCH(1,1)                         | 0.80%        | 2.20%        | 6.19%        | <u>9.69%</u> |
| <i>NLL</i> P-GARCH(1,1)            | 0.80%        | 2.20%        | 6.19%        | <u>9.69%</u> |
| <i>A</i> <sup>2</sup> P-GARCH(1,1) | 0.90%        | 2.60%        | 6.89%        | 11.59%       |
| MDN                                | 0.00%        | <u>1.20%</u> | <u>4.90%</u> | 11.09%       |
| <i>NLL</i> P-MDN                   | 0.00%        | <u>1.20%</u> | <u>4.90%</u> | 11.09%       |
| <i>A</i> <sup>2</sup> P-MDN        | <u>0.10%</u> | <u>1.20%</u> | 4.30%        | <u>9.69%</u> |
| Out of Sample                      |              |              |              |              |
| GARCH(1,1)                         | 0.60%        | <u>1.20%</u> | 3.00%        | 5.80%        |
| <i>NLL</i> P-GARCH(1,1)            | 0.60%        | <u>1.20%</u> | 3.00%        | 5.80%        |
| <i>A</i> <sup>2</sup> P-GARCH(1,1) | 0.60%        | 1.40%        | <u>4.20%</u> | <u>9.00%</u> |
| MDN                                | 0.00%        | 0.40%        | 2.00%        | 3.80%        |
| <i>NLL</i> P-MDN                   | 0.00%        | 0.40%        | 2.00%        | 3.60%        |
| <i>A</i> <sup>2</sup> P-MDN        | <u>0.10%</u> | 0.60%        | 1.80%        | 4.60%        |

### 7.5.1 Value-at-Risk Evaluation

Finally, we assess the effect that all these models have on the quantile accuracy for different potential VaR estimates. Table 7.5.1 shows the empirical percentage quantile exceedance rate for the percentiles 0.1%, 1%, 5% and 10%. These percentiles were chosen because they are the ones most commonly reported with VaR. The table compounds what is shown in the pit histograms above. In-sample the *A*<sup>2</sup> *P-MDN* has the best performance overall, however, out-of-sample the *A*<sup>2</sup> *P-GARCH* has the best performance in VaR estimation terms.

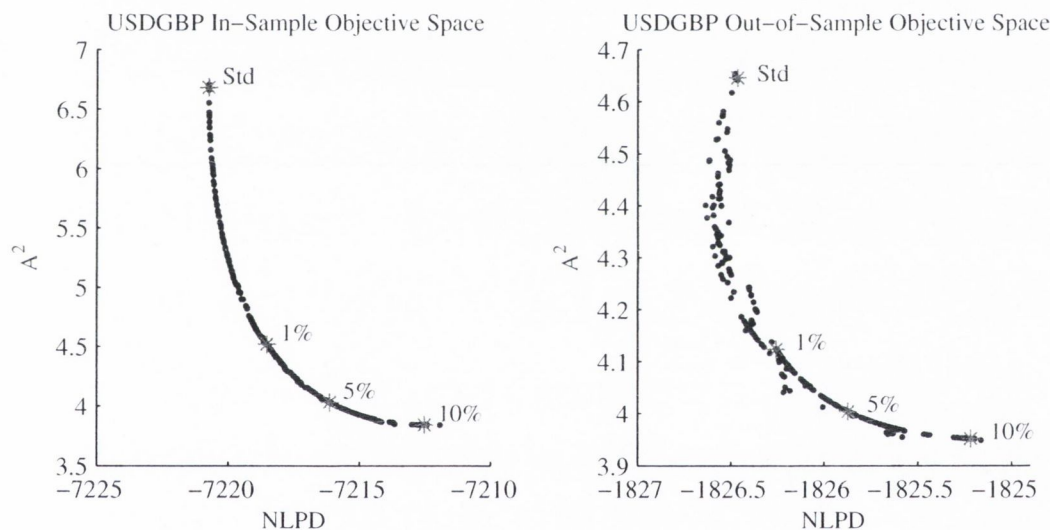
### 7.5.2 Discussion

In this experiment a comparison of MDN and GARCH on a finance application is presented. The results show that the much more flexible MDN models can successfully describe the training data. However, the MDNs tend to perform poorly out-of-sample with the much more simple GARCH models out performing them. This suggests that the MDN is over-fitting the training data in some way and that the model it produces does not generalise well. Schittenkopf et al. (2000) and Schittenkopf and Dorffner (2001) suggest several ways of adapting the basic MDN model to improve predictions out-of-sample.



## 7.6 Experiment 3: Sliding Window Analysis

The objective of this experiment is to show how the Pareto approach to optimisation can consistently return better GARCH models when compared against the Standard Model. To do this we have taken a series of 7,000 daily closing prices for the US Dollar (USD)/British Pound (GBP) exchange rate from the period 1 August, 1979 to 30 May, 2006. The data was transformed into log returns and divided into 10 training and test sets. Each training set had 2,000 contiguous returns and each test set consisted of the following 500 contiguous returns. Starting at the first observation on the 1 August, 1979 the first training and test set consisted of the first 2,500 returns. For the second training and test set the starting point is moved forward 500 days and the next 2,500 returns are used for training and testing. This process of historical analysis is called a *sliding window* and results in no overlapping of test data.

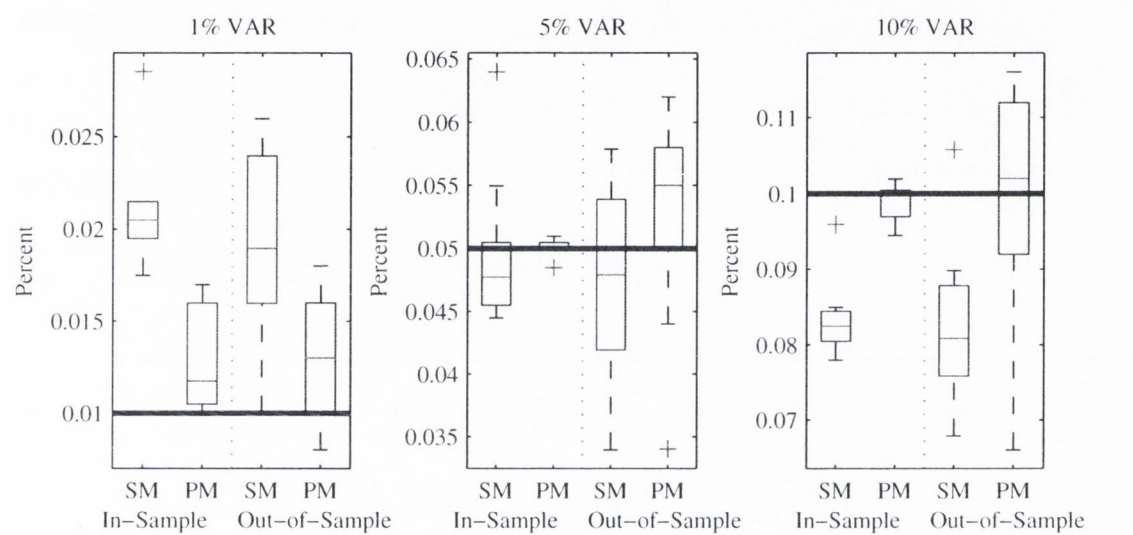


**Figure 7.16:** The left plot depicts the Pareto front obtained after training. Also highlighted are; the Standard Model (Std); the Pareto front model selected for estimating the 1st percentile (1%); 5th percentile (5%), and 10th percentile (10%). The plot on the right shows where these models are positioned in the out-of-sample objective space.

### 7.6.1 Value-at-Risk Evaluation

To evaluate the results of this experiment we take Value-at-Risk as our application and aim to find the models that will give the best estimate of the 1st, 5th and 10th percentiles. A GARCH(1,1) model with a Student t distribution is used as the base model in each

experiment<sup>4</sup>. The Standard Model is used as our benchmark result. A Pareto GARCH(1,1) model with T distribution is trained on each data set for 1,000 epochs. In this experiment we use a different selection mechanism to the IBM experiment above. To select models from the Pareto front we determined the models that had percentile exceedances closest to the actual percentile for each of the three VaRs. For example, for prediction of the 1% exceedance we took the model that had the most accurate 1% exceedance rate on the in-sample data. Figure 7.16 is an example Pareto front from one of the 10 produced during training. The selected models are highlighted in the in and out-of-sample objective spaces.



**Figure 7.17:** Shown above are box plots of the results of selected models across ten runs of the USDGBP data. In each plot the in-sample results are shown on the left and the out-of-sample results are shown on the right. The target percentile is marked by a heavy horizontal line. SM refers to the Standard Model results and PM refers to the Pareto Model results. With the exception the out-of-sample result for the 5% VaR the Pareto Models perform better than the Standard Model.

Figure 7.17 shows three box plots of the results for each of the VaR percentiles over 10 runs. The in-sample and out-of-sample results are compared. The results show how the Pareto models can consistently produce better percentile estimates than the standard GARCH optimisation approach. Table 7.5 summarises these plots.

<sup>4</sup>It is not possible to compare a *CRPS* model in this case due to the fact that we are assuming a Student t distribution.

**Table 7.5:** A summary of results shown in Figure 7.17. The average percentile exceedance and standard deviation over the ten runs are shown.

|          | In Sample    |              |              | Out of Sample |              |              |
|----------|--------------|--------------|--------------|---------------|--------------|--------------|
| Model    | 1.00%        | 5.00%        | 10.00%       | 1.00%         | 5.00%        | 10.00%       |
| Standard | 2.10%        | 4.95%        | 8.32%        | 1.90%         | 4.81%        | 8.18%        |
|          | $\pm 0.0029$ | $\pm 0.0059$ | $\pm 0.0052$ | $\pm 0.0051$  | $\pm 0.0076$ | $\pm 0.0109$ |
| Pareto   | 1.29%        | 5.01%        | 9.89%        | 1.32%         | 5.24%        | 9.94%        |
|          | $\pm 0.0028$ | $\pm 0.0007$ | $\pm 0.0022$ | $\pm 0.0034$  | $\pm 0.0084$ | $\pm 0.0161$ |

### 7.6.2 Discussion

In this experiment the ability of our Pareto Model to produce consistently better quantile estimates over the Standard Model is demonstrated. Over ten runs, the multi-objective strategy for optimisation and selection performs substantially better than the Standard Model for the percentiles selected. The ability of the approach described in this thesis to optimise density forecasting models that estimate distributions other than a Gaussian is also shown here. The framework described is sufficiently general to be quickly and easily adapted to optimise most density forecasting models assuming any distribution type.



## Chapter 8

# Medical Decision Support: Anticoagulant Drug Therapy

### 8.1 Introduction

The supervised learning methodology from machine learning research is very useful in medical decision support as it facilitates the development of prediction systems from historic data. The notion of ‘supervision’ evokes the idea of a learner who associates class labels to training examples much like a doctor may associate a set of symptoms to a disease type. These labelled examples can be used to train a prediction system that will be able to assign class labels to unlabelled examples.

There are a number of problems that exist within the medical domain that are well suited to a regression solution. For example;

- Glycated Haemoglobin (HbA1C) levels for diabetes patients (Braatvedt et al., 1997).
- Life expectancy for children with cystic fibrosis (Aurora et al., 2000).
- Life expectancy for cancer patients (Vigano et al., 1999).
- The life-span of artificial hip-joints (Young et al., 1998).
- International Normalized Ratio (INR) values for patients on anticoagulant therapy (Ginsberg et al., 2001; Byrne, 2002)

A variety of techniques exist for building prediction models for regression problems such as those outlined above. In addition to the classical linear regression models from statistics

(Weisberg, 2005), artificial neural networks are also well suited to problems of this type. However, a particular problem with numeric predictions is that precise numeric predictions produce estimates that give the user a false impression of perfect accuracy. For example, a neural network may estimate a patient's body temperature for Friday of next week as  $37.17384627^{\circ}\text{C}$ . If a practitioner were to act on this prediction without taking into account that there is a large variance in the possible outcomes they might put the patient's health at risk. For this reason *point* predictions are often avoided by discretising the outcome variable into a series of class labels, e.g.  $\{< 37.8^{\circ}\text{C}, \geq 37.8^{\circ}\text{C}\}$ . These issues of imprecision provide a strong case for producing probabilistic forecasts in this domain as the uncertainty of outcomes is of immense concern to both medical practitioners and patients.

Another serious problem exists in the situation where a point prediction is within some acceptable range but in reality there is a significant likelihood of a completely unacceptable outcome. Consider a situation where a prosthesis is predicted to last for 12 years but there is a 15% chance of it failing within 2 years. By predicting a complete probability distribution this risk can be highlighted and the risks and benefits of a specific course of action can be fully explored by the practitioner.

In this chapter a probabilistic model is used to estimate dosages for a blood thinning drug called Warfarin. Dosage prediction of Warfarin is particularly difficult because the range of acceptable blood thickness is narrow and maintaining a patient's blood thickness within this range is a difficult task. However, the evaluation section shows how using a well optimised density forecasting model will provide the best results in terms of mitigating patient risk.

## 8.2 Anticoagulant Drug Therapy

Warfarin is the only oral anticoagulant licensed for use in the Republic of Ireland. The indication for its use in many patients is to reduce the embolic risk associated with diseases such as atrial fibrillation, left ventricular dysfunction, deep vein thrombosis and mechanical aortic valve replacement (Byrne, 2002). While a patient is receiving therapy with an oral anticoagulant, careful monitoring and adjustment of their dosage is necessary so that the length of time it takes for their blood to clot is maintained within a predefined range, known as the International Normalised Ratio (INR) range (Sachdev et al., 1999). Giving a patient too low a dose increases their risk of developing a clot while giving a patient too

**Table 8.1:** The fifteen attributes compiled for each case in our input set.

|                       |                                  |
|-----------------------|----------------------------------|
| Age                   | Compliance                       |
| Weight                | Previous Warfarin dosage         |
| Height                | Measured INR                     |
| Marital Status        | Duration of therapy              |
| Prescribed Medication | Target INR value                 |
| Gender                | Hours since last Warfarin dosage |
| Smoking               | Any adverse events               |
| Alcohol               |                                  |

high a dosage exposes them to potential bleeding complications (Scottish Intercollegiate Guidelines Network, 1999). The current method of determining the dosage of Warfarin required by a particular patient is on a trial and error basis. The reason for this approach is that every patient will respond to Warfarin differently as each patient’s pharmacological response is affected by many exogenous factors other than the dosage prescribed, some of which are listed in Table 8.1. Section 8.3 describes how MDNs can be used for estimating anticoagulant drug dosages and Section 8.4 describes how the density forecasting models could be used in practice.

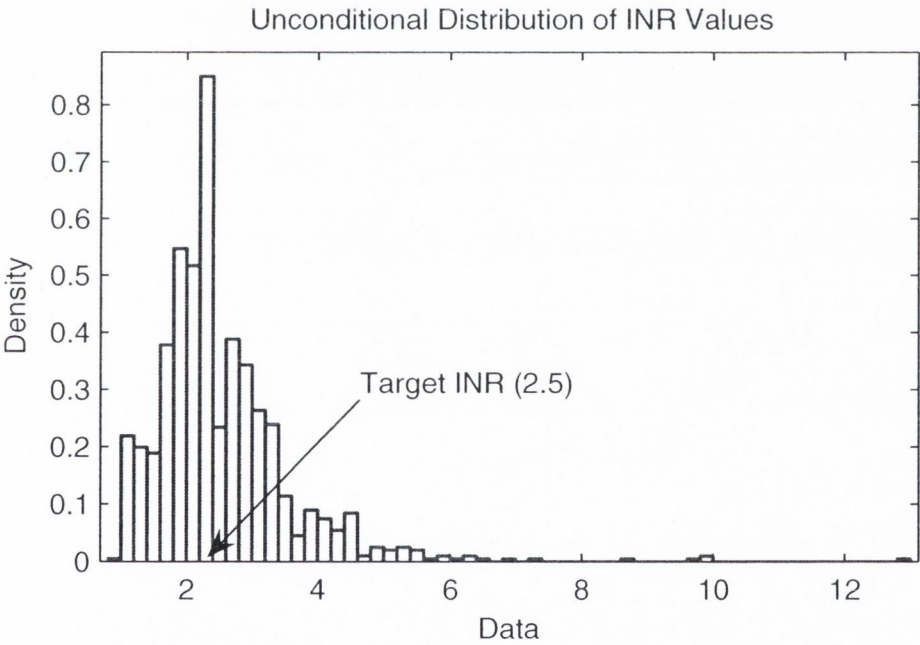
The data were collected from 145 patients undergoing anticoagulant therapy. The target variable is the patient’s observed INR value. The Target INR value for all patients in our data set is 2.5 (Target INR range of 2.0-3.0). If the patient’s INR measurement drops below 1.5, they are said to be at an unacceptable risk of clotting and if a patient’s INR rises above 4.5 they are exposed to an increased risk of bleeding. Fifteen attributes were recorded for each patient each time his/her INR value was measured. Table 8.1 provides a summary of these attributes. For each patient there are approximately 7 complete readings. The complete Warfarin dataset consists of 1,006 instances of patient attributes and resulting INR values. The data is grouped by patient so that cross-validation of model performance requires predictions on patients that were not observed in the model training set. Figure 8.3 is a histogram of the INR values for the 1,006 instances.

8.3 Experiment

Three different mixture density network models were selected for comparison. These were; *Standard MDN* optimised using the typical Scaled Conjugate Gradient approach; the Mixture Density Network (MDN) that achieves optimum sharpness in the Pareto MDN



front denoted *NLPD Pareto MDN*; and the MDN model that achieves optimum calibration in the Pareto MDN front denoted *A<sup>2</sup> Pareto MDN*. To determine the architecture of the MDN a Standard MDN model was trained using ten-fold cross-validation on the first training set of the experiment. The architecture that performs best out-of-sample over all ten folds is used as the starting architecture in all subsequent experiments. In this case, the architecture used had 5 hidden units and a 2 Gaussian Mixture Model. The Standard MDN was trained until convergence or until it completed 5,000 epochs. Algorithm 2 was used for training of the Pareto MDN model. Therefore each training set was split into a 90% partition of training and a 10% partition for validation. The Standard MDN was used as the initial starting point for the search algorithm. The algorithm was then run for 10,000 epochs. The parameters for the optimisation process are summarised in Table 6.2. The results in the following tables are based on average scores across the ten-folds of cross-validation.



**Figure 8.1:** A histogram of INR values of the 1,006 cases in the data set. Every case in this dataset has been assigned a target INR value of 2.5. It is clear that there is a skew in the data with the majority of cases centered around the mode, 2.5. There is a fat, right tail with values stretching out as far as 13. The left tail, however, only extends as far as 0.5. Physicians are interested in cases that are on the tails of the distribution, as these values represent patients that are outside their therapeutic index and are at risk of bleeding or clotting.

**Table 8.2:** This is a table of estimated risk of the patient exceeding specific INR values determined from the predicted distributions shown in Figure 8.4. This should be used in conjunction with the distributions and not as a substitute.

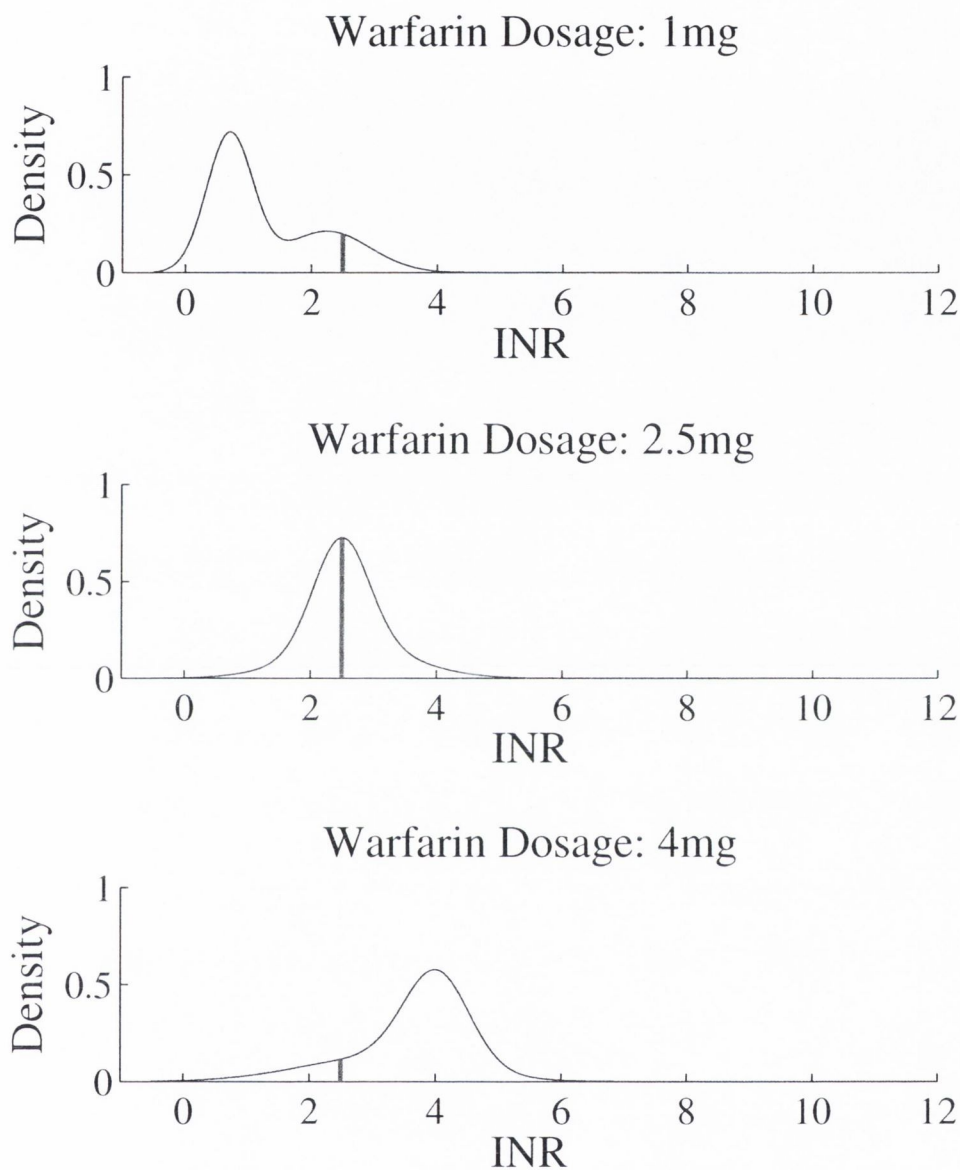
| Dosage | $\text{INR} \leq 1.5$ | $1.5 < \text{INR} < 4.5$ | $\text{INR} \geq 4.5$ |
|--------|-----------------------|--------------------------|-----------------------|
| 1 mg   | 68.1%                 | 31.8%                    | 0.1%                  |
| 2.5 mg | 4.6%                  | 94.4%                    | 1.0%                  |
| 4 mg   | 3.8%                  | 80.6%                    | 15.6%                 |

## 8.4 Predicting Density Functions for Anticoagulant Drug Therapy

In this section it is shown how the Pareto MDN model can be used to aid a physician when prescribing Warfarin. The physician’s objective is to prescribe the dosage that maximises the likelihood of the patient’s INR value being 2.5 at their next visit. An example of how our model can help a physician achieve this goal follows. The target INR value is based on the patient’s disease state and is specified by the prescribing physician. Guidelines regarding specific ranges are endorsed by the British Society for Haematology (1998).

Given a patient’s attributes (see Table 8.1), a physician can use the Pareto MDN system to estimate the affect a Warfarin dosage will have on a patient. This can be achieved by carrying out a series of “What if?” scenario tests. For practical reasons the physician must decide which dosage of Warfarin is best for his patient from the tablet sizes available. For example, a physician may wish to test the affect a 1mg, 2.5mg or a 4mg dosage will have on his patient’s INR. To achieve this the physician notes the current state of the patient in terms of the attributes in Table 8.1, along with the most recent INR value. The prediction model is updated with this information. Then, the physician can proceed with scenario testing different dosages. Figure 8.4 shows the typical result from a scenario test. Here, a random out-of-sample patient is selected to model the possible effects of different dosages, and the the input feature “Previous Warfarin Dosage” is changed according to the values that need to be tested. The figure shows the affect different dosages have on the predicted INR distributions for that patient. The model used in this example was the Pareto MDN that scored best in terms of calibration on the validation set. The predicted probability distributions clearly show that a prescription of 2.5mg will greatly reduce the patient’s risk of both clotting and bleeding.

It is possible to summarise some of the most important aspects of the predicted dis-



**Figure 8.2:** This figure shows the predicted INR distributions for a patient under different Warfarin dosages. The target INR for the patient is 2.5 (bold line). Three scenarios are presented; the predicted INR distribution if the patient is prescribed (top) 1mg Warfarin (middle) 2.5mg Warfarin (bottom) 4mg Warfarin.



**Table 8.3:** Results on the out-of-sample data over 10-fold cross-validation on the Warfarin data set. The *NLPD* model represents the members from the non-dominated set that maximised sharpness. The  $A^2$  models represents the members that maximised calibration. The average hit rate error over the percentiles 0.01, 0.05, 0.1, 0.9, 0.95, 0.99 is used. These results confirm the Pareto-MDN’s ability to optimise density forecasts under multiple objectives.

|                        | <i>NLPD</i>            | $A^2$                  | <i>Hit Rate</i>         |
|------------------------|------------------------|------------------------|-------------------------|
| Standard MDN           | 1.106<br>±0.151        | 2.577<br>±3.772        | 0.0834<br>±0.030        |
| <i>NLPD</i> Pareto MDN | <b>0.945</b><br>±0.114 | 1.300<br>±0.709        | 0.0782<br>±0.032        |
| $A^2$ Pareto MDN       | 1.001<br>±0.094        | <b>1.279</b><br>±0.970 | <b>0.0722</b><br>±0.030 |

tributions by presenting a table of the risks of the patient’s INR moving outside their therapeutic index. (see Table 8.2). This table simply displays the percentage of probability that the predicted distribution gives to the patient’s INR value entering levels that are seen as dangerous i.e. less than 1.5 and more than 4.5. This additional information is most useful in borderline cases where it is difficult to distinguish the difference between two probability distributions.

### 8.5 Evaluation

Table 8.3 shows comparative results on both the *NLPD* and  $A^2$  cost functions for three model types. Performance of the Pareto MDN is benchmarked against the Standard MDN over 10-fold cross-validation. Not surprisingly, the *NLPD* Pareto MDN model achieves the best sharpness score and  $A^2$  Pareto MDN model achieves the best calibration score on average.

The unconditional distribution of INR values is shown in Figure 8.1. (Mac Namee et al., 2002) showed the effect of bias in this dataset and suggested a technique to improve results using an ensemble of point predictors. By maximising calibration, it is possible to correct the bias in the predicted distributions, resulting in the tails of the predicted distributions on a calibrated model being correctly calibrated. As the tail predictions are of more interest, this is an important improvement to the predictions. Improvements in tail estimates can be assessed using the hit rate method described earlier. The hit rate column in Table 8.3 compares the quality of tail estimates over ten-folds for a selected few

percentiles. The  $A^2$  models perform best. This improvement is statistically significant in comparison to both the *NLPD* and the Standard MDN models when assessed using a two-tail  $t$ -test ( $p = 0.05$ ).

### 8.5.1 Comparative Evaluation

In this comparative evaluation set the classification accuracy of a number of different prediction approaches is compared against the probabilistic approach described in this thesis. The dataset has been converted into a classification problem by labelling cases that have an INR of less than or equal to 1.5 as **clotting**, those with an INR greater than or equal to 4.5 as **bleeding** and patients with an INR between these two values as **normal**. This results in 835 cases of normal, 124 of clotting and 47 of bleeding. Table 8.4 below compares the classification accuracy for each class over 10-fold cross-validation for each of the following classifiers.

- C4.5 - decision tree algorithm (Quinlan, 1993).
- Logistic Regression - multinomial logistic regression model with a ridge estimator (Witten & Frank, 2005).
- Naive Bayes - simple probabilistic classifier (Witten & Frank, 2005).
- kNN - k nearest neighbour classifier (Doyle et al., 2005).

A threshold probability for each class is decided upon based on the percentage contribution of each class to the unconditional distribution of the training data. A predicted density function produced by an MDN model is converted into a classification of patient state by determining the predictive density for each class and selecting the class that exceeds its threshold.

Also shown in Table 8.4 are the specificity and sensitivity of the classifiers. Specificity relates to the proportion of people that have an unsafe INR value and the classifier has made this prediction. Sensitivity is the proportion of people that have a safe INR value and the classifier has predicted they are safe. To calculate the specificity and sensitivity of the classifiers the number of classes are reduced from 3 to 2. This was achieved by combining **bleeding** and **clotting** into a new **warning** class and leaving the **normal** class as it was. This approach simplifies the problem, but the results are useful to demonstrate the effect of skewed data on classifiers in standard medical evaluation terms. The large difference in



**Table 8.4:** The columns **clotting**, **normal** and **bleeding** represent the class accuracies for each model on the Warfarin classification dataset. The **specificity** and **sensitivity** of each of the classifiers is also shown. Percentages are based on the average value over 10-fold cross-validation on the out-of-sample data.

| Classifier             | normal | clotting | bleeding | Specificity | Sensitivity |
|------------------------|--------|----------|----------|-------------|-------------|
| C4.5                   | 95.2%  | 29.0%    | 12.8%    | 24.9%       | 95.0%       |
| Logistic Regression    | 97.4%  | 27.4%    | 23.4%    | 26.5%       | 97.2%       |
| Naive Bayes            | 92.2%  | 10.5%    | 17.0%    | 13.0%       | 91.2%       |
| kNN (k=1)              | 91.3%  | 21.8%    | 10.6%    | 19.5%       | 90.5%       |
| kNN (k=3)              | 97.0%  | 9.7%     | 6.4%     | 8.9%        | 96.7%       |
| Standard MDN           | 76.9%  | 71.0%    | 72.3%    | 76.7%       | 75.8%       |
| <i>NLPD</i> Pareto MDN | 76.6%  | 77.4%    | 74.5%    | 82.4%       | 75.6%       |
| $A^2$ Pareto MDN       | 79.6%  | 78.2%    | 76.6%    | 83.1%       | 78.6%       |

specificity and sensitivity highlights the imbalance in the classifier techniques. The MDN models perform substantially better than any other model in terms of specificity. This comes at the price of a small deterioration in sensitivity. However, it could be rationally argued that specificity is a more important metric to succeed in than sensitivity in this prediction task. In future work it would be interesting to apply the Neyman-Pearson to aid model selection.

Table 8.4 clearly shows that the skew in the data creates a bias that causes very poor classification accuracy for the minority classes in the non-MDN classifiers, but extremely high prediction accuracy in the majority class (**normal**). This suggests that the classifier bases its predictions on the trivial assumption that the majority class is more likely than the minority class. This is a result of the very poor specificity of these models. Fortunately, this weakness is not present in the density forecasting models that have corrected for skewed data by thresholding predictions with the unconditional distribution of the training data. The Pareto MDN shows a consistent accuracy across each class showing that it has overcome the bias problem. Among the MDN models the  $A^2$  Pareto MDN model is the best performing across all classes and in terms of both sensitivity and specificity, highlighting the importance of calibration.

The consistency of a calibrated model allows the physician to be more confident when using this system in practice. Combining this with a well calibrated probability distribution estimate of the INR further enhances the insight the user gets from the predictive model. By clearly displaying the uncertainty in the predictions, density forecasting miti-



gates some of the concerns of using artificial neural networks in the medical domain and provides an attractive alternative to the more traditional prediction techniques.

## Chapter 9

# Conclusions

### 9.1 Introduction

This thesis has investigated a major field of machine learning, regression. In particular, the ability of density forecasting models to produce sharp and calibrated predictions in regression tasks was addressed.

In Chapter 2 the problem was formally introduced and the most popular methods for density forecast modelling were reviewed. The relationship between Maximum Likelihood and negative log-likelihood was explained and two model types, MDN and GARCH were chosen for experimentation throughout the thesis. In Chapter 3 the objectives of density forecasting, sharpness and calibration, were described in detail. Various methods of evaluation of density forecasts were presented and three methods were selected; NLPD, CRPS and PIT. Chapter 4 reviewed the various optimisation methods available when density forecasting. Evolutionary Algorithms emerged as a suitable approach, specifically a multi-objective evolutionary strategy.

Chapter 5 provided examples of the most common difficulties when optimising density forecasting models. For example, over-fitting of the conditional variance, sensitivity to outliers and bias toward sharp rather than calibrated models. Once the problems were identified, a path to a solution could be found. A new quality score called the  $A^2$  score based on the Anderson-Darling test for uniformity was motivated and demonstrated to work successfully as a calibration score. This calibration score and the popular NLPD score were shown to be conflicting objectives near the global optimal of models. Once this

had been confirmed a multi-objective framework for optimisation of density forecasting models could justifiably be described and used.

Chapter 6 provided a detailed description of the proposed algorithm. Implementation details were also provided to develop an MDN model that could be optimised on both Sharpness (NLPD) and Calibration ( $A^2$ ). The resulting model was called a Pareto MDN. Also, a Pareto GARCH model was fully described that implemented the proposed optimisation method. Initial experimentation on synthetic data showed how the proposed method could successfully achieve its goal of producing a Pareto front of solutions varying in sharpness and calibration.

Chapter 7 was the first of two evaluation chapters. A risk management application called Value-at-Risk was used as the problem space for this chapter. Three evaluations were presented. First the proposed Pareto GARCH approach was compared against the standard GARCH optimisation algorithm and a new CRPS GARCH model that uses the CRPS quality score during training instead of NLPD. The results from this experiment show that the Pareto GARCH model finds the best set of solutions on the criteria of sharpness and calibration and this was confirmed through evaluation in terms of Value-at-Risk. The second evaluation compared the Pareto GARCH and Pareto MDN models on another VaR problem. Interestingly, the Pareto MDN outperforms the Pareto GARCH in-sample, but the model was found to have over-fitted the training data and perform poorly out-of-sample. The third example shows how the Pareto GARCH model can consistently out-perform the standard GARCH model. This was achieved by applying a sliding window analysis of the proposed technique and comparing results with the standard model.

Finally, Chapter 8 presented density forecasting as a solution to a medical decision problem. The Pareto MDN model was compared with the standard MDN model and a number of classification techniques typically used in medical decision making. A calibrated Pareto MDN model was shown to perform better in terms of specificity and sensitivity than any other model tested on an INR prediction problem. A scenario testing application for doctors was suggested that would aid practitioners and patients find the best Warfarin dosage more quickly.

## 9.2 Thesis Contributions

The findings of this research can be summarised as follows:



- Density forecast calibration can be optimised using a new calibration score called  $A^2$ .
- Calibration and sharpness are conflicting objectives as the density forecasting model nears a global optimum in one or other of these objectives.
- Due to the conflicting nature of calibration and sharpness a reformulation of the optimisation is necessary to optimise on both objectives simultaneously and produce a Pareto set of optimal solutions.
- This proposed optimisation approach can be implemented for a Mixture Density Network and a GARCH model.
- Density forecasts in applications in Finance and Medical Decision Making can be improved through the use of the proposed algorithm.

## 9.3 Future Work

### 9.3.1 Dominance Uncertainty

Within multi-objective optimisation the dominance of models is deterministic. This is a harsh method of discarding models as often the difference in model accuracies are very small and if the nuances of the particular training set were only slightly different the dominance decision would be reversed. It would be interesting to address this problem by assessing the probability of dominance and maintaining an archive of solutions which are, with some known probability, mutually non-dominating. Fieldsend and Everson (2005) examine methods for estimating the probability of dominance. This could potentially improve the quality of the archive of models returned after training.

### 9.3.2 Model Uncertainty

It should be noted that density forecasting is an estimation of the complete uncertainty for the given prediction task. There exists another layer of uncertainty that may not be captured by the model, the uncertainty in the prediction itself - we call this model uncertainty. This whole other layer of uncertainty has not yet been addressed to any great degree in the literature. However, there has been some research carried out for parametric approaches where the underlying distributional assumption is treated as correct, in this

case the effect of model uncertainty error can be estimated and reported as a point-by-point confidence interval for the density forecast (Sodelind & Svensson, 1997). It would be interesting to further develop the method described in this thesis to incorporate a measure of model uncertainty into the framework.

### 9.3.3 Different Objective Functions

In this thesis NLPD and  $A^2$  were used as the objective functions of choice due to their conflicting nature and ease of evaluation. However, there is the possibility of adding additional objective functions, for example, a quantile error score could be incorporated in the search process to tune model predictions for a specific quantile. Alternatively, a combined independence test could be added or different problem specific calibration or sharpness scores could be used. The flexibility of the evolutionary algorithm means that there are innumerable different possibilities for new objective functions.

## 9.4 Concluding Remarks

Density forecasts are becoming more and more prevalent in our daily lives. For example, important macro-economic estimates such as interest rate movements are now being presented to the public in terms of density forecasts. With increased use and acceptance of density forecasts the challenge of improving the quality of these estimates is greater now more than ever. This thesis has contributed to the research on density forecasting by showing that the objectives of sharpness and calibration of a density forecast are not always complimentary but instead are often conflicting. This issue has been addressed by developing a new optimisation approach that simultaneously optimises both objective functions to return a set of trade-off solutions. The proposed approach is described in sufficiently general terms that it can be adapted to optimise most density forecast modelling methods for regression. Empirical analysis on two problem domains using two different density forecasting models validates the proposed approach and demonstrates its benefits.

# Bibliography

- Abbass, H. (2001). A memetic pareto evolutionary approach to artificial neural networks. *Australian Joint Conference on Artificial Intelligence*, 1, 1–11.
- Adrian E. Raftery, David Madigan, J. A. H. (97). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92, 179–191.
- Aldrich, J. (1997). R.A. Fisher and the making of maximum likelihood 1912-1922. *Statistical Science*, 12, 162–176.
- Andersen, T., & Bollerslev, T. (1988). Answering the skeptics: Yes, standard volatility models do provide accurate forecasts. *International Economic Review*, 39, 885–905.
- Anderson, T., & Darling, D. (1954). A test of goodness of fit. *Journal of the American Statistical Association*, 19, 765–769.
- Aurora, P., Wade, A., Whitmore, P., & Whitehead, B. (2000). A model for predicting life expectancy of children with cystic fibrosis. *European Respiratory Journal*, 16, 1056–1060.
- Back, T. (1996). *Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms*. Oxford Univ. Press.
- Baillie, R., & Bollerslev, T. (1989). The message in daily exchange rates: a conditional variance tale. *Journal of Business and Economic Statistics*, 7, 297–305.
- Bartlmae, K., & Rauscher, F. (2000). Measuring DAX market risk: A neural network volatility mixture approach. *Presentation at the FFM2000 Conference, London*.
- Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematics and Statistics*, 41, 164–171.



- Berkowitz, J. (2001). Testing density forecasts, with applications to risk management. *Journal of Business & Economic Statistics, American Statistical Association*, 19, 465–474.
- Bishop, C. M. (1994). *Mixture density networks* (Technical Report). Neural Computing Research Group, Aston University.
- Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press, Inc.
- Bishop, C. M., & Qazaz, C. S. (1996). Bayesian inference of noise levels in regression. *International Conference on Artificial Neural Networks, ICANN '96*.
- Bollerslev, T. (1986). Generalized autoregressive conditional hetroskedasticity. *Journal of Econometrics*, 31, 307–327.
- Bollerslev, T. (1987). A conditionally heteroskedastic time series model for speculative prices and rates of return. *Review of Economics and Statistics*, 69, 542–547.
- Braatvedt, G., Drury, P., & Cundy, T. (1997). Assessing glycemic control in diabetes: Relationships between fructosamine and HbA1c. *New Zealand Medical Journal*, 110, 459–62.
- Brier, G. W. (1950). Verification of forecasts epressed in terms of probability. *Monthly Weather Review*, 78, 1–3.
- British Society for Haematology (1998). Haemostasis and thrombosis task force. guidelines on oral anticoagulation: Third edition. British Journal of Haematology.
- Brockwell, P. J., & Davis, R. A. (1991). *Time series: Theory and methods 2nd edition*. Springer, New York.
- Brooks, C., Burke, S., & Persand, G. (2001). Benchmarks and the accuracy of GARCH mode estimation. *Journal of Forecasting*, 17, 45–56.
- Broomhead, D. S. ; Lowe, D. (1988). *Radial basis functions, multi-variable functional interpolation and adaptive networks*. ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN.

- Byrne, T. S. (2002). *Warfarin therapy in the Republic of Ireland: The potential role of neural networks in optimising therapy*. Doctoral dissertation, Trinity College Dublin.
- Candela, J., Rasmussen, C., Sinz, F., Bousquet, O., & Scholkopf, B. (2006). Evaluating predictive uncertainty challenge. *MLCW LNAI*, 3944, 1–27.
- Carney, J., Cunningham, P., & Bhagwan, U. (1999). Confidence and prediction intervals for neural network ensembles. *International Joint Conference in Neural Networks*.
- Carney, M., Cunningham, P., Dowling, J., & Lee, C. (2005). Predicting probability distributions for surf height using an ensemble of mixture density networks. In *Proceedings of the 22nd International Conference in Machine Learning* (pp. 113–121).
- Charalambous, C. (1992). Conjugate gradient algorithm for efficient training of artificial neural networks. *IEE Proceedings-G*, 139, 301–311.
- Chatfield, C. (1993). Calculating interval forecasts. *Journal of Business and Economic Statistics*, 11, 121–144.
- Christoffersen, P., & Jacobs, K. (2004). Which GARCH model for option valuation? *Management Science*, 50, 1204–1221.
- Christoffersen, P. F. (1998). Evaluating interval forecasts. *Intl. Econ. Review*, 39, 841–862.
- Coello, C. (1998). *A comprehensive survey of evolutionary-based multiobjective optimization techniques* (Technical Report). Laboratorio Nacional de Informatica Avanzada.
- Cooke, W. (1906). Weighting forecasts. *Monthly Weather Review*, 34, 23–24.
- Cornford, D., Nabney, I. T., & Bishop, C. M. (1999). Neural network-based wind vector retrieval from satellite scatterometer data. *Neural Computing and Applications*.
- Cramer, J. S. (1986). *Econometric applications of maximum likelihood methods*. Cambridge University Press.
- Crnkovic, C., & Drachman, J. (1997). “Quality Control”, in VAR: Understanding and applying Value-at-Risk. In *Risk publications*. London.

- Das, I., & Dennis, J. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimisation*, 14, 63–69.
- Dawid, A. P. (1984). Statistical theory: The prequential approach. *Journal of the Royal Statistical Society A*, 147, 278–292.
- De Wick, O. L. (2005). *Multiobjective optimization: History and promise* (Technical Report). MIT.
- Deb, K. (2001). *Multi-objective optimisation using evolutionary algorithms*. Wiley.
- DeGroot, M. H., & Fienberg, S. E. (1983). The comparison and evaluation of forecasters. *The Statistician*, 32, 12–22.
- Diebold, F., Hahn, J., & Tsay, A. (1998a). *Real-time multivariate density forecast evaluation and calibration: Monitoring the risk of high-frequency returns on foreign exchange* (Technical Report). National Bureau of Economic Research.
- Diebold, F. X., Gunther, T. A., & Tay, A. S. (1998b). Evaluating density forecasts with applications to financial risk management. *International Economic Review*, 39, 863–83.
- Doyle, D., Loughrey, J., Nugent, C., Coyle, L., & Cunningham, P. (2005). Fionn: A framework for developing CBR systems. *Expert Update*, 8, 11–14.
- Drucker, H. (1997). Improving regressors using boosting techniques. *ICML* (pp. 107–115).
- Duffie, D., & Pan, J. (1997). An overview of value at risk. *The Journal of Derivatives*, 5, 7–49.
- Edgeworth, F. (1881). *Mathematical psychics*. Augustus M Kelley Pubs.
- Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap*. Chapman Hall, London.
- Engel, R. (2001). GARCH 101: The use of ARCH/GARCH models in applied econometrics. *The Journal of Economic Perspectives*, 15, 157–168.
- Engle, R. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation. *Econometrica*, 50, 987–1008.



- Engle, R., & Manganelli, S. (1999). *CAViaR: Conditional value at risk by quantile regression* (Technical Report). MP Page - NBER, Working Paper.
- Engle, R. F. (2002). Dynamic conditional correlation: a new simple class of multivariate GARCH models. *Journal of Business and Economic Statistics*, 20, 339–350.
- Epstein, E. (1969). A scoring system for probabilities of ranked categories. *Journal of Applied Meteorology*, 8, 985–987.
- Faussett, L. (1994). *Fundamentals of neural networks*. Prentice Hall.
- Fieldsend, J., & Everson, R. (2005). Multi-objective optimisation in the presence of uncertainty. *International Conference on Evolutionary Computation*.
- Fieldsend, J., Everson, R. M., & Singh, S. (2003). Using unconstrained elite archives for multiobjective optimization. *IEEE Trans. Evolutionary Computation*, 7, 305–323.
- Fieldsend, J., & Singh, S. (2002). Pareto multi-objective non-linear regression modelling to aid CAPM analogous forecasting. *IEEE International Conference on Computational Intelligence for Financial Engineering*.
- Fieldsend, J., & Singh, S. (2005). Pareto evolutionary neural networks. *IEEE Trans. Neural Networks*, 16, 338–354.
- Fieldsend, J. E. (2003). *Novel algorithms for multi-objective search and their application in multi-objective evolutionary neural network training*. Doctoral dissertation, University of Exeter.
- Finch, S., Mendell, N., & Thode, H. (1989). Probabilistic measures of adequacy of a numerical search for a global maximum. *Journal of the American Statistical Association*, 84, 1020–1023.
- Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution*. Wiley Publishing, New York.
- Franses, P., & Ghijsels, H. (1999). Additive outliers, GARCH and forecasting volatility. *International Journal of Forecasting*, 15, 1–9.
- Fraser, A. M., & Dimitriadis, A. (1993). Forecasting probability densities by using hidden Markov models with mixed states. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison Wesley.

- Friedman, J. H. (1997). On bias, variance, 0/1loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1, 55–77.
- Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Computation*, 4, 1–58.
- Ginsberg, J., Crowther, M., White, R., & Ortel, T. (2001). Anticoagulant therapy. *Hematology (American Society of Hematology Education Program)* (pp. 339–57).
- Glosten, L., Jagannathan, R., & Runkle, D. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *Journal of Finance*, 48, 1779–1801.
- Gneiting, T., Balabdaoui, F., & Raftery, A. (2005a). *Probabilistic forecasts, calibration and sharpness* (Technical Report). Department of Statistics, University of Washington.
- Gneiting, T., & Raftery, A. (2004). *Strictly proper scoring rules, prediction, and estimation* (Technical Report 463). University of Washington.
- Gneiting, T., Raftery, A. E., Balabdaoui, F., & Westveld, A. (2003). Verifying probabilistic forecasts: Calibration and sharpness. *Proceedings Workshop on Ensemble Forecasting, Val-Morin, Quebec*.
- Gneiting, T., Raftery, A. E., Westveld, A. H., & Goldman, T. (2005b). Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review*, 133, 1098–1118.
- Goffe, W. L., Ferrier, G., & Rogers, J. (1994). Global optimization of statistical functions with simulated annealing. *Journal of Econometrics*, 60, 65–99.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison Wesley.
- Gonzalez, J., Rojas, I., Ortega, J., Pomares, H., Fernandez, F., & Daz, A. (2003). Multi-objective evolutionary optimization of the size, shape and position parameters of radial basis function networks for function approximation. *IEEE Transactions on Neural Networks*, 14, 1478–1498.
- Good, I. J. (1952). Rational decisions. *Journal of the Royal Statistical Society. Series B (Methodological)* (pp. 107–114). The MIT Press.

- Granger, C. W. J., & Newbold, P. (1986). *Forecasting economic time series*. Academic Press, New York.
- Hamill, T. M. (2000). Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, 129, 550–561.
- Herrera, F., Lozano, M., & Verdegay, J. L. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review*, 12, 265–319.
- Hersbach, H. (2002). Decomposition of the continuous ranked probability score for ensembles prediction systems. *Weather Forecasting*, 15, 559–570.
- Heskes, T. (1997a). Balancing between bagging and bumping. *Advances in Neural Information Processing Systems* (pp. 466–472).
- Heskes, T. (1997b). Practical confidence and prediction intervals. *Advances in Neural Information Processing Systems* (pp. 466–472). MIT Press.
- Higgins, M. L., & Bera, A. K. (1992). A class of nonlinear ARCH models. *International Economic Review*, 33, 137–158.
- Hodur, R. M. (1997). The naval research laboratorys coupled ocean/atmosphere mesoscale prediction system (COAMPS). *Monthly Weather Review*, 125, 1414–1430.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.
- Horn, J., & Nafpliotis, N. (1993). *Multiobjective optimization using the niched Pareto Genetic Algorithm* (Technical Report 930005). Illinois Genetic Algorithms Laboratory, University of Illinois.
- Hsieh, D. (1989). Testing for nonlinear dependence in daily foreign exchange rate changes. *Journal of Business*, 62, 339–368.
- Husmeier, D. (1999). *Neural networks for conditional probability estimation: Forecasting beyond point predictions*. Springer.
- Husmeier, D., & Taylor, J. (1998). Neural network for predicting conditional probability densities: Improved training scheme combining EM and RVFL. *Neural Networks*, 11, 89–116.



- Hyndman, R. J. (1995). Highest-density forecast regions for non-linear and non-normal time series models. *Journal of Forecasting*, 14, 431–441.
- Jacobs, R. A., Jordan, M. I., Nowlan, S., & Hinton, G. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3, 79–87.
- Jerrell, M. E., & Campione, W. A. (2001). Global optimization of econometric functions. *Journal of Global Optimization*, 20, 273–295.
- Jorion, P. (1995). Predicting volatility in the foreign exchange market. *Journal of Finance*, 50, 507–528.
- Jorion, P. (2001). *Value at risk: The new benchmark for managing financial risk. second edition*. McGraw-Hill.
- Koza, J. (1992). *Genetic programming: On the programming of computers by means of natural selection*. MIT Press, Cambridge, MA.
- Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22, 76–86.
- Kurihara, Y., Tuleya, R. E., & Bender, M. A. (1997). The GFDL hurricane prediction system and its performance in the 1995 hurricane season. *Monthly Weather Review*, 126, 1306–1322.
- Ledolter, J. (1989). The effect of additive outliers on the forecasts from ARIMA models. *International Journal of Forecasting*, 5, 231–240.
- Lo, J., & Bassu, D. (2002). Robust approximation of uncertain functions where adaptation is impossible. *Proceedings of IEEE International Joint Conference in Neural Networks, Hawaii* (pp. 1889–1894).
- Locarek-Junge, H., & Prinzler, R. (1997). Estimating value-at-risk using neural networks. *Application of Machine Learning and Data Mining in Finance*.
- Mac Namee, B., Cunningham, P., Byrne, S., & Corrigan, O. (2002). The problem of bias in training data in regression problems in medical decision support. *Artificial Intelligence in Medicine*, 24, 51–70.
- Malkiel, B. G. (1973). *A random walk down Wall Street*. W. W. Norton and Company.

- Mandelbrot, B. (2004). *Fractals and chaos: The Mandelbrot set and beyond*. Springer.
- Manganelli, S., & Engle, R. (2001). *Value at Risk models in finance* (Technical Report). European Central Bank.
- MathWorks Matlab (2002). *GARCH toolbox user's guide*. Version 2.0.1 edition.
- McCullough, B., & Renfro, C. (1998). Benchmarks and software standards: A case study of GARCH procedures. *Journal of Economic and Social Measurement*, 25, 59–71.
- McLachlan, G. (2000). *Finite mixture models*. Wiley.
- Miazhyńska, T., Dorffner, G., & Dockner, E. (2003). Risk management application of the recurrent mixture density network models. *ICANN/ICONIP*.
- Moller, M. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* (pp. 525–533).
- Molteni, F., Buizza, R., Palmer, T., & Petroliagis, T. (1996). The ECMWF ensemble prediction system: Methodology and validation. *Royal Meteorological Society*, 122, 73–119.
- Murphy, A., & Winkler, R. (1977). Reliability of subjective probability forecasts of precipitation and temperature. *Applied Statistics*, 26, 41–47.
- Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica*, 59, 347–370.
- Neuneier, R. (1995). Optimal strategies with density estimating neural networks. *ICANN '95*.
- Neuneier, R., Hergert, F., Finnoff, W., & Ormoneit, D. (1994). Estimation of conditional densities: A comparison of neural network approaches. *ICANN'94*. Springer.
- Nix, D., & Weigend, A. S. (1995). Learning local error bars for nonlinear regression. *Advances in Neural Information Processing Systems 7 (NIPS'94)* (pp. 489–496).
- Noceti, P., Smith, J., & Hodges, S. (2003). An evaluation of tests of distributional forecasts. *Journal of Forecasting*, 22, 447–455.

- Ormoneit, D. (1993). Estimating probability densities using neural networks. Master's thesis, Technische Universitat Munchen.
- Ormoneit, D., & Neuneier, R. (1996). Experiments in predicting the German stock index dax with density estimating neural networks. *The IEEE/IAFE 1996 Conference on Computational Intelligence*.
- Pareto, V. (1896). *Cours d'economie politique*. Lausanne.
- Park, B. (2002). An outlier robust GARCH model and forecasting volatility of exchange rate returns. *Journal of Forecasting*, 21, 381–393.
- Quinlan, J. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers Inc.
- Rasmussen, C. (2004). *Gaussian processes in machine learning* (Technical Report). Max Planck Institute for Biological Cybernetics.
- Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian processes for machine learning*. MIT Press.
- Rechenberg, I. (1994). *Evolutionsstrategie '94* (Technical Report).
- Richmond, K. (2001). Mixture density networks, human articulatory data and acoustic-to-articulatory inversion of continuous speech,. *In Proceedings Workshop on Innovation in Speech Processing WISP'2001*.
- Riskmetrics (1996). *Riskmetrics technical document* (Technical Report). Morgan Guaranty Trust Company of New York.
- Rosenblatt (1952). Remarks on a multivariate transformation. *Annals of Mathematics and Statistics*, 23, 470–472.
- Sachdev, G., Ohlrogge, K., & Johnson, C. (1999). Review of the fifth american college of chest physicians consensus conference on antithrombotic therapy; outpatient management of adults. *Am J Health-Syst Pharm*, 56, 1505–1514.
- Schervish, M. J. (1989). A general method for comparing probability assessors. *The Annals of Statistics*, 17, 1856–1879.



- Schittenkopf, C., & Dorffner, G. (2001). Risk-neutral density extraction from option prices: Improved pricing with mixture density networks. *IEEE Transactions on Neural Networks*, 12, 716–726.
- Schittenkopf, C., Dorffner, G., & Dockner, E. (2000). Forecasting time-dependent conditional densities: A semi-nonparametric neural network approach. *Journal of Forecasting*, 19, 355–374.
- Schittenkopf, C., Dorner, G., & Dockner, E. (1998). Volatility prediction with mixture density networks. *International Conference on Artificial Neural Networks*.
- Scholes, F. B. M. (1973). The pricing of options and corporate liabilities. *The Journal of Political Economy*, 81, 637–654.
- Schwefel, H. P. (1987). *Collective intelligence in evolving systems*. Berlin: Springer.
- Scottish Intercollegiate Guidelines Network (1999). Pregnancy and the puerperium. in: Antithrombotic therapy. SIGN Publication No. 35. Edinburgh: SIGN.
- Siegler, M. A., Jain, U., Raj, B., & Stern, R. M. (1997). *Automatic segmentation, classification, and clustering of broadcast news audio* (Technical Report). Carnegie Mellon University.
- Simons, K. (2000). The use of value at risk by institutional investors. *New England Economic Review*, Nov-Dec, 21–30.
- Sodelind, P., & Svensson, L. (1997). New techniques to extract market expectations from financial instruments. *Journal of Monetary Economics*, 40, 383–429.
- Specht, D. (1991). Generalized regression neural network. *IEEE Transactions on Neural Networks*, 2, 1.
- Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, 3, 109–118.
- Stine, R. A. (1985). Bootstrap prediction intervals for regression. *Journal of the American Statistical Association*, 80, 1026–1031.
- Talagrand, O., Vautard, R., & Strauss, B. (1997). Evaluation of probabilistic prediction systems. In *Proceedings, ECMWF Workshop on Predictability (20-22 October 1997)* 1 - 25.

- Taleb, N. (2001). *Fooled by randomness: The hidden role of chance in the markets and in life, first edition*. W. W. Norton & Company.
- Thombs, L. A., & Schucany, W. R. (1990). Bootstrap prediction intervals for autoregression. *Journal of the American Statistical Association*, 85, 486–492.
- Timmermann, A. (2000). Density forecasting in economics and finance. *Journal of Forecasting*, 19, 231–234.
- Veall, M. R. (1989). Applications of computationally-intensive methods to econometrics. *Bulletin of I.S.I., 47th Session* (pp. 75–88).
- Vigano, A., Dorgan, M., Bruera, E., & Suarez-Almazor, M. (1999). The relative accuracy of the clinical estimation of the duration of life for patients with end of life cancer. *Cancer*, 86, 170–176.
- Wallis, K. F. (2003). Chi-squared tests of interval and density forecasts, and the Bank of Englands fan charts. *International Journal of Forecasting*, 19, 165–175.
- Wang, Y., & Wahl, F. (1997). Multiobjective neural network for image reconstruction. *Vision, Image and Signal Processing, IEE Proceedings-*, 144, 233–236.
- Wei, W. W. S. (1990). *Time series analysis*. Addison-Wesley, Redwood City, Cal.
- Weigend, A., & Srivastava, A. (1995). Predicting conditional probability distributions: A connectionist approach. *International Journal of Neural Systems*, 6, 109–118.
- Weigend, A. S., & Nix, D. A. (1994). Predictions with confidence intervals (local error bars). *ICONIP'94 Seoul*.
- Weigend, A. S., & Shi, S. (2000). Predicting daily probability distributions of S&P500 returns. *Journal of Forecasting*, 19, 375–392.
- Weisberg, S. (2005). *Applied linear regression, 3rd edition*. Wiley.
- Whitely, D. (1996). Evaluating evolutionary algorithms. *Artificial Intelligence*, 85, 245–276.
- Williams, W., & Goodman, M. (1971). A simple method for the construction of empirical confidence limits for economic forecasts. *Journal of American Statistical Association*, 66, 752–754.

- Wilson, L. J., Burrows, W., & Lanzinger, A. (1999). A strategy for verification of weather element forecasts from an ensemble prediction system. *Monthly Weather Review*, 127, 956–970.
- Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques, 2nd edition*. Morgan Kaufmann.
- Yao, X. (1999). Evolutionary artificial neural networks. *Proceedings of the IEEE* (pp. 1423–1447).
- Yao, X., & Liu, Y. (1997). Fast evolution strategies. *Lecture notes in computer science*, 1213, 151–162.
- Yao, X., & Liu, Y. (1998). Towards designing artificial neural networks by evolution. *Applied mathematics and computation*, 91, 83–90.
- Young, N., Cheah, D., Waddell, J., & Wright, J. (1998). Patient characteristics that affect the outcome of total hip arthroplasty: a review. *Canadian Journal of Surgery*, 41, 188–95.
- Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3, 257–271.