

Local and Personalised Models for Prediction,
Classification and Knowledge Discovery on Real
World Data Modelling Problems

Yuan-Chun (Peter) Hwang

A thesis submitted to Auckland University of Technology in
fulfilment of the requirements for the degree of Doctor of
Philosophy

2009

School of Computing & Mathematical Sciences
Supervisors: Prof. Nikola Kasabov and Dr. Qun Song
Consultant: Prof. Robyn North

This thesis was proofread by Catriona Carruthers

TABLE OF CONTENTS

TABLE OF CONTENTS	ii
LIST OF FIGURES	vii
LIST OF TABLES.....	xi
ACKNOWLEDGEMENTS	1
ABSTRACT	3
CHAPTER 1 INTRODUCTION	6
1.1 Motivation and Objective.....	6
1.2 Main Contributions	7
1.3 Thesis Structure.....	8
CHAPTER 2 REAL WORLD DATA MODELLING ISSUES - A CRITICAL ANALYSIS	11
2.1 Evolving Problems	11
2.2 Unique Problem Subspaces	12
2.3 Noise.....	13
2.4 Outliers	15
2.5 Missing data.....	17
2.6 Imbalanced data	19
2.7 Relevance of Features.....	22
2.8 Conclusion	25
CHAPTER 3 MODELLING TECHNIQUES FOR COMPUTATIONAL INTELLIGENCE – A LITERATURE REVIEW	27
3.1 State of the Art Global Models	27
3.2 Local Models.....	30

3.3 Individualised (Personalised) Model through Transductive Reasoning.....	31
3.4 Statistical Methods.....	33
3.5 Artificial Neural Networks (ANN).....	38
3.6 Fuzzy Logic Systems.....	45
3.7 ANFIS: Adaptive-Network-Based Fuzzy Inference System	53
3.8 Generalisation Error Estimation and Model Selection.....	64
3.9 Conclusion.....	66
CHAPTER 4 DYNFIS – AN IMPROVED DYNAMIC NEURAL FUZZY INFERENCE SYSTEM FOR LOCAL MODELLING	68
4.1 Introduction.....	68
4.2 Algorithm Description.....	70
4.3 Knowledge Extraction	73
4.4 Benchmark Dataset Case Study: Mackey-Glass Dataset.....	73
4.5 Application of DyNFIS in Neural Network Forecasting Competition (NN3) for Time Series Prediction	76
4.6 Conclusion.....	79
4.7 Discussion	80
CHAPTER 5 MUFIS: A NOVEL NEURO-FUZZY INFERENCE SYSTEM USING MULTIPLE TYPES OF FUZZY RULES	81
5.1 Algorithm Description.....	81
5.2 Case Study and Analysis.....	86
5.3 Real world data modelling case study on renal function evaluation	90
5.4 Knowledge Discovery	93

5.5 Conclusion	96
5.6 Discussion	97
CHAPTER 6 INTEGRATED TEMPORAL AND SPATIAL MULTI-MODEL SYSTEMS.....	98
6.1 Algorithm.....	100
6.2 Example.....	104
6.3 Conclusion and Discussion.....	105
CHAPTER 7 THE APPLICATION OF THE MULTI-MODEL SYSTEM TO SOLVE A REAL WORLD TIME-SERIES DATA MODELLING PROBLEM	107
7.1 Data Description	107
7.2 Data Preparation Process.....	109
7.3 Weekly Pattern Analysis	117
7.4 Farm Zone Analysis.....	121
7.5 Data Smoothing	124
7.6 Multi-Model System (MMS).....	127
7.7 Experimental Results and Comparison.....	129
7.8 Conclusions	134
CHAPTER 8 PERSONALISED REGRESSION MODEL WITH INCREMENTAL FEATURE SELECTION.....	135
8.1 Algorithm.....	136
8.2 Feature Ranking	137
8.3 Incremental feature selection.....	138
8.4 Personalised regression Model.....	139
8.5 Error Measure.....	140

8.6 Conclusion	143
CHAPTER 9 PERSONALISED REGRESSION MODELS FOR PREGNANCY OUTCOME PREDICTION BASED ON SCOPE DATA	144
9.1 Problem Overview	144
9.2 Data Description	145
9.3 Imbalanced Dataset on Personalised Model.....	146
9.4 Method Application	147
9.5 Results.....	151
9.6 Chapter Conclusion and Discussion	153
CHAPTER 10 CONCLUSION AND FUTURE RESEARCH	156
10.1 DyNFIS – Dynamic Neuro-Fuzzy Inference System	156
10.2 MUFIS: A Novel Neuro-Fuzzy Inference System Using Multiple Types of Fuzzy Rules	157
10.3 Multi-Model System – Temporal and Spatial	157
10.4 Personalised Regression Model	158
10.5 Summary	159
10.6 Future Research	159
10.7 NeuCom.....	160
10.8 Future Publications	161
REFERENCES	162
LIST OF PUBLICATIONS	179
APPENDIX A NEUCOM - A NEURO-COMPUTING DECISION SUPPORT ENVIRONMENT	180
APPENDIX B INTRODUCTION TO SCOPE STUDY	187

Introduction	187
Pregnancy Problems	187
Prenatal Care Today	188
Predict to Prevent	188
Aims.....	188
Research	189
Clinical Dataset.....	190

LIST OF FIGURES

Figure 2.1 A single outlier in a group of 11 input vectors can significantly compromise a model.....	15
Figure 2.2 Illustration of low prevalence disease data in two dimensional space, 50 healthy patients, 5 patients with the disease.	20
Figure 2.3 Example of ROC curve. The more area under the curve the better the model	21
Figure 3.1 Illustration of a global model. A single linear function is used for all input vectors.....	28
Figure 3.2 Illustration of a local model consists of three sub-models.	30
Figure 3.3 Illustration of a personalised model. A model is created on the fly using a subset of training data near the test input vector (\mathbf{x})	32
Figure 3.4 Linear and non-linear model prediction on a noisy dataset.....	36
Figure 3.5 Weights are applied to the inputs and the weighted sum is then passed through a function to produce the final output y	38
Figure 3.6 An illustration of the structure of a single hidden layer multi-layer perceptron network.	40
Figure 3.7 A structure of a multi-layer perceptron with two hidden layers (bias omitted for clarity of the illustration).....	41
Figure 3.8 Two basic signal flows in a multi-layer perceptron.....	42
Figure 3.9 A three-layer back-propagation neural network with data feed forward through layers.	42
Figure 3.10 An illustration of the RBFN architecture with three radial basis functions for the prediction problems	44
Figure 3.11 Examples of fuzzy membership functions.....	46
Figure 3.12 A block diagram of a basic fuzzy inference system	52
Figure 3.13 Takagi-Sugeno fuzzy inference system with two rules. From Jane's paper for ANFIS (Jang, 1993)	53

Figure 3.14 Illustration of the ANFIS system. From Jang's ANFIS paper (Jang, 1993).....	53
Figure 3.15 Example of ECM Clustering algorithm. X_i : input vector (*), Cc_j^k : cluster centre, C_j^k : cluster, Ru_j^k cluster radius (Kasabov & Song, 2002).	59
Figure 3.16 Triangular membership function showing the three parameters a, b and c.	61
Figure 4.1 Triangular membership function replaced with Gaussian membership function	68
Figure 4.2 Mackey-Glass dataset with 500 input vectors for testing.	74
Figure 4.3 The values of the 11 time-series problems in the NN3 competition. Blue lines are the training data and red lines are the predicted values for t+1 to t+18.....	77
Figure 5.1 Number of ZM rules used in high variation regions of the problem space. Green line: test data's normalised output values, blue bar: number of ZM fuzzy rules used. This indicates that ZM fuzzy rules are used more often when the data contains high variation.	88
Figure 5.2 The location of fuzzy membership function centres and the patient in the Principle Component Analysis (PCA) space using the first two components. The prediction for the current patient is made using an integration of the three TS fuzzy rules and one ZM fuzzy rule.	95
Figure 6.1 Multi-Model System framework, provides multiple views of the problem and integrates the output from each model based on its previous prediction error. Modified from Kasabov's book on evolving connectionist system in 2007(Kasabov, 2007a).....	100
Figure 6.2 A block diagram of the TWNFI algorithm.	102
Figure 6.3 An example scenario of time-series pattern that changes over time. Red bar indicates the 53 rd time point.....	105
Figure 7.1 Shows an original set of data which was then processed to identify outliers using the process described below.....	114
Figure 7.2 Five-day moving average for the season from Figure 7.1.....	115

Figure 7.3 The difference between the original milking volume and the five-day moving average volumes (from Figure 7.1 and Figure 7.2).....	116
Figure 7.4 Daily milking volume in a season	117
Figure 7.5 Five-day moving average of the daily milk volume in a season	118
Figure 7.6 Differences between the daily milking volume and the five-day moving average volume.....	119
Figure 7.7 Autocorrelation values show the relationship between the “today” milk volume (indicated on the x-axis as 0) and previous days.....	119
Figure 7.8 Aggregated autocorrelation result of all farms and seasons. No significant average correlation is found, the highest being negative 0.3 at 5 days’ back.	121
Figure 7.9 Normalised milking volume for Zone 1 (Mean and standard deviation – red; Min-Max volumes – in blue)	122
Figure 7.10 Normalised milking volume for Zone 2.....	122
Figure 7.11 Normalised milking volume for Zone 3.....	123
Figure 7.12 Normalised milking volume for Zone 4.....	123
Figure 7.13 Pickup volume across seasons.....	125
Figure 7.14 Data smoothing illustration.....	125
Figure 7.15 Pickup volume data smoothed using the second-order linear smoothing method.....	126
Figure 7.16 The system architecture. The online model is self-improving through correction of the two models contribution weights toward the final output.	127
Figure 7.17 Absolute error from each model in MMS across one season. TWNFI and WRLSE performs very differently, each can be significantly better than the other at different time. The MMS framework allows better overall prediction accuracy to be achieved than using either system alone....	129
Figure 7.18 A comparison between the MMS and Linear Regression. Average MAE of 1 to 4 day-ahead prediction results on 12 randomly selected farms.	130

Figure 7.19 Predicted and actual daily production volumes of a random farm (3 rd season) (Dotted line: Actual Volumes; Solid line: MMS Predicted Volumes).....	130
Figure 7.20 ME of predicted volumes by MMS (the 3 rd season)	131
Figure 7.21 ME of predicted volumes by LR (the 3 rd season)	131
Figure 8.1 Synthetic data with two classes and two variables. Univariate analysis shows both variables have moderate discriminating power but when combined, the discriminating power increases significantly, copied from an overview of feature selection issues (Guyon & Elisseeff, 2003).....	137
Figure 8.2 Correlation plot between RMSE and AUC during training. Low RMSE can be translated to high AUC but the translation is not 1 to 1. Minor changes in RMSE may not affect AUC and vice versa.	142
Figure 9.1 Prediction accuracy at different threshold including class 1 accuracy (disease), class 2 accuracy (healthy), overall accuracy, ROC curve and area under the ROC curve.	152

LIST OF TABLES

Table 4.1 Prediction accuracy comparison of several offline algorithms on t+85 Mackey-Glass dataset.....	75
Table 5.1 Prediction results of off-line learning models on Mackey-Glass t+6 training and testing data.....	87
Table 5.2 RMSE comparison between various models on GFR dataset	92
Table 7.1 <i>Description of variables used to describe a milk pickup from a farm.</i>	108
Table 7.2 <i>Description of variables used to describe a farm.</i>	109
Table 7.3 The number of pickups affected by different type of exceptions	113
Table 7.4 Comparative analysis of the 1 to 4 days ahead daily prediction error between the proposed MMS and the currently used LR models on 12 farms' data (averaged)	131
Table 7.5 4-day ahead prediction error on 12 randomly selected farms' data	133
Table 7.6 Four days ahead prediction error (average) on 575 farms' data	133
Table 9.1 Result of the proposed personalised model.	151
Table 9.2 Results of the logistic regression model with AIC feature selection.	153

ACKNOWLEDGEMENTS

There is so much gratitude in my heart that I can never truly express it with mere words.

I would like to start by thanking my primary supervisor, Prof. Nikola Kasabov, for his support, supervision to my PhD study and for the opportunity to work on these interesting real world case studies.

I would especially like to thank my secondary supervisor, Dr. Qun Song. He was always available for discussions and has always provided me with guidance and friendship. He was there to push me over the bottlenecks in my research when a push was just what I needed. Through his guidance, I was introduced to fuzzy inference systems, which have become the foundation of my PhD study.

I would like to thank Prof. Robyn North, for her guidance and advices. Thanks to the rest of the SCOPE team, especially Prof. Allen Rodrigo and Dr. Mik Black, I have learned a lot and received much inspiration from them.

I must especially thank my close friend and colleague, Mrs. Joyce D'Mello, for all her kindness and the tremendous support she has offered since I joined KEDRI.

I would like to thank past and present members of KEDRI – Knowledge Engineer and Discovery Research Institute, who have directly or indirectly influenced my work and shared their ideas without holding back. All the discussions with fellow researchers and students at KEDRI have been delightful and inspirational especially Dr. Paul Shaoning Pang, Dr. Zeke Chang, Dr. Simeu Gomes Wysoski, Paulo Gottgroy, Anju Verma, Boris Bačić, Snjezana Soltic, Raphael Hu, Harya Widiputra and Stefan Schliebs for their inspirational ideas.

Finally, I would like to thank my parents, grandparents and my partner, Annie, for their unconditional love and support.

This thesis is dedicated to my grandmother, 呂懷瑾, and the loving memory of my grandfather, 黃賢齊, who passed away during my PhD study. I will forever miss him.

ABSTRACT

This thesis presents several novel methods to address some of the real world data modelling issues through the use of local and individualised modelling approaches. A set of real world data modelling issues such as modelling evolving processes, defining unique problem subspaces, identifying and dealing with noise, outliers, missing values, imbalanced data and irrelevant features, are reviewed and their impact on the models are analysed.

The thesis has made nine major contributions to information science, includes four generic modelling methods, three real world application systems that apply these methods, a comprehensive review of the real world data modelling problems and a data analysis and modelling software.

Four novel methods have been developed and published in the course of this study. They are: (1) DyNFIS – Dynamic Neuro-Fuzzy Inference System, (2) MUFIS – A Fuzzy Inference System That Uses Multiple Types of Fuzzy Rules, (3) Integrated Temporal and Spatial Multi-Model System, (4) Personalised Regression Model.

DyNFIS addresses the issue of unique problem subspaces by identifying them through a clustering process, creating a fuzzy inference system based on the clusters and applies supervised learning to update the fuzzy rules, both antecedent and consequent part. This puts strong emphasis on the unique problem subspaces and allows easy to understand rules to be extracted from the model, which adds knowledge to the problem.

MUFIS takes DyNFIS a step further by integrating a mixture of different types of fuzzy rules together in a single fuzzy inference system. In many real world problems, some problem subspaces were found to be more suitable for one type of fuzzy rule than others and, therefore, by integrating multiple types of fuzzy rules together, a better prediction can be made. The type of fuzzy rule

assigned to each unique problem subspace also provides additional understanding of its characteristics.

The Integrated Temporal and Spatial Multi-Model System is a different approach to integrating two contrasting views of the problem for better results. The temporal model uses recent data and the spatial model uses historical data to make the prediction. By combining the two through a dynamic contribution adjustment function, the system is able to provide stable yet accurate prediction on real world data modelling problems that have intermittently changing patterns.

The personalised regression model is designed for classification problems. With the understanding that real world data modelling problems often involve noisy or irrelevant variables and the number of input vectors in each class may be highly imbalanced, these issues make the definition of unique problem subspaces less accurate. The proposed method uses a model selection system based on an incremental feature selection method to select the best set of features. A global model is then created based on this set of features and then optimised using training input vectors in the test input vector's vicinity. This approach focus on the definition of the problem space and put emphasis the test input vector's residing problem subspace.

The novel generic prediction methods listed above have been applied to the following three real world data modelling problems:

1. Renal function evaluation which achieved higher accuracy than all other existing methods while allowing easy to understand rules to be extracted from the model for future studies.
2. Milk volume prediction system for Fonterra achieved a 20% improvement over the method currently used by Fonterra.
3. Prognoses system for pregnancy outcome prediction (SCOPE), achieved a more stable and slightly better accuracy than traditional statistical methods.

These solutions constitute a contribution to the area of applied information science.

In addition to the above contributions, a data analysis software package, NeuCom, was primarily developed by the author prior and during the PhD study to facilitate some of the standard experiments and analysis on various case studies. This is a full featured data analysis and modelling software that is freely available for non-commercial purposes (see Appendix A for more details).

In summary, many real world problems consist of many smaller problems. It was found beneficial to acknowledge the existence of these sub-problems and address them through the use of local or personalised models.

The rules extracted from the local models also brought about the availability of new knowledge for the researchers and allowed more in-depth study of the sub-problems to be carried out in future research.

CHAPTER 1 INTRODUCTION

1.1 Motivation and Objective

Real world data modelling has always been a difficult task. There has not been one method that claims to outperform all other method on all problems. This is because real world problems are usually highly complex and unique, often consisting of a mixture of sub-problems. It is evident by the recent focus on the mixture of models approach to achieve higher prediction accuracy (Cevikalp & Polikar, 2008; Islam, Xin, Shahriar Nirjon, Islam, & Murase, 2008; Kasabov, 2007a; Kim, Pang, Je, Kim, & Bang, 2002; Lei, Yang, & Wu, 2006; Minh Ha, Abbass, & McKay, 2008; Pang, 2004; Xin & Yong, 1998; Zhou & Jiang, 2003).

This principle has been applied in many other fields in the world. Take consumer product marketing as an example, it may be good to design one product that meets the needs of all customers reasonably well but it is better to identify groups of customers with slightly different needs and to design a specialised solution for each group. This is demonstrated by the number of different types of shampoo available in the supermarket. Most well known brands offer multiple products targeting different groups of customers that have different needs.

The same applies to data modelling; if one is to predict whether a patient will have disease or not, there may be several groups of patients that have the same disease but for each group, the cause of the disease may be different for each group. There are even more groups of patients that are healthy for different reasons.

There have been many approaches proposed to address some of the unique problem subspaces by breaking down a problem into multiple sub-problems and to address each one independently. SVM-Tree (Pang, 2004) and SVM-ensemble (Kim et al., 2002) are two well known algorithms that adopt

this type of approach. There are also a large number of applications that apply this approach to solve real world data modelling problems such as breast cancer diagnosis (Übeyli, 2005), glucose monitoring (Kurnik et al., 1999), and motor engine fault diagnosis (Sharkey, Chandroth, & Sharkey, 2000).

A survey of previous literatures shows that there are three types of approach to breaking down a problem:

1. Divide and conquer: if the problem is too difficult for the one model, break it down into easier sub-problems recursively until these are easy enough to be solved with simple solutions.
2. Clustering: divide the problem space into various regions using clustering techniques, train multiple models for each region and use the best model for each region.
3. Individualised model with transductive reasoning is another way to address the issue. It does not break the problem down as the two approaches listed above do and instead, it ignores the problem subspaces that are irrelevant to the current prediction. This, therefore, simplifies the problem and focus only on the sub-problem of concern.

The objective of this thesis is to address the issue of real life data modelling, particularly by addressing problems that consist of a mixture of sub-problems using a mixture of local models or a personalised model strategy.

1.2 Main Contributions

There are nine main contributions of this thesis:

1. Comprehensive analysis and review of the real world data modelling problems.
2. A novel generic method: Dynamic Neuro-Fuzzy Inference System (DyNFIS), published in the ICONIP 2008 Conference Proceedings. Entered in Neural Network Forecasting Competition (NN3) in 2007,

achieved 10th place in the 11-time-series datasets among other neural networks models.

3. A novel generic method: neuro-fuzzy inference system with multiple fuzzy rules (MUFIS), published in the WCCI 2008 conference proceeding.
4. A renal function evaluation (GFR) system based on MUFIS.
5. A novel generic method, an integrated multi-model system using both temporal and spatial models for different views. Published in KEDRI / Fonterra 2007 technical report.
6. A milk volume prediction system based on the integrated multi-model system. Delivered to Fonterra in 2007. Published in a commercial, confidential technical report.
7. A generic personalised regression method with incremental feature selection for classification problems.
8. The Data analysis and applying the personalised regression method on the Pregnancy outcome prediction case study (SCOPE study). Published in a technical report for (SCOPE) study.
9. NeuCom - A Neuro-computing Decision Support Environment (Hwang et al., 2009). This software was primarily developed and maintained by the author prior and during the PhD study. It was used in all case studies as an aid to: visualise, manipulate, cluster and transform data, rank features develop models and measure generalisation errors (see Appendix A for more details). The generic methods proposed in this thesis are to be included in the next release of NeuCom.

1.3 Thesis Structure

The thesis is structured as follows:

CHAPTER 1 presents an introduction to the PhD study and a brief description of the issues related to real world data modelling.

CHAPTER 2 presents a critical analysis of the issues related to real world data modelling problems and review methods that have been proposed to address them.

CHAPTER 3 presents a review of relevant literatures on data modelling, normalisation, feature selection, modelling, clustering and fuzzy inference systems. It reviews a set of existing methods that are adopted as part of the proposed novel methods.

CHAPTER 4 presents a generic, novel fuzzy inference system, DyNFIS (Hwang & Song, 2008), which improves the original DENFIS algorithm by adopting supervised learning on the fuzzy membership function and uses Gaussian membership function instead of the triangular membership function. This algorithm was validated on the Mackey-glass benchmark dataset and later entered in the NN3 competition to compete with other well established algorithms. The algorithm achieved 10th place among other 90 competitors in the 11-time-series competition, without extensive optimisation applied by some of the competitors.

CHAPTER 5 presents a generic, novel fuzzy inference system, MUFIS (Hwang, Song, & Kasabov, 2008), which improves DyNFIS by allowing multiple types of fuzzy rules to be used in a single fuzzy inference system. This allows more suitable consequent function to be used in a fuzzy rule and, therefore, leads to better prediction accuracy.

CHAPTER 6 presents a generic, novel, multi-model system that implements a modified version of the multi-model framework proposed by Kasabov in 2007 (Kasabov, 2007b) that utilises both temporal and spatial models to allow two contrasting views on a single problem.

CHAPTER 7 presents a case study on a milk production volume prediction problem using the methods proposed in chapter 6, allowing the temporal model to make a prediction based only on only the recent data and the spatial model to make a prediction based only on the historical data only.

CHAPTER 8 presents a generic, novel, personalised regression model for classification problems. This method focuses on defining the problem space through the use of extensive incremental feature selection procedure. The process is based on univariate analysis of the data and previous studies on these variables, when available. For each prediction, a personalised regression model is created dynamically by optimising the global model with local training input vectors that are in the vicinity of the test input vector.

CHAPTER 9 presents a case study on pregnancy outcome prediction using the personalised regression method proposed in Chapter 8.

CHAPTER 10 presents the discussion, conclusion and future work.

CHAPTER 2 REAL WORLD DATA MODELLING ISSUES - A CRITICAL ANALYSIS

There are significant differences between real world problem and synthetic problem. Synthetic problems are man-made to demonstrate the ability of a method to solve a particular problem. For these problems, the focus is on highlighting a particular type of problem and how it was solved.

Real world problems, on the other hand, are real challenges that may involve various issues and problems that are beyond the researchers' imagination.

There are several issues in, or characteristics of, real world data modelling problems that may have an effect on today's modelling methods.

- Evolving problems
- Unique problem subspaces
- Noise
- Outliers
- Missing values
- Imbalanced data
- Irrelevant features

The details of each issue are discussed below:

2.1 Evolving Problems

Many real world problems evolve over time with new patterns emerging without precedence. These evolving problems have appeared in many fields, including biomedical sciences (Marshall, Song, Ma, MacDonell, & Kasabov, 2005; Rodrigo & Learn, 2000), Finance (Widiputra, Pears, Sergueeva, & Kasabov, 2008), online document classification (Z. Chen, Huang, & Murphey, 2007) and

Ecology (Damousis, Alexiadis, Theocharis, & Dokopoulos, 2004). There is usually a large amount of historical data available and new data is collected on an ongoing basis. As the amount of data increases, the resources required to develop a model also expands. It is not possible to develop a fixed model that works on future data if the problem changes over time.

The ever increasing amount of data in this type of problem is also an issue as some neural network models requires iterative training using all available data and, therefore, the more data that is available, the longer the training time. There is also the issue of storing the data if more is always coming, in some cases, this is impossible.

Various global and local models with an incremental learning capability have been proposed to address this issue (Domeniconi & Gunopulos, 2001; Kasabov & Song, 2002; Okamoto, Ozawa, & Abe, 2003; Ozawa, Pang, & Kasabov, 2008; Wenhua & Jian, 2004; Widiputra et al., 2008), by adding the ability to learn knowledge from new data without losing previously learned knowledge and to discard most of the data once it has been processed.

The individualised model with transductive reasoning also addresses part of this problem. This type of model is created dynamically based on only the relevant data. This reduces the time required to train a model due to smaller size of the training data and allows the latest data to be used, but increases the time required to search for relevant data in the entire dataset.

2.2 Unique Problem Subspaces

A unique problem subspace is defined as a specific region in the problem space that has significantly different characteristics from the rest. See Figure 3.2. Because of this, a global model may not work well in all these different regions since it focus on solving the overall problem.

These regions may be identified through clustering processes such as k-means (MacQueen, 1967), fuzzy c-means (Dunn, 1973) or hierarchical clustering (Johnson, 1967), which assigns input vectors to groups (clusters) based on their similarity. The similarity is measured using distance measures such as Euclidean and Manhattan distances. These clusters can be treated differently or contribute to the output at different levels, based on their characteristics.

Many global and local models (Bishop & Svensén, 2003; Kasabov & Song, 2002; Koskela, Varsta, Heikkonen, & Kaski, 1998; Vernieuwe, Verhoest, De Baets, Hoeben, & De Troch, 2007) make use of cluster information and apply appropriate methods depend on the characteristics of the cluster to achieve better prediction accuracy.

2.3 Noise

Noise, or measurement error, refers to a difference in value when the same object is measured multiple times. It can be either systematic error or Gaussian white noise.

Systematic error - the measurement is biased and leads to situation where the measured value is constantly higher or lower than the actual value. Systematic error may be corrected if the amount of bias is known.

Gaussian white noise or nature process variation - the measurement differs from the actual value at random. It may be filtered to remove the noise, but it will lead to loss of information in the details.

Noise is a by-product of the normal data collection process, as most measuring equipment carries a certain degree of error. This may be caused by the data source, measuring method and/or equipment and usually cannot be completely eliminated. The term “noise” used in this thesis refers to Gaussian white noise.

There is a significant difference in the negative impacts caused by noise depends on whether it appears in the input variables or the output. The noise that appear in the output data of the dataset is named “Output Noise” and the noise that appear in the input variables is named “Input Noise”

2.3.1 Output Noise

Output noise refers to the uncertainty in the output values which are unpredictable in nature. This limits the performance of the model. For example, if we have 10 patients with identical clinical records and a model is trained to predict the risk of these patients to have disease X. In theory, these 10 patients should all have the same risk, but in reality, all 10 patients have very different risk, it could range between 5% to 15%. If we train a model with all 10 patients, the predicted risk will most likely to be 10% for all patients. This is caused by inaccurate estimate of risk in the training data, hence the output noise. There is no logical way to improve this prediction.

Output noise has a negative impact on all data modelling problems, particularly on classification problems where the values are binary.

2.3.2 Input Noise

Input noise has a more complicated influence on the model than output noise. Input variables define the problem space. Noise blurs the definition of the problem space and is known to increase generalisation error in proportion to the amount of noise (Sugiyama, Okabe, & Ogawa, 2004). This can cause misperception on the problem and therefore causes incorrect solution to be derived. In feature selection methods, input noise causes incorrect evaluation of a feature’s discriminating power. In clustering, it affects the position of data points and leads to incorrect clusters to be identified.

See Figure 3.4 for an example of input noise affecting a model's prediction. In a data modelling problem that contains useful patterns, they are likely to be masked by the noise and leads to a reduction in the model's prediction accuracy.

2.4 Outliers

In addition to noise, random and abnormal events sometimes occur in real life and they are reflected in the data, appearing as outliers (Ruey, 1988). They are often caused by human error, natural events or equipment failure. Outliers have a negative influence on the models and should therefore be identified and removed. (See Figure 2.1 for an example of an outlier's influence)

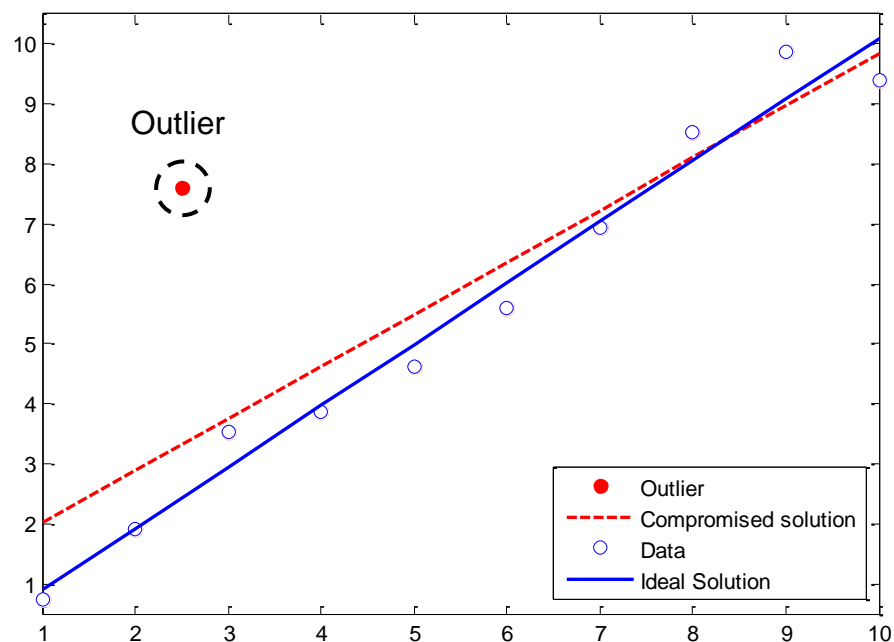


Figure 2.1 A single outlier in a group of 11 input vectors can significantly compromise a model.

It can be difficult to identify outliers from normal data. One way is to identify them during the data collection process when something abnormal happens and correct them accordingly based on the type of event. This may be the only way to remove outliers that happen to be very similar to normal data. This does require a good knowledge of the data domain.

Unfortunately this information is available for only a very small number of problems when the researcher was aware of possible causes of outliers and has a procedure in place to record them as part of the data collection process. If outliers are not identified before they are entered into the dataset, removing them will be a subjective exercise to decide whether a value is an outlier or not.

When a suspected outlier is identified, removing it may not be a simple process. Some outliers can be transformed to normal data if the cause of the outlier is known, which can help improving the quality of the dataset. This is particularly important on smaller experiments where every piece of data is crucial.

2.4.1 Identifying Outliers

Most outlier detection methods are based on mean and standard deviation for data with normal distribution, Chauvenet's Criterion (Ross, 2003; Taylor, 1997) is an example of this type of method, with the assumption that if a value falls outside the acceptable range from the mean, it is likely to be an outlier.

Scatter plot and histogram are the two commonly used visual aids for manually identifying outliers for problems with few variables. For real world problems with large numbers of variables, it becomes very difficult, if not impossible, to identify outliers. This is due to the fact that many variables may be noisy and less relevant to the problem. There is no sure way of deciding whether one value is an outlier or an emerging problem subspace.

2.4.2 Dealing with Outliers

If a model is trained with data that contains outliers, the model's performance is likely to be compromised by the incorrect training it has received. (See Figure 2.1)

There are several ways to deal with outliers, as follows:

2.4.2.1 Delete the outliers

If there is only a very small number of outliers in a large dataset, the outliers can simply be removed or treated as missing values if they can be identified.

2.4.2.2 Transform the outliers

It is possible to transform outliers into normal data without introducing significant problem if the cause of the outliers is known.

Consider this scenario: a farm produces x litres of milk per day and the milk is picked up every evening. In one instance, the farm was not able to finish the milking on time and therefore some of the milk has to be picked up in the next shift. On the milk collection company's record, the current day's milk pickup volume is lower than normal, and the next day's pickup volume is higher than normal. The input vectors for these two days may be identified as outliers by the milk collection company. However, since the cause of this outlier is known, it can be transformed to correct value by adjusting the milk volume of these two days based on the recent trend.

2.5 Missing data

Here missing data is referred to as the missing values in a variable.

2.5.1 Random Missing (Scheffer, 2002)

1. Missing completely at random (MCAR)

For an input vector, if the missing variable is unrelated to other variables and the data was collected randomly, this is called missing completely at random. This type of random missing may be treated through imputation methods.

2. Missing at random (MAR)

“Missing values are not randomly distributed across all observations but are randomly distributed within one or more subsamples (ex., missing more among whites than non-whites, but random within each subsample)” (Garson, 2008)

This type of random missing may be treated through imputation methods taking into relevant variables into consideration.

3. Not missing at random (NMAR)

The value of a certain variable in an input vector is missing because of its value. For example, low income patients are less likely to provide their income details. This type of missing value may not be treated directly. One must identify the cause and then treat it manually.

Here, the data is missing occasionally for a reason. This is different from Structure missing, as explained later in this section, where the data is missing by design.

2.5.2 Random Missing Value Treatment

There are various methods to treat both MCAR and MAR using its relation to other variables or its distribution. Commonly used methods are expectation maximisation (EM) (Jamshidian & Jennrich, 1997), multiple imputation (Schafer, 1999), medium values, minimum value, maximum value and K-Nearest Neighbour (KNN) method (Hastie et al., 1999; Qinbao, Martin, Xiangru, & Jun, 2008) are all often used.

It is critical to know the cause of the missing value before treating it. For example, if the cause of the missing values is due to the fact that the measuring equipment is unable to detect any value below 0.5, then the missing values should be filled with either the minimum value of the population or zero. If the missing values are treated with an inappropriate method such as average of the observed individuals, then it will lead to mean of the variable to be higher than the actual mean and leads to higher generalisation error.

Some models, such as C4.5 decision tree (Quinlan, 1993), K-Nearest Neighbour (Dasarathy, 1990) and Weighted K-Nearest Neighbour (Lora, Santos, Exposito, Ramos, & Santos, 2007), can work with missing values directly, without imputing the values prior to the training process. However, the prediction made for input vectors with missing values may not be as trust worthy as the rest.

2.5.3 Structural Missing

Structural missing refers to values that are missing for a valid reason and therefore should not be treated as random missing values. Take pregnancy clinical examinations for example, previous pregnancy details should be missing for patients that are in their first pregnancy. This type of missing value may be caused by the experiment design and no attempts should be made to fill in the missing value in this case.

2.6 Imbalanced data

This issue refers to the dataset for classification problems where one class has significantly more input vectors than others thereby causing the minority class to be “overwhelmed” by the majority class (Japkowicz, 2000b; Kubat & Matwin, 1997).

In many research fields, particularly in biomedical science, prediction of any disease with low prevalence will have imbalanced dataset as the number of input vectors in the disease class is many times less than the healthy class. Preeclampsia is a typical example, only 5% of pregnant women are expected to have it in their first pregnancy (Cnossen et al., 2006).

Most of today’s modelling methods are designed with the assumption that the dataset is balanced between classes since their objective function is overall accuracy (Japkowicz, 2000a). When these methods are applied to imbalanced datasets, the outcome will be biased toward the class with more input vectors since that class contributes more to overall accuracy.

In addition, if the problem space is large and/or the input vectors in the disease group are few, there may not be enough disease input vectors to cover the problem space sufficiently and resulting in an incomplete model. See Figure 2.2 for an example, most of the problem space is filled with the healthy patients. With so few patients with disease, it is very likely that some of other sub-groups (types) of disease patients may not be included in the current dataset.

There is no solution to this problem; no matter how the minority class is reinforced or treated. This problem is less severe on large datasets as there may be at least a few patients from each sub-group of disease patients.

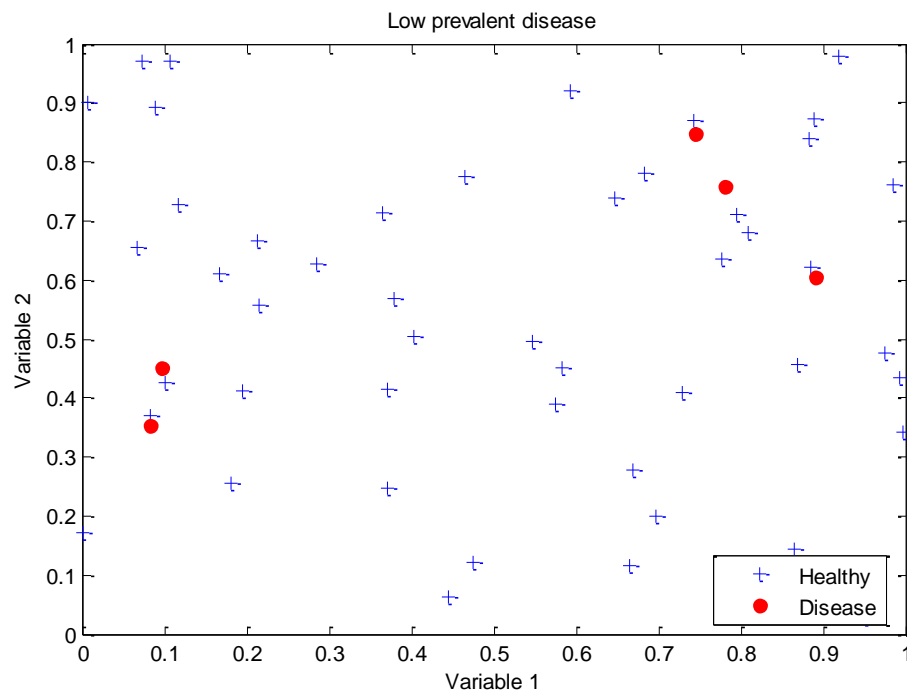


Figure 2.2 Illustration of low prevalence disease data in two dimensional space, 50 healthy patients, 5 patients with the disease.

There are several methods that may be used to address this issue:

2.6.1 Receiver Operating Characteristic (ROC) Curve (Swets, 1988)

This evaluation metric aims to introduce carefully controlled bias toward the minority class to increase its accuracy at the cost of the accuracy of the other class.

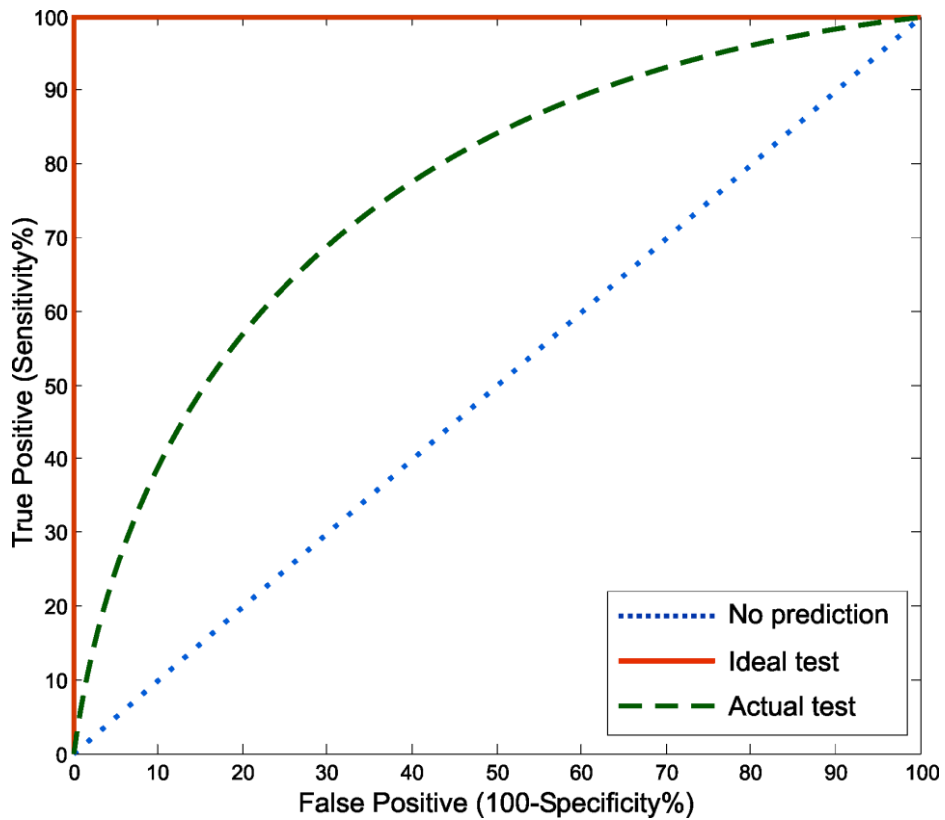


Figure 2.3 Example of ROC curve. The more area under the curve the better the model

The ROC curve shows the trade off between accuracy in both classes at different thresholds and allows the researchers to decide on the best threshold.

2.6.2 Re-Sampling(Japkowicz, 2000a)

The objective of this method is to increase the number of input vectors in the minority class using random sampling or a focused sampling method to create a balanced dataset. The random sampling method re-samples the input vectors with added noise for all input vectors in the minority class. The focused sampling only re-samples the input vectors close to the boundaries.

2.6.3 Down-Sizing(Japkowicz, 2000a)

The objective of this method is to remove random input vectors from majority class using either random downsizing or focused downsizing to achieve a balanced dataset.

The random downsizing method randomly removes input vectors from the majority class without regard to position.

The focused downsizing method removes input vectors that are far away from the boundaries in the majority class.

2.6.4 One-Class Learning

Instead of creating a classifier to discriminate between classes, this approach uses the so called one-class classifier to measure the amount of similarity between the input vector and the target class (D. Tax & Juszczak, 2002). The one-class classifier has been shown to work better than the two-class classifier on imbalanced datasets (Bhavani & Adam, 2004).

2.7 Relevance of Features

Many redundant or irrelevant variables are often collected to avoid missing out an important predictor that may be crucial to the success of the model. There are often too many variables used to describe a problem and this leads to the “curse of dimensionality”(Bellman, 1961). This causes significant issues for both statistical and neural network models and must be addressed before the data is used to train a model.

This is a very complicated issue and is largely related to classification problem such as medical diagnosis (Guyon & Elisseeff, 2003). All distance and cluster based prediction methods are severely affected by this due to the fact that the distance is measured based on the difference between input vectors in m -variables. If an irrelevant variable is used as part of this measurement, it may offset the usefulness of other useful variables.

A large number of studies have been carried out in the past to address this issue, with the aim of minimising the number of variables used to define the problem space while maximising the prediction accuracy. However, it is still

reasonably difficult to obtain a good set of features from datasets with a moderate amount of noise.

Take microarray datasets for example, this type of dataset usually has a very small number of input vectors and thousands of variables. Due to a very small number of input vectors populating an extremely large problem space, there are not enough input vectors to properly represent the problem. Every input vector can have a very strong influence on the quality measure of the variables and therefore the useful variables may be selected just by chance and will not be useful when applied to future data.

This was evidenced by applying an univariate feature selection method, i.e. Signal to Noise Ratio (SNR) (Goh, Song, & Kasabov, 2004), which was used to measure variables' ability to discriminate input vectors of different classes based on their mean and standard deviation, to the lymphoma dataset (Shipp et al., 2002). This gave us a SNR value for each variable. When 10% of the input vectors were removed randomly from the dataset and the SNR was applied again, the SNR value for each variable was significantly different and the ranking of the variables was changed because of this.

Feature selection is often included as part of the data modelling algorithm as a model include both the features, which defines the problem space, and the parameters of the model. There are two commonly used methods: the filter method and the wrapper method. An integrated method has recently been proposed. These methods are explained as follows:

2.7.1 Filter Method

This method selects features without regard to which prediction algorithm and parameters are used. This method selects features based on analysis of the training data before the model is trained with it.

The filter method (Hall, 1999; Koller & Sahami, 1996) has the advantage of being less computationally expensive. The filter method allows a subset of features to be obtained quickly for problems with a large number of variables.

2.7.2 Wrapper Method

This method wraps feature selection around the prediction algorithm. It uses cross-validation, explained in next chapter, to evaluate the performance of a feature subset on a prediction algorithm and decides whether adding or removing feature variable would improve the prediction accuracy.

For example, consider that we have n features, one could repeat cross validation n times, each time remove one feature from the feature set without repeat. The feature set that leads to the highest accuracy of the model is kept for the next round of feature elimination. Repeat the procedure until only the desired number of features remains or desired accuracy is achieved.

The wrapper method (John, Kohavi, & Pfleger, 1994; Kohavi & John, 1997) generally performs better than the filter method as it takes the model's performance into consideration at the expense of being more time consuming due to the iterative process.

2.7.3 Integrated Method

A model is a combination of two elements. First, the set of features that is used define the problem space. Second, the parameter of the model, for example, connection weights between nodes in MLP. However, many neural network models has a set of training parameters to govern how the parameters of the model is updated given a training dataset and the training parameters also have strong influence on the performance of the model.

The limitation of the wrapper method is that when the model requires a set of training parameters, these parameters are not optimised as part of the wrapper method.

Feature subset A may provide best accuracy for a model with training parameter set X, but this may not be the case on model with training parameter set Y. It is therefore best to optimise both feature subset and model training parameters together to obtain the best combination of both on a given problem. This may be seen as a wrapper method that optimises both feature subset and model training parameters. It should also use cross-validation to obtain a generalised solution.

The model training parameters and feature subsets may both be considered as parameters that need optimising when developing a model. It then may be possible to use well known optimisation techniques such as Genetic Algorithm and Evolutionary Strategy or the recently proposed “Versatile Quantum-inspired Evolutionary Algorithm” (vQEA) (Schliebs, Platel, & Kasabov, 2008) to handle this task by optimising both the features subset and the model’s training parameter together.

2.8 Conclusion

Real world problems are complex and difficult to handle. Most, if not all, of the issues stated above will limit the potential of the final model. Issues such as noise, missing values, outliers may be addressed prior to the modelling process if care is taken to avoid introducing bias, while the others may be addressed as part of the modelling algorithm.

The issue of unique problem subspaces has been recognised in many previous studies and is known to have a significant impact on prediction accuracy. In the opinion of the author, it would be beneficial to put more emphasis on the unique problem subspaces.

The contribution of this thesis includes four novel and generic methods for real world data modelling that aims to address the unique problem subspaces issue at different levels.

The first method is named “DyNFIS – a dynamic neural fuzzy inference system”, which extends the original Dynamic Neural Fuzzy Inference System (DENFIS) algorithm (Kasabov & Song, 2002) by adding supervised learning and uses a more sophisticated membership function. This was entered in the NN3 neural network forecasting competition and achieved 10th place in the 11-time-series competition.

The second method is named “MUFIS – a fuzzy inference system that allows multiple fuzzy rule types”, which extends DyNFIS by adding the capability of using both Takagi-Sugeno and Zedeh-Mamdani rule types in a single fuzzy inference system.

The third method is a multi-model system that integrates both temporal and spatial models to provide contrasting views of the problem.

The fourth method is a generic personalised regression that focuses on defining the problem space through extensive incremental feature selection and then optimises the global regression model for each test input vector using only the part of the dataset that’s relevant.

The research and the methods presented in this thesis are a continuation of much previous literature and therefore a selection of relevant literatures is reviewed in the next chapter.

CHAPTER 3 MODELLING TECHNIQUES FOR COMPUTATIONAL INTELLIGENCE – A LITERATURE REVIEW

Kasabov 2007 put predictive models into three different categories (Kasabov, 2007b).

1. Global Model

A global model is single model that learns from the entire dataset. The developed model is then applied on future data.

2. Local Model

The local model is a fixed mixture of models trained on the entire dataset. However, when it is applied to future data, only one or a subset of the relevant models will contribute to the prediction.

3. Personalised Model

A personalised model is an individualised model that is created dynamically for each prediction, using only relevant input vectors through transductive reasoning.

3.1 State of the Art Global Models

Most of today's predictive models are global (inductive) models, where the model learns from the training data and then applied on future data. Linear Regression (Tibshirani, Friedman, & Hastie, 2001), Multi-layer Perceptron (MLP) (Hornik, Stinchcombe, & White, 1989; Sheng-Sung, Chia-Lu, & Chien-Min, 2006), Support Vector Machine (SVM) (Boser, Guyon, & Vapnik, 1992; Hagan, Demuth, & Beale, 1996; Kim et al., 2002; Vapnik, 1998), Adaptive-Network-Based Fuzzy Inference System (ANFIS)(Jang, 1993; Jang & Sun, 1995; Jang, Sun, & Mizutani, 1997) and Echo State Network (ESN) are examples of global models. MLP and SVM machine learning algorithms were proposed many years ago and are still the two most widely used neural network models.

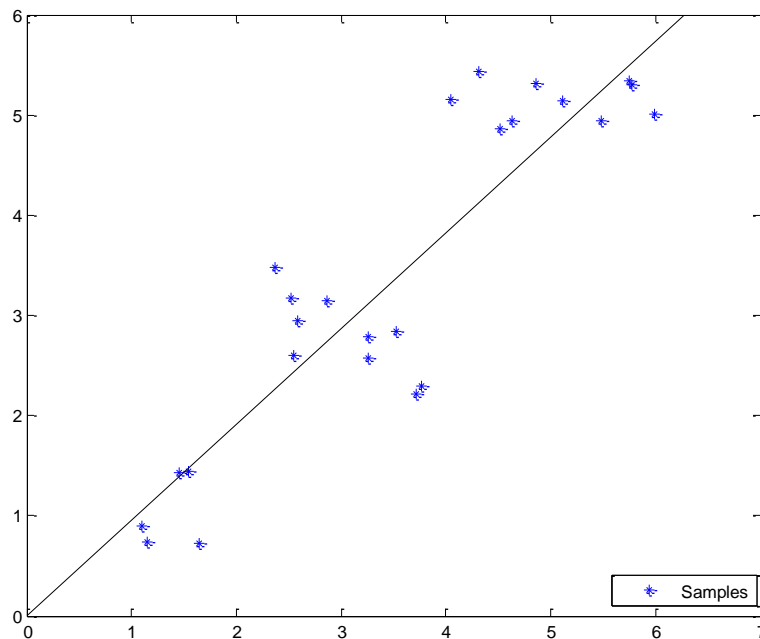


Figure 3.1 Illustration of a global model. A single linear function is used for all input vectors

The global model is a single, fixed, reusable model, trained with the entire dataset and can then be applied on future data.

There are a few limitations with this type of model:

First, if a new pattern emerges in the future, the existing model will not be able to handle it as the model has not been trained to recognise this pattern and a new model may need to be developed. This can be time consuming depending on the model and the complexity of the problem.

Second, as the model is developed based on all available data with the objective of minimising overall prediction error, it will be biased toward the majority of the data. A pattern without enough support will have little influence on the model.

This is similar to the issue with interpolation versus extrapolation. If the new pattern is similar to some existing pattern, then it is considered interpolation, where there is enough support the prediction made for this pattern. However, if the new pattern is very different from any of the existing patterns, then it is

considered extrapolation, where the prediction made for this new pattern is less meaningful and subject to greater uncertainty.

Recent research in the field of machine learning has focused on model ensembles which use a mixture of models to achieve better overall accuracy. Several studies have reported that an ensemble of models works better than a global model (Cevikalp & Polikar, 2008; Islam et al., 2008; Kim et al., 2002; Lei et al., 2006; Minh Ha et al., 2008; Pang, 2004; Xin & Yong, 1998; Yao & Liu, 1996; Zhou & Jiang, 2003).

There are many strategies that are commonly used to create an ensemble: bagging (Kim et al., 2002), boosting (Islam et al., 2008) and clustering (Kasabov & Song, 2002) are well known strategies. Depending on the strategy used, the ensembles generally try to either generate different view of one problem or break down the problem into smaller problems and tackle each problem independently or sometimes both.

The final output for this type of model can be categorised by the following two methods: model selection or model fusion.

3.1.1 Model Selection

Model selection types of local models uses one or a few, sub-models that are deemed most suitable for a given input vector and aggregate the output from these sub-models. SVM Tree (SVMT) (Pang, 2004) and Combination of Multiple Classifiers (CMC) (Woods, Kegelmeyer, & Bowyer, 1997) are two examples of this type of local model.

3.1.2 Model Fusion

Model fusion type local models use the weighted average of the output from all sub-models (Franco & Nanni, 2009; Freund & Schapire, 1999; D. M. J. Tax, van Breukelen, Duin, & Josef, 2000; Tin Kam, Hull, & Srihari, 1994). The weight can be fixed, adjustable or dynamically updated.

3.2 Local Models

Local models (Fontenla-Romero, Alonso-Betanzos, Castillo, Principe, & Guijarro-Berdiñas, 2002; Kasabov, 2001; Kasabov & Song, 2002; Lei et al., 2006; Lucks & Oki, 1999; Poggio, 1994; Song & Kasabov, 2005; Yamada, Yamashita, Ishii, & Iwata, 2006a) is a type of model ensemble that breaks down the problem into many smaller sub-problems, based on its position in the problem space. The sub-problems can be defined through a clustering process such as k-means, fuzzy c-means and hierarchical clustering that group similar input vectors based on their similarity (distance measure).

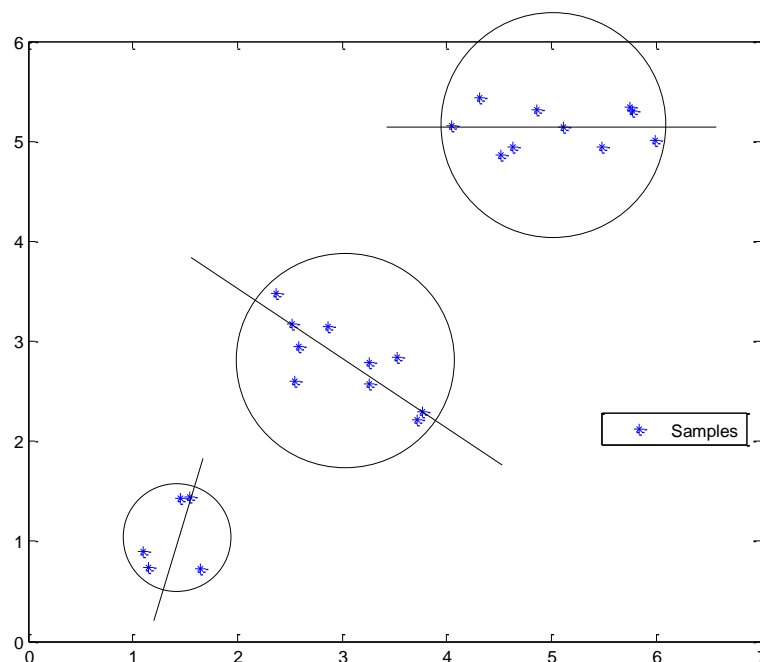


Figure 3.2 Illustration of a local model consists of three sub-models.

This type of model assumes that each cluster is a unique problem subspace and a sub-model should be developed for it. The quality of the cluster is, therefore, the foundation of this type of model.

The data clustering parameters often need to be adjusted, according to the sub-model's requirements or the characteristic of the problem. Many models, such as linear regression, need the number of input vectors to be significantly

greater than the number of variables and, therefore, the clusters must be large enough to support this type of sub-model. Hence, local models may require more training data than global model to ensure that each sub-model is trained with sufficient amount of input vectors.

In addition, the clustering process is strongly affected by the amount of noise in the data of irrelevant or redundant features, as it affects the distance measure used by most clustering methods.

3.3 Individualised (Personalised) Model through Transductive Reasoning

Transductive reasoning (Kasabov & Pang, 2003; Mitchell, 1997; Song & Kasabov, 2004; Song & Kasabov, 2006; Song, Ma, & Kasabov, 2006; Vapnik, 1998) was originally proposed by Vapnik in 1998 to develop an individualised model through transductive reasoning for a given input vector without first developing a generalised model in the intermediate stage. This approach has been widely used to solve various real life problems like text classification (Y. Chen, Wang, & Dong, 2003; Joachims, 1999), speech recognition (Joachims, 2003), image recognition (Li & Chua, 2003) and language translation (Ueffing, Haffari, & Sarkar, 2007).

The main difference here is that transductive reasoning focuses on finding a solution for each prediction instead of the creating a generalised solution for the problem and then use it for each prediction.

The model is created dynamically for each prediction, which utilises all available data and uses the most suitable parameters, features or model to make the prediction.

“k-nearest neighbour” (Soucy & Mineau, 2001; Yamada, Yamashita, Ishii, & Iwata, 2006b), may be considered the most simple form of individualised model through transductive reasoning.

It takes a subset of training data (neighbours) that is relevant to the current test input vector and makes the prediction based on the output of the neighbours.

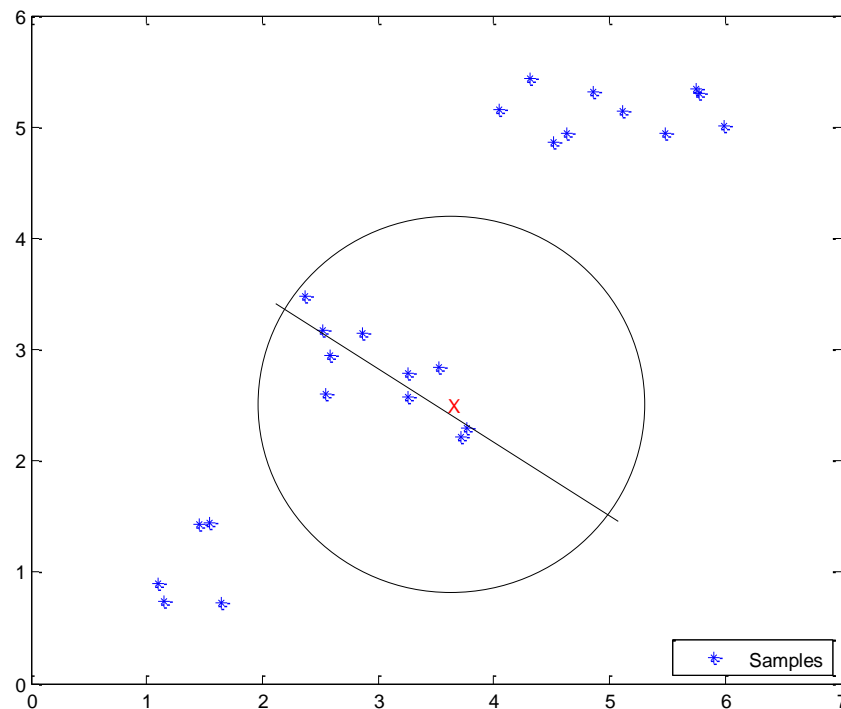


Figure 3.3 Illustration of a personalised model. A model is created on the fly using a subset of training data near the test input vector (X)

Transductive reasoning has the following benefits over global and local models:

1. In a real world problem where the amount of data increases on an ongoing basis, an individualised model through Transductive reasoning will utilise the latest data but only the part of the data that is relevant to the test input vector is used to make the prediction
2. Since only a relevant subset of the input vectors in the training data is used to derive the solution, it may reduce the affects of outliers, or incorrect identification of sub-problems.

The limitation of Transductive reasoning is in its reliance on good definition of problem space. A good definition of problem space is important to all models, however, it may be more so on individualised model through Transductive

$$\beta = [\beta_0, \beta_1, \beta_2, \dots, \beta_q]^T$$

and y is a $m \times 1$ vector:

$$y = [y_1, y_2, \dots, y_m]^T.$$

The i^{th} row of the joint data matrix $[A; y]$, denoted by $[a_i; y_i]$, is related to the i^{th} input-output data pair $([x_{i1}, x_{i2}, \dots, x_{iq}], y_i)$, through

$$a_i = [1, x_{i1}, x_{i2}, \dots, x_{iq}].$$

$[a_i; y_i]$ is referred as the i^{th} data pair of the learning data set.

To obtain β , we modify equation (3.2) by incorporating an error vector e for random noise or error as follows:

$$\mathbf{A} \beta + e = y \quad (3.4)$$

The search for a β that minimises the sum of squared error can be achieved through

$$E(\beta) = \sum_{i=1}^m (y_i - a_i \beta)^2 = e^T e = (y - \mathbf{A} \beta)^T (y - \mathbf{A} \beta) \quad (3.5)$$

where $e = y - \mathbf{A} \beta$ is the error vector produced by a specific choice of β .

The theorem of least square estimator is given as follow:

3.4.1.1 Least Square Estimator: (LSE)

The square error in equation (3.5) is minimised when $\beta = b$, called the least square estimator (LSE), which satisfies the normal equation

$$\mathbf{A}^T \mathbf{A} b = \mathbf{A}^T y \quad (3.6)$$

If $\mathbf{A}^T \mathbf{A}$ is non-singular, b is unique and is given by

$$b = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T y \quad (3.7)$$

3.4.1.2 Weighted Least Square Estimator (WLSE)

LSE assumes that every element of the error vector e has the same weight toward the overall squared error. However, each input vector may contribute differently in real life applications. To consider this factor, we can construct a weighting matrix and have the weighted squared error as

$$E_w(\beta) = \sum_{i=1}^m w_i (y_i - a_i \beta)^2 = (y - A\beta)^T W (y - A\beta) \quad (3.8)$$

where W is a diagonal matrix, which defines the contribution of each input vector, it can be defined in different ways depends on the design of the experiment. In an example of creating a local function in a cluster centre, the weight may be (1-normalised Euclidean distance) between the training input vectors and the cluster centre.

$$W = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & w_m \end{pmatrix}$$

and $0 < w_i \leq 1$; $i = 1, 2, \dots, m$.

with the weighted b_w being:

$$b_w = (A^T W A)^{-1} A^T W y \quad (3.9)$$

LSE is a very efficient modelling algorithm that provides very fast training and prediction. It is most suited for datasets that contain higher levels of noise as the linear model generalises the noise and provides a linear function that best fits the data as a whole. Figure 3.4 shows a linear problem with a medium level of noise. Linear functions like LSE will fit a single line that is very close to the ideal solution while noise may affect non-linear functions and produce a compromised solution.

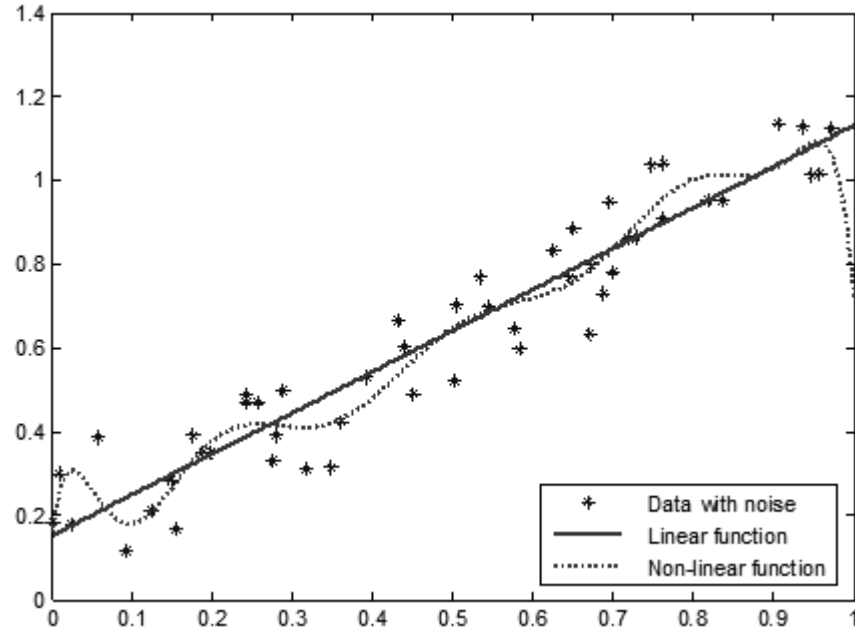


Figure 3.4 Linear and non-linear model prediction on a noisy dataset.

On the positive side, LSE provides a simple linear function that is stable, and easy to use. It is therefore widely used in every field.

On the negative side, since the model is designed to solve a global problem, this algorithm disregards the potentially unique characteristics of the problem subspaces and assumes only one solution exists for the entire problem.

3.4.1.3 Weight Recursive Least Square Estimator (WRLSE)

The weighted recursive least-square estimator – WRLSE (Tibshirani et al., 2001), or weighted on-line linear regression, creates a linear function that is updated continuously with every new data point.

The linear function can be expressed as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1 + \dots + \beta_q x_q \quad (3.10)$$

and for obtaining this function there is a learning data set that is composed of p data pairs $\{ ([x_{i1}, x_{i2}, \dots, x_{iq}], y_i), i = 1, 2, \dots, p\}$, we can calculate $\mathbf{b} = [b_0 \ b_1 \ b_2 \ \dots \ b_q]^T$, the least-square estimator (LSE) of $\boldsymbol{\beta} = [\beta_0 \ \beta_1 \ \beta_2 \ \dots \ \beta_q]^T$, by using the following formula:

$$b_w = (A^T W A)^{-1} A^T W y \quad (3.11)$$

Where

$$A = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1q} \\ 1 & x_{21} & x_{22} & \cdots & x_{2q} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{p1} & x_{p2} & \cdots & x_{pq} \end{pmatrix} \quad (3.12)$$

$$W = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & w_p \end{pmatrix} \quad (3.13)$$

$$y = [y_1 \ y_2 \ \dots, \ y_p]^T \quad (3.14)$$

and w_j is the distance between j -th example and the corresponding cluster centre, $j = 1, 2, \dots, p$.

Here, we rewrite the Equation (3.11) as shown:

$$\begin{array}{l} \text{Initial} \\ \text{Weighted LSE} \end{array} \quad \begin{cases} P_w = (A^T W A)^{-1} \\ b_w = P_w A^T W y \end{cases} \quad (3.15)$$

where w is the weight defined in Equation (3.13) and λ is a forgetting factor whose typical value is between 0.8 and 1. The initial values of P_w and b_w are calculated using Equation (3.15)

Let the k -th row vector of matrix A defined in Equation (3.12) be $a_k^T = [1 \ x_{k1} \ x_{k2} \ \dots \ x_{kq}]$ and the k -th element of y be y_k , then b can be calculated iteratively as follows:

$$\begin{array}{l} \text{Weighted} \\ \text{Recursive} \\ \text{LSE} \end{array} \quad \begin{cases} b_{k+1} = b_k + w_{k+1} P_{k+1} a_{k+1} (y_{k+1} - a_{k+1}^T b_k) \\ P_{k+1} = \frac{1}{\lambda} \left(P_k - \frac{w_{k+1} P_k a_{k+1} a_{k+1}^T P_k}{\lambda + a_{k+1}^T P_k a_{k+1}} \right) \end{cases} \quad (3.16)$$

$$k=n, n+1, \dots, p-1$$

3.5 Artificial Neural Networks (ANN)

3.5.1 Brief History of Perceptrons

Frank Rosenblatt published one of the earliest neural networks in 1958 named “The Perceptron” (Rosenblatt, 1958), which has influenced many artificial neural networks since. The basic perceptron is shown in Figure 3.5.

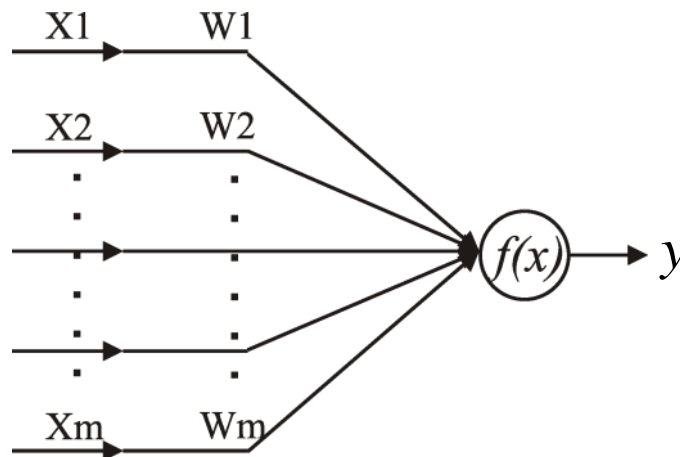


Figure 3.5 Weights are applied to the inputs and the weighted sum is then passed through a function to produce the final output y

For an input vector pair $[x_i, y_i]$, the connection weight w_m is updated if the output of the perceptron $f(x)$ is different from the y_i .

The single layer perceptron is only capable of learning linear separable data as demonstrated by Minsky and Papert in 1969 (Minsky & Papert, 1969). They showed that the XOR function cannot be approximated with this method and incorrectly conjectured that this applies to networks with multiple layers of perceptrons. This leads to a slowdown in research in the field of connectionism until the 1980s.

In 1986, David Rumelhart, Geoffrey Hinton and Ronald Williams introduced back-propagation on a multi-layer perceptron network (Rumelhart, Hinton, & Williams, 1986). This method remains one of the most widely used

supervised learning methods, which can be used in either batch or incremental learning.

The learning process of the original back-propagation for batch learning can be slow and many new learning methods have been proposed to speed up the process (Riedmiller & Braun, 1992; Scott & Christian, 1990).

In 1989, Hornik showed that a multilayer perceptron is a type of universal approximator that can learn virtually any type of function provided enough hidden nodes are used (Hornik et al., 1989).

Later in 1999, Freund and Schapire showed that by using the perceptron algorithm in higher dimensional space with kernel functions, non-linear separable data can be handled (Freund & Schapire, 1999).

3.5.2 Multi-Layer Perceptron (MLP)

MLP (Hornik et al., 1989; Rumelhart et al., 1986) is a feed forward neural network model that can learn the input and output relationship of a dataset through adjusting layers of perceptrons and their connection weights.

The weights are adjusted by applying error back-propagation method, which minimises the difference between the predicted output and actual output.

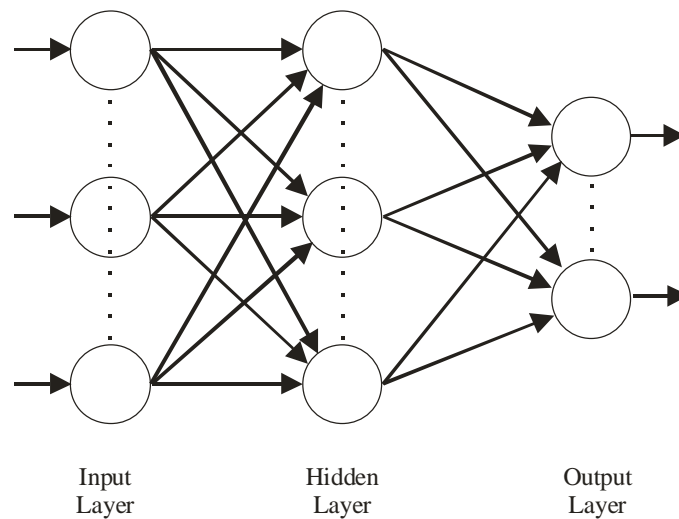


Figure 3.6 An illustration of the structure of a single hidden layer multi-layer perceptron network.

Sigmoid function is the most widely used activation function for the hidden layer. Sigmoid or sum function can be used at the output layer level depending on the type of problem. The connection weights are initially randomised and then optimised through error back-propagation.

The input of the data is fed forward from the input layer through the hidden layer and output layer and the predicted output is then derived.

If the predicted output differs from the actual output, the error is then propagated back through the layers and the connection weights and the activation functions are adjusted based on the learning rate to minimise the error. Multiple iterations of this procedure on all training data are required until the network converges.

The multi-layer perceptron with back-propagation (MLP-BP) is one of the best known models. Figure 3.7 shows an illustration of a MLP-BP with two hidden layers. The signal flow for each node is illustrated in Figure 3.8. A version of MLP-BP with two hidden layer is shown in Figure 3.7. Detailed algorithm description is shown below:

$x^{(p)}$ and $t^{(p)}$ denote the p^{th} training data pair; $z^{(p)}$ denotes the actual output of the network; $w_{ji}^{(l)}$ denotes the weight from neuron i to neuron j , and $w_{kj}^{(h)}$ denotes the weight from neuron j to neuron k . A sigmoid transfer function (3.18) is used as the activation function in the neurons of the hidden layer and the output layer. The goal of training is to get the minimum value of error E :

$$E = \sum_{p=1}^P E^{(p)} = \sum_{p=1}^P \left\{ \sum_{k=1}^m (t_k^{(p)} - z_k^{(p)})^2 / 2 \right\} \quad (3.17)$$

3.5.2.1 Initiation

Set the maximum error e_{\max} ; the maximum number of training epochs eps_{\max} ; the current number of training epochs $\text{Eps} = 0$; the learning rate ε ; and the initial values of the connection weights.

3.5.2.2 Input

Select one data pair $[x^{(p)}, t^{(p)}]$ from the training data set as current input and desired output, and set $\text{Eps} = \text{Eps} + 1$.

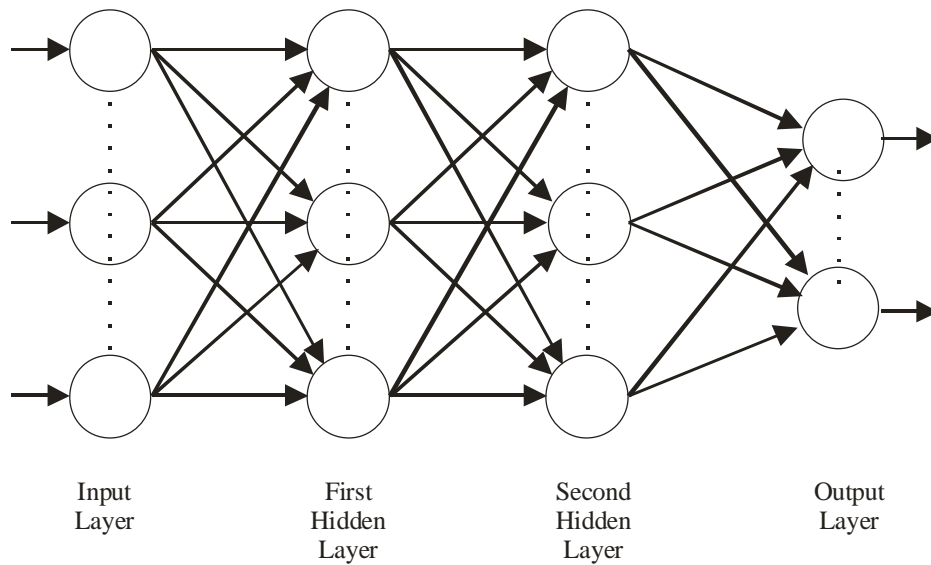


Figure 3.7 A structure of a multi-layer perceptron with two hidden layers (bias omitted for clarity of the illustration).

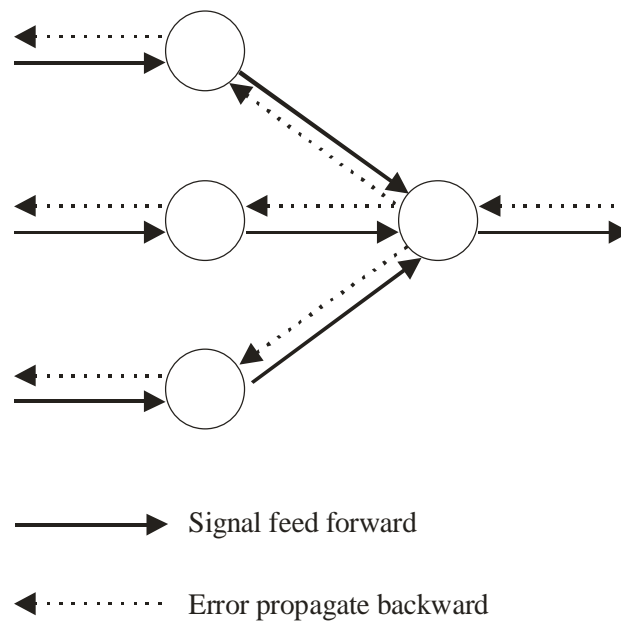


Figure 3.8 Two basic signal flows in a multi-layer perceptron

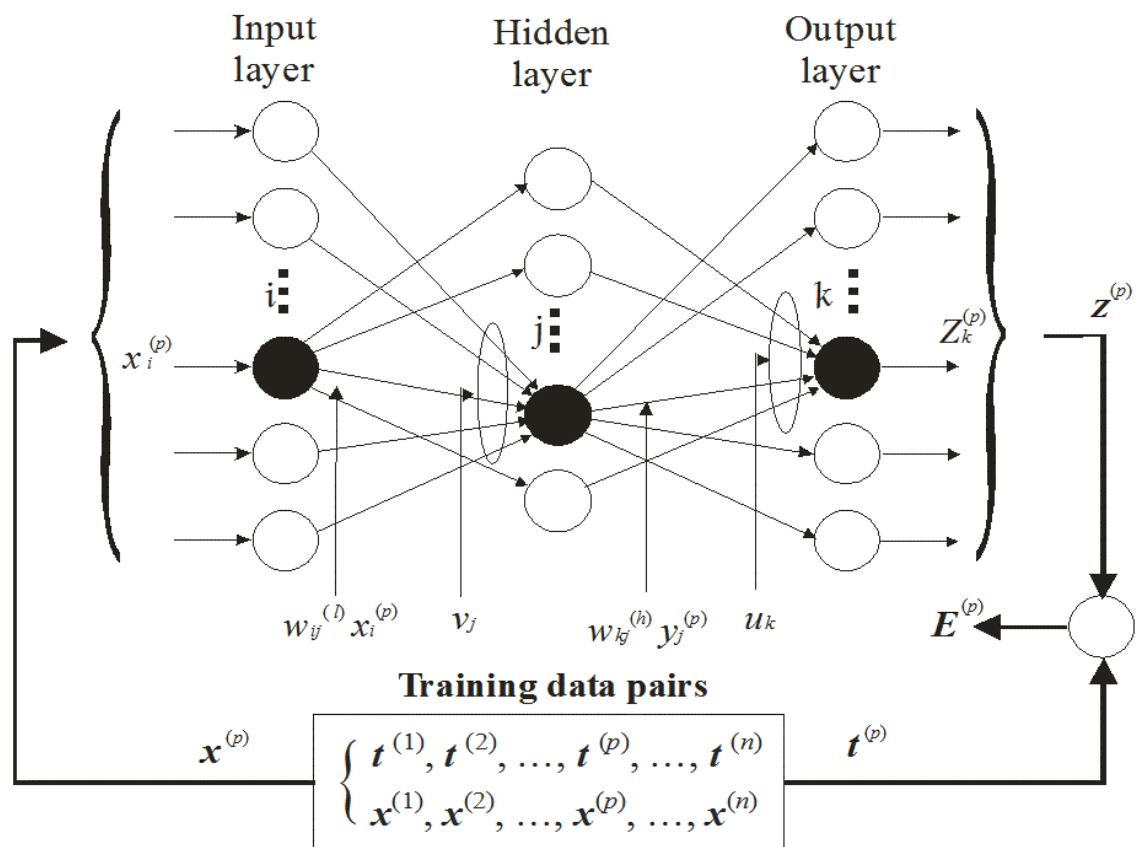


Figure 3.9 A three-layer back-propagation neural network with data feed forward through layers.

3.5.2.3 Signal feed forward

Calculate $y^{(p)}$ and $z^{(p)}$, outputs of the hidden layer and output layer, using sigmoid function.

$$\begin{cases} y_j^{(p)} = f(v_j) = 1 / (1 + \exp(-v_j)) \\ z_k^{(p)} = f(u_k) = 1 / (1 + \exp(-u_k)) \end{cases} \quad (3.18)$$

where

$$\begin{cases} v_j = \sum_i w_{ji}^{(l)} x_i^{(p)} \\ u_k = \sum_j w_{kj}^{(h)} y_j^{(p)} \end{cases} \quad (3.19)$$

3.5.2.4 Error propagate backward - part 1

Optimise the connection weights $w^{(h)}$ between the hidden layer and output layer

$$\begin{cases} \delta_k^{(h)} = (t_k^{(p)} - z_k^{(p)}) z_k^{(p)} (1 - z_k^{(p)}) \\ \Delta w_{kj}^{(h)} = \varepsilon \delta_k^{(h)} y_j^{(p)} \\ w_{kj}^{(h)} \leftarrow w_{kj}^{(h)} + \Delta w_{kj}^{(h)} \end{cases} \quad (3.20)$$

3.5.2.5 Error propagate backward- part 2

Optimise the connection weight $w^{(l)}$ base on the new $w^{(h)}$

$$\begin{cases} \delta_j^{(l)} = (\sum_k \delta_k^{(h)} w_{kj}^{(h)}) y_j^{(p)} (1 - y_j^{(p)}) \\ \Delta w_{ji}^{(l)} = \varepsilon \delta_j^{(l)} x_i^{(p)} \\ w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} + \Delta w_{ji}^{(l)} \end{cases} \quad (3.21)$$

The step above will be repeated to optimise the connection weights for any additional layers.

3.5.2.6 Termination condition

The training terminates when $E < e_{\max}$ or $Eps > eps_{\max}$, otherwise, repeat the steps 3.5.2.2 - 3.5.2.6

3.5.3 Radial Basis Function Network (RBFN)

Radial Basis Function Network (RBFN) (Lucks & Oki, 1999; Poggio, 1994) is a two-layer feed forward neural network that uses Gaussian function in the hidden nodes. Each node reacts to input in a small local space near its Gaussian function's centre. The output node uses a weighted sum function on the output of the hidden nodes for prediction problems and a sigmoid function is used for classification problems.

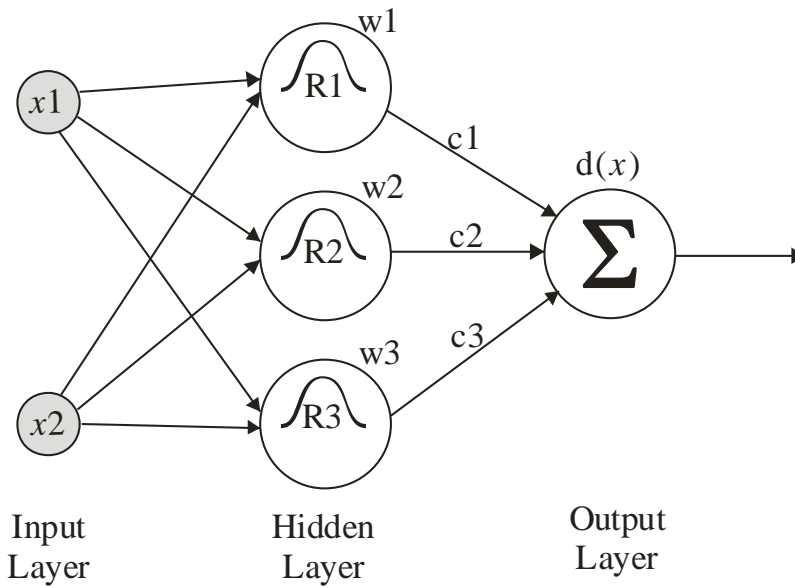


Figure 3.10 An illustration of the RBFN architecture with three radial basis functions for the prediction problems

Figure 3.10 shows the architecture of a RBFN with three hidden nodes.

$R_i(x)$ is a Gaussian function as shown below:

$$R_i(x) = \exp \left(- \frac{\|x - u_i\|^2}{2\sigma_i^2} \right) \quad (3.22)$$

where x is the input vector, u_i is the centre of a Gaussian function and $R_i(x)$ is the activation level of the i^{th} hidden node. Unlike MLP, there is no connection weight between the input layer and the hidden layer.

The closer the input vectors to the centre of the Gaussian function, the higher the output.

The output is the weighted sum of the output values from each Gaussian function as shown below:

$$d(\mathbf{x}) = \sum_{i=1}^H c_i w_i = \sum_{i=1}^H c_i R_i(\mathbf{x}) \quad (3.23)$$

where c_i is the output value associated with the i^{th} receptive field. It can also be explained as the connection weight between the i^{th} receptive field and the output unit.

3.6 Fuzzy Logic Systems

3.6.1 A Brief History

In 1965, Lotfi Zadeh introduced the concept of the fuzzy set (Zadeh, 1965), which defines the elements in a set with degrees of membership. The idea is that it is more efficient to tell the fan to spin faster when the computer gets warm than to have a set of rules to set the fan's spin speed at each temperature.

In 1975, Ebrahim Mamdani proposed a fuzzy inference system (FIS) (Mamdani & Assilian, 1975) based on Zadeh's 1973 fuzzy logic paper (Zadeh, 1973) to control a steam engine and boiler set using fuzzy set as the consequent function of the rule. This type of fuzzy rule has hence been referred to as the "Zadeh-Mamdani" (ZM) fuzzy rule.

In 1985, Takagi and Sugeno proposed a fuzzy inference system that is similar to ZM with the same antecedent but the output is a linear function instead of fuzzy set. This type of fuzzy rule is here on referred to as "Takagi-Sugeno" (TS) fuzzy rule.

The rules used in FIS can easily be interpreted by researchers and human knowledge can also be inserted into the system. This makes FIS an open system

that experts can understand and interact with, unlike many neural network models that acts like a black box which only allows experts to view the system's input and output.

The details of fuzzy set, membership function and fuzzy operation are explained below:

3.6.2 Fuzzy Sets and Membership Functions (Kasabov, 1996; Wang, 1994)

If X denotes a universal set, a fuzzy set A is defined by a membership function $\mu_A: X \rightarrow [0, 1]$ which describes the degree of membership of the elements of A . Higher value represents a higher level of membership degrees. Examples of commonly used membership functions are shown below:

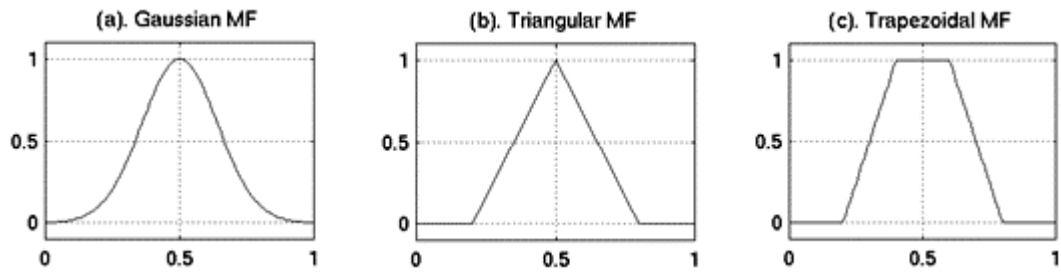


Figure 3.11 Examples of fuzzy membership functions

3.6.2.1 Gaussian membership function

The Gaussian membership function has two parameters σ (width) and c (centre), described as follows:

$$\mu(x) = \exp\left(\frac{-(x-c)^2}{\sigma}\right) \quad (3.24)$$

3.6.2.2 Triangular membership function

Triangular membership function has three parameters, a ; b ; c , described as follows:

$$\mu(x) = f(x; a, b, c) = \begin{cases} 0 & x \leq a \\ \frac{x - a}{b - a} & a \leq x \leq b \\ \frac{c - x}{c - b} & b \leq x \leq c \\ 0 & c \leq x \end{cases}$$

The parameters a and c define the edge of the triangle function and the parameter b locates the peak.

3.6.2.3 Trapezoidal membership function

Trapezoidal membership depends on four parameters, a ; b ; c ; d , given by

$$\mu(x) = f(x; a, b, c, d) = \begin{cases} 0 & x \leq a \\ \frac{x - a}{b - a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d - x}{d - c} & c \leq x \leq d \\ 0 & d \leq x \end{cases}$$

The two parameters, a and d , locate the “feet” of the trapezoid and the parameters b and c locate the “shoulders”.

3.6.3 Operations

Let μ_A and μ_B be two membership functions that define two fuzzy sets, A and B respectively. Four fuzzy operations are outlined as follows:

3.6.3.1 Subset

A is contained in B or A is a subset of B, denoted by

$$A \subseteq B \text{ if } \mu_A(x) \leq \mu_B(x) \quad \forall x \in X$$

or

$$A \subset B \text{ if } \mu_A(x) < \mu_B(x) \quad \forall x \in X$$

3.6.3.2 Complement, Negation

The membership function $\mu_{\bar{A}}(x)$ of the complement of A (denoted by \bar{A}) is defined by:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad \forall x \in X$$

The relative complement of A with respect to B is defined by:

$$\mu_{\bar{A}B}(x) = \mu_B(x) - \mu_A(x) \quad \forall x \in X \text{ if } \mu_B(x) > \mu_A(x)$$

3.6.3.3 Intersection

The intersection of A and B is defined by:

$$A \cap B = \{x | x \in A \wedge x \in B\} \quad \forall x \in X$$

Extreme operator:

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \min\{\mu_A(x), \mu_B(x)\} \quad \forall x \in X$$

Product operator:

$$\mu_{A \cap B}(x) = \mu_A(x) \mu_B(x) \quad \forall x \in X$$

3.6.3.4 Union

The union of A and B is defined by:

$$A \cup B = \{x | x \in A \vee x \in B\} \quad \forall x \in X$$

Extreme operator:

$$\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) = \max\{\mu_A(x), \mu_B(x)\} \quad \forall x \in X$$

Sum operator:

$$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \mu_B(x) \quad \forall x \in X$$

3.6.4 Fuzzy Relations

A relation represents the presence or absence of association, interaction or interconnection between the elements of two or more sets.

A fuzzy relation $R(x, y)$ is a fuzzy subset of $X \times Y$

For membership function $\mu(x, y)$

$$R = \{\mu(x, y): X \times Y \rightarrow [0, 1]\}$$

or

$$R = \{ (x, y), \mu_R(x, y) \} = \cup (x, y) \mu_R(x, y).$$

A fuzzy relation $R(x_1, x_2, \dots, x_n)$ on sets X_1, X_2, \dots, X_n , is a fuzzy subset of $X_1 \times X_2 \times \dots \times X_n$.

$$R = \{\mu(x_1, x_2, \dots, x_n): X_1 \times X_2 \times \dots \times X_n \rightarrow [0, 1]\}.$$

or

$$R = \cup \{ (x_1, x_2, \dots, x_n) \mu_R(x_1, x_2, \dots, x_n) \}: X_1 \times X_2 \times \dots \times X_n \rightarrow [0, 1].$$

3.6.5 Fuzzy Composition

A composition relation of fuzzy relations $R(x, y)$ and $S(y, z)$ is a relation $C(x, z)$ obtained after applying relations R and S one after another.

Given:

$$R(x, y), \quad (x, y) \in X \times Y \quad , \quad R: X \times Y \rightarrow [0, 1],$$

$$S(y, z), \quad (y, z) \in Y \times Z \quad , \quad S: Y \times Z \rightarrow [0, 1],$$

Composition $C(x, z)$

Maxmin composition:

$$\mu_c(x, z) = \max\{\min(\mu_R(x, y), \mu_s(y, z))\}; \quad x \in X, y \in Y, z \in Z.$$

Max product composition:

$$\mu_c(x, z) = \max\{\mu_R(x, y) \cdot \mu_s(y, z)\}; \quad x \in X, y \in Y, z \in Z.$$

3.6.6 Fuzzy If-Then Rules

A fuzzy if-then rule assumes the form “if x is A then y is B ”, where A and B are linguistic values defined by fuzzy sets on universes of discourse X and Y respectively. The condition part of the rule, i.e. “ x is A ”, is called an antecedent or and the action part of the rule, i.e. “ y is B ”, is called a consequence.

Several types of fuzzy rules have been used up to now. Different fuzzy rules will result in different fuzzy inference systems. There are several kinds of fuzzy rules including:

3.6.6.1 Zadeh-Mamdani fuzzy rules:

A generalised form of the Zadeh-Mamdani fuzzy rules is:

if x_1 is A_1 and x_2 is A_2 and ... and x_n is A_n , then y is B ,

where “ x_1 is A_1 ”, “ x_2 is A_2 ”, ... , “ x_n is A_n ” are n fuzzy propositions as the antecedent of the fuzzy rule; x_i , $i = 1, 2, \dots, n$, and y is a fuzzy variable defined over universes of discourse X_i , $i = 1, 2, \dots, n$, and Y respectively; and A_i , $i = 1, 2, \dots, n$, and B are fuzzy sets defined by their fuzzy membership functions $\mu_{A_i}: X_i \rightarrow [0, 1]$, $i = 1, 2, \dots, n$, and $\mu_B: Y \rightarrow [0, 1]$.

3.6.6.2 Fuzzy rules with degree of confidence:

As well as the simple form of Zadeh-Mamdani fuzzy rules described above, fuzzy rules having coefficients of uncertainty have often been used in practice.

3.6.6.3 Takagi-Sugeno fuzzy rules:

A generalised form of the Takagi-Sugeno fuzzy rules is:

if x_1 is A_1 and x_2 is A_2 and ... and x_n is A_n , then y is $f(x_1, x_2, \dots, x_n)$.

if $f(x_1, x_2, \dots, x_n)$ is C which is a crisp constant, we call it a zero order Takagi-Sugeno fuzzy rule; if function $f(x_1, x_2, \dots, x_n)$ is linear, the rule is called a first order Takagi-Sugeno fuzzy rule; and, such rules are called a high-order Takagi-Sugeno fuzzy rules if the non-linear function is used as the consequent function.

3.6.6.4 Generalised fuzzy production rules:

These kinds of rules can be seen as weighted rules, where each of the rules contributes to a certain degree to the final decision. Very often the fuzzy propositions in the antecedent of the rule are not equally important for the rule to infer an output value. A generalised fuzzy production rule with degrees of importance (DI_i) of the fuzzy propositions in the antecedent and certainty factors (CF) of the validity of the consequence has the form of:

if x_1 is $A_1 (DI_1)$ and x_2 is $A_2 (DI_2)$ and ... and x_n is $A_n (DI_n)$, then y is $B (CF)$

3.6.7 Fuzzy Inference Systems

The Figure 3.12 shows a block diagram of a basic fuzzy inference system, which is composed of four functional parts:

1. Fuzzification

Fuzzification is a process of finding the membership degrees to which input data belong to the fuzzy sets in the antecedent of a fuzzy rule.

2. Fuzzy rule set

This set contains a number of 'if-then' fuzzy rules.

3. Aggregation

Aggregation performs a fuzzy reasoning operation by aggregating the fuzzy values within the rules with the connective operations.

4. Defuzzification

Defuzzification is a process of calculating a single-output numerical value to a fuzzy output variable on the basis of the inferred resulting membership function for this variable.

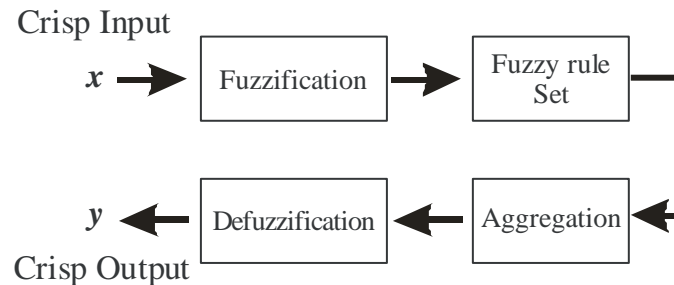


Figure 3.12 A block diagram of a basic fuzzy inference system

There are several types of fuzzy inference systems that have been used in various areas. The differences between them lie with the types of fuzzy inferences and the fuzzy if-then rules employed. The two most popular types of fuzzy inference systems are described as follows:

3.6.8 Mamdani Inference Engine (Zadeh, 1973)

Zadeh-Mamdani fuzzy rules are used here. The overall fuzzy output is derived by applying the union operation to the qualified fuzzy outputs. Each of the fuzzy output is equal to the minimum of the firing strength and the output membership function of each rule.

3.6.9 Takagi-Sugeno Inference Engine

Takagi-Sugeno (Takagi & Sugeno, 1985) fuzzy rules are used. The consequence of the rule is a linear function and the final output is the weighted average of the output of all rules. The consequence with higher order functions may be used in place of the linear function on more complex problems.

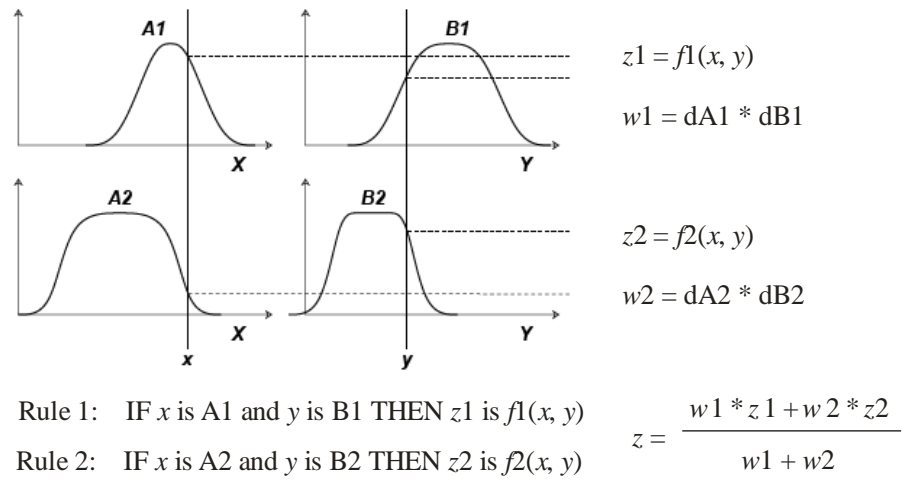


Figure 3.13 Takagi-Sugeno fuzzy inference system with two rules.

From Jane's paper for ANFIS (Jang, 1993)

3.7 ANFIS: Adaptive-Network-Based Fuzzy Inference System

ANFIS (Jang, 1993; Jang & Sun, 1995; Jang et al., 1997) is a fuzzy inference system that is capable of hybrid learning from both human knowledge, in the form of if-then rules, and from the dataset through input-output pairs. This model is one of the most widely used fuzzy inference systems and is, therefore, used as benchmark in some case studies in this thesis.

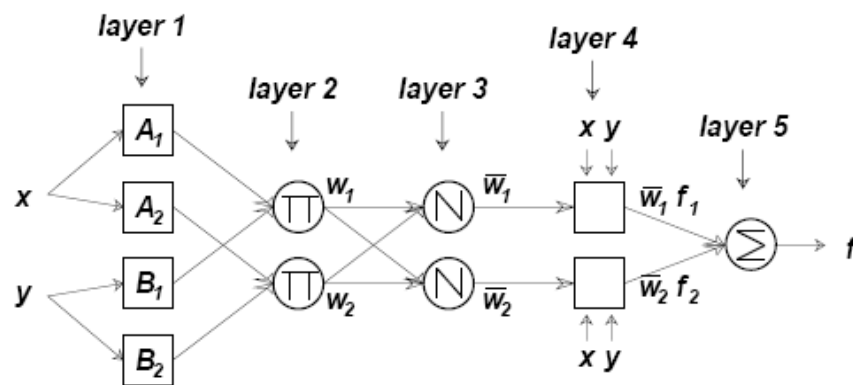


Figure 3.14 Illustration of the ANFIS system. From Jang's ANFIS paper (Jang, 1993).

ANFIS's system architecture has five layers that take crisp input, feeds through the layers and generates crisp output. The details of the algorithm are explained below:

3.7.1.1 Layer 1 - Fuzzification

Every node in layer 1 is a membership function, denoted:

$$O_i^1 = \mu A_i(x) \quad (3.25)$$

O_i^1 is the membership function of A_i and it specifies the degree of membership of x .

Various type of membership function can be used, the Gaussian function is commonly chosen as shown below:

$$\mu A_i(x) = \exp\left[-\left(\frac{x - C_i}{\sigma_i}\right)^2\right] \quad (3.26)$$

where x is the input, and c_i and σ_i are the parameters of the Gaussian function.

3.7.1.2 Layer 2 – the Π nodes

This layer multiplies the incoming signal from Layer 1 and sends the product out.

$$w_i = \mu A_i(x) \times \mu B_i(y), i=1,2 \quad (3.27)$$

3.7.1.3 Layer 3, the N nodes

The nodes on this layer calculate the ratio of the i^{th} rule's activation level to the sum of all the rules' firing strength.

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, i=1,2. \quad (3.28)$$

3.7.1.4 Layer 4 – Consequent layer

$$O_i^4 = \bar{w}_i f_i \quad (3.29)$$

\bar{w}_i is the output of Layer 3. f_i is the consequence of the rule, which is a linear function in the case of Takagi-Sugeno type rules.

3.7.1.5 Layer 5 – Crisp output

$$O_i^5 = \text{Overall output} = \sum_i \bar{w}_i f_i = \frac{\sum_i \bar{w}_i f_i}{\sum_i \bar{w}_i} \quad (3.30)$$

3.7.2 Evolving fuzzy inference systems

Many fuzzy inference systems (Angelov, 2002; Angelov & Filev, 2002, 2004; Kasabov, 2001; Kasabov & Song, 2002; Leng, McGinnity, & Prasad, 2005) have been proposed to optimise the fuzzy inference system through online learning. A review of similar systems was made by M. Watts in 2009 (Watts, 2009). This type of fuzzy system is capable of adapting to new data without losing the knowledge it has gained from prior learning.

These models are also known as online models. They are capable of learning one input vector at a time in a serial fashion and can adapt to new patterns. Most of these models are design for problems with data that is collected continuously, they usually discard the input vectors after processing and usually have low computation requirement. In many cases, the prediction accuracy may not be its only requirement, the speed of processing is also very important.

On contrast, Offline models learn from a set of input vectors at a time, sometimes the entire dataset. There are often no computation requirements and therefore more complex computation may be carried out to achieve optimal prediction accuracy.

These fuzzy inference systems are usually constructed through a two parts operation. First it defines or identifies the antecedent part of the fuzzy rules. This may be set manually to cover the entire problem space (Jang, 1993), or it may be based on online clustering such as Evolving Clustering Method (Kasabov & Song, 2002), subtractive clustering (Chiu, 1994) or others. Second, it constructs

or updates the consequent function through various online learning methods such as weighted recursive least square estimator. The system is updated with every new data pair $[x_i \ y_i]$. The update may be applied on either both antecedent and consequent part of the fuzzy rule or only the consequent part.

ANFIS, as described earlier in this chapter, is a classic example for non-clustering approach to construct its fuzzy inference system. It first populates the problem space with fuzzy rules using n membership functions per input variable. For each input data pair $[x_i \ y_i]$, the consequent function is updated with least-square methods and the membership function is updated using gradient descent method. This is called a hybrid learning system, combining both least-squares and gradient descent method.

Evolving Takagi-Sugeno system (eTS) (Angelov & Filev, 2004) is another example that uses clustering approach to construct its fuzzy inference system. This system is based on a combination of online clustering method called subtractive clustering (Chiu, 1994) and Kalman-Filter for its consequent function. Unlike ANFIS, where the rule structure is pre-defined, i.e. if grid method is used to define its rule structure and we use three fuzzy rules per input variable on a problem with 10 input variables, the problem space will be covered with 30 fuzzy membership functions. ETS allows the rule structure to be updated when necessary and therefore the number of rules may change during the course of learning. The number of rules expands as data points that are significantly different from existing data enter the system to accommodate new knowledge.

Two of the main methods proposed in this thesis are based on an existing evolving fuzzy inference system named “Dynamic Evolving Neural-Fuzzy Inference System”. This algorithm is therefore described below in details.

3.7.3 DENFIS: Dynamic Evolving Neural-Fuzzy Inference System

(Kasabov & Song, 2002)

DENFIS is a fuzzy inference system that is capable of online and offline learning through online clustering. This paper had over 180 citations on Google Scholar at the time of writing.

DENFIS starts by clustering the data and creates a fuzzy inference system that is based on the clusters. A maximum distance-based clustering algorithm, Evolving Clustering Method, is used to cluster the input data. Once the clusters are derived, a Takagi-Sugeno fuzzy rule is created for each cluster. These rules are then optimised through back-propagation method. For each prediction, m most activated rules are dynamically chosen to derive the final output. New rule sets can be inserted into or extracted from the model.

The DENFIS algorithm is described below:

3.7.3.1 ECM: Evolving Clustering Method (online)

ECM is a fast one-pass algorithm to find clusters within the input data. The algorithm is described below:

- Step 0: Create the initial cluster C_1 and set the position of the first training data as a cluster centre C_{c1} with the cluster radius R_{u1} .
- Step 1: If all training data has been processed, terminates the algorithm, otherwise calculate the distance between the current training input vector x_i and the cluster centre C_{cj} . $D_{ij} = ||x_i - C_{cj}||$, $j=1,2,\dots,n$.
- Step 2: If there is a cluster centre C_{cj} , where the $D_{ij} = ||x_i - C_{cj}||$, $j=1,2, \dots, n$, is equal to or less than the radius R_{uj} , then x_i is assumed to belong to cluster C_m and no new cluster is created and no existing cluster is updated. Go back to step 1.
- Step 3: Find a cluster C_a from all existing cluster centres with $S_{ij} = D_{ij} + R_{uj}$, $j = 1,2, \dots, n$, and select the cluster centre C_{ca} with the smallest S_{ia} :

$$S_{ia} = D_{ia} + Ru_a = \min\{S_{ij}\}, j = 1, 2, \dots, n.$$

- Step 4: If S_{ia} is greater than $2 \times D_{thr}$, then x_i does not belong to any of the existing clusters and a new cluster is created as described in Step 0 and the algorithm returns to Step 1.
- Step 5: If S_{ia} is not greater than $2 \times D_{thr}$, the cluster C_a is updated by moving Cc_a and enlarging the cluster radius Ru_a . The updated Ru_a^{new} is set to equal to $S_{ia} / 2$ and the new cluster centre Cc_a^{new} is set as follows:

$$Cc_a^{new} = x_i - \left((Cc_a - x_i) \times \frac{(S_{ia} / 2)}{D_{ia}} \right) \quad (3.31)$$

5. Repeat Step 1 to 5.

D_{thr} is the distance threshold of the cluster, which defines the maximum radius of the cluster. S_{ia} defines inverse the level how much the x_i belongs to cluster centres C_a . The smaller the S_{ia} , the more x_i belongs to cluster C_a .

Note that x_i can belong to multiple clusters as there may be overlapping of the cluster radius. Euclidean distance is used as the distance measuring method in ECM.

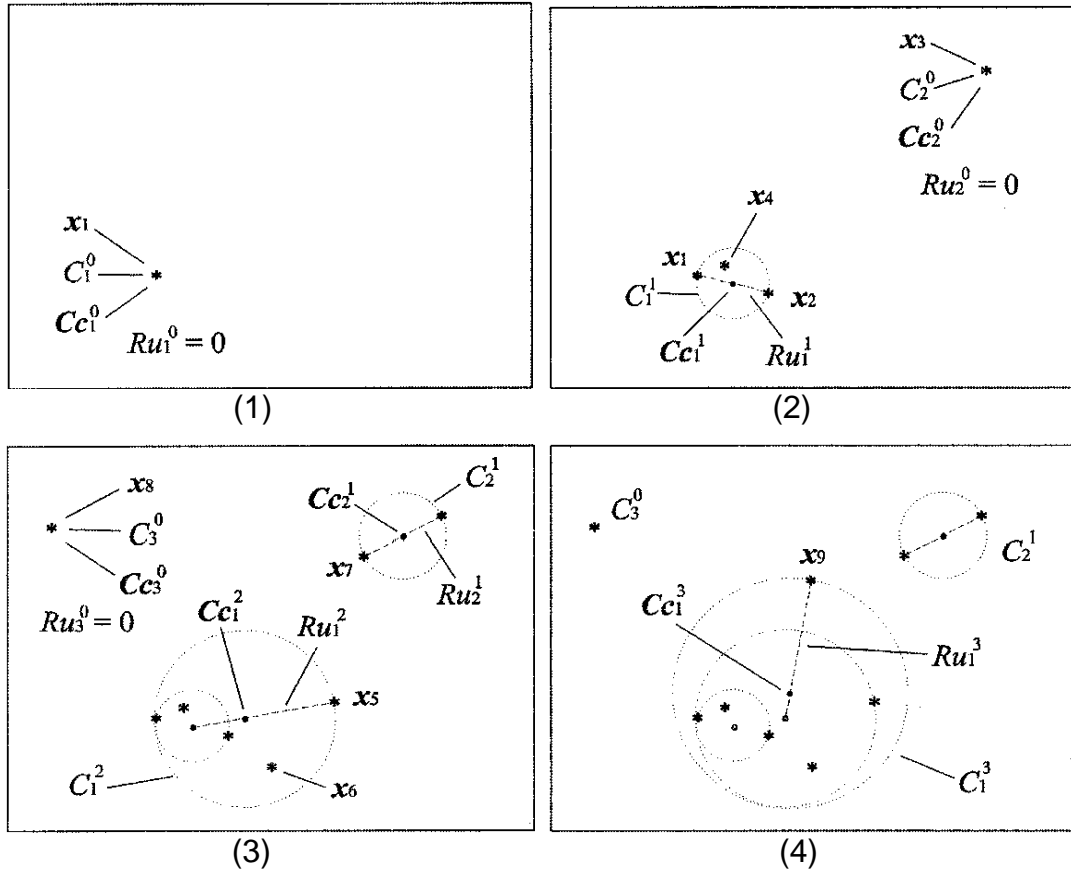


Figure 3.15 Example of ECM Clustering algorithm. x_i : input vector (*), Cc_j^k : cluster centre, C_j^k : cluster, Ru_j^k cluster radius (Kasabov & Song, 2002).

Figure 3.15 shows the step by step of ECM clustering process.

- (1) The initial cluster is created for the first input vector x_1 .
- (2) x_2 : update cluster $C_1^0 \rightarrow C_1^1$
 x_3 : create a new cluster C_2^0
 x_4 : belongs to C_1^1 , no action required.
- (3) x_5 : update cluster $C_1^1 \rightarrow C_1^2$
 x_6 : belongs C_1^2 , no action required
 x_7 : update cluster $C_2^0 \rightarrow C_2^1$
 x_8 : create a new cluster C_3^0
- (4) x_9 : update cluster $C_1^2 \rightarrow C_1^3$

ECM processes input vectors in a one-input-vector-at-a-time manner, and therefore the order of the input vectors being process affects the final output. This is evident in the way the first cluster is created for the first input vector. This

design was necessary since ECM is an online clustering method where data is made available one input vector at a time. However, this does not prove to be a significant problem in practice as the cluster centre may be slightly different based on the order of input vectors being process, when inspected closely by visualising the input vectors in each cluster, the input vectors were very similar, as ECM originally intended.

3.7.3.2 ECMc: Evolving Clustering Method (Offline)

Ideally a cluster centre should be positioned at the centre of the gravity among all input vectors in the cluster, as it is what the term “cluster centre” implies. As shown in Figure 3.15, the cluster centre does not necessarily fall on the centre of the gravity of the cluster in ECM online.

An offline version of ECM (ECMc) was proposed to address the problem of cluster centres not positioned at the centre of gravity. It optimises the cluster centres after the clusters are derived from the online ECM algorithm and moved the cluster centres to the centre of the gravity.

A constrained optimisation method was applied to the clusters that minimise the following objective function:

$$J = \sum_{j=1}^n J_j = \sum_{j=1}^n \left(\sum_{x_i \in C_j} \|x_i - Cc_j\| \right) \quad (3.32)$$

where $i=1,2, \dots, p$. p is the number of input vectors.

The constraints are defined below:

$$\|x_i - Cc_j\| \leq Dthr, \quad j=1,2, \dots, n \quad (3.33)$$

Once the cluster centres are updated, the input vectors are reallocated to the nearest cluster.

$$\begin{aligned} &\text{If } \|x_i - Cc_j\| \leq \|x_i - Cc_k\|, \quad \text{for each } j \neq k \\ &x_i \text{ belongs to } Cc_j, \text{ otherwise } x_i \text{ belongs to } Cc_k \end{aligned} \quad (3.34)$$

$$\mu(x) = mf(x, a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (3.35)$$

where b is the value of the cluster centre, $a=b-d \times Dthr$ and $c=b+ d \times Dthr$, $d=1.2-2$; $Dthr$ is the distance threshold, a clustering parameter from ECM or ECMc for limiting the maximum cluster size.

For an input vector, $x = [x_1 \ x_2 \ \dots \ x_q]$, the output of the system, y is the weighted average of each of the m activated rules as follows:

$$y = \frac{\sum_{i=1}^m w_i f_i(x_1, x_2, \dots, x_q)}{\sum_{i=1}^m w_i} \quad (3.36)$$

3.7.3.3 DENFIS online learning process

The consequence of the Takagi-Sugeno fuzzy rule is created and updated by a (weighted) least-square estimator as described in 3.4.1.2 The linear function is expressed as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_q x_q$$

The coefficient β is obtained through the following formula

$$\begin{aligned} \beta &= [b_0 \ b_1 \ \dots \ b_q]^T \\ b &= (A^T A)^{-1} A^T y \end{aligned} \quad (3.37)$$

or for the weighted version of the LSE.

$$b = (A^T W A)^{-1} A^T W y \quad (3.38)$$

where

$$A = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1q} \\ 1 & x_{21} & x_{22} & \cdots & x_{2q} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{p1} & x_{p2} & \cdots & x_{pq} \end{pmatrix}$$

and

$$y = [y_1 \ y_2 \ \dots \ y_p]^T$$

and W is a diagonal matrix:

$$W = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & w_p \end{pmatrix}$$

Equation(3.37) and (3.38) can be rewritten as

$$\text{LSE} \quad \begin{cases} P = (A^T A)^{-1} \\ b = P A^T y \end{cases} \quad (3.39)$$

$$\text{Weighted LSE} \quad \begin{cases} P_w = (A^T W A)^{-1} \\ b_w = P_w A^T W y \end{cases} \quad (3.40)$$

In the DENFIS online mode, the weighted recursive LSE is used with the following equation:

$$\text{WR LSE} \quad \begin{cases} b_{k+1} = b_k + w_{k+1} P_{k+1} a_{k+1} (y_{k+1} - a_{k+1}^T b_k) \\ P_{k+1} = \frac{1}{\lambda} \left(P_k - \frac{w_{k+1} P_k a_{k+1} a_{k+1}^T P_k}{\lambda + a_{k+1}^T P_k a_{k+1}} \right) \end{cases} \quad (3.41)$$

$k = n, n+1, \dots, p-1$

The forgetting factor λ is set between 0.8 and 1.

The DENFIS online model learning procedure is explained below:

1. Perform ECM clustering on the initial set of data n_0 to obtain M clusters
2. For every cluster C_i , find p_i data points are closest to C_i , $i = 1, 2, \dots, M$;

3. Create a fuzzy rule for each cluster. The antecedent of the fuzzy rule is the cluster centre. The consequent function is created using equation (3.39) or (3.40). The distance between p_i and the cluster centre is used to create the weight matrix.
4. The size of p_i is a model training parameter. It defines the number of data points used to derive the consequent function of the fuzzy rules.

As new input vector enters the system, new fuzzy rules may be created and some rules updated. A new fuzzy rule is created if a new cluster is found in ECM. If no new clusters are created, one or more fuzzy rules are updated by using equation **Error! Reference source not found..**

For each input vector, the DENFIS online model dynamically creates a Takagi-Sugeno fuzzy inference system using m activated rules. m is a model training parameter that should be adjusted based on the characteristic of the problem. The rules are chosen based on the position of the input vector. Since the rules are updated constantly, two input vectors with the same values at different time point may have different inferences as the fuzzy rule may have been updated before the second input vector entered the system.

3.7.3.4 DENFIS offline learning process

The offline version of DENFIS differs from the online version only in the clustering method where the ECM algorithm is replaced with ECMc. Since ECM and ECMc are both unsupervised clustering methods, there is no supervised learning in the offline version of DENFIS to update the fuzzy rules' membership function parameters.

3.8 Generalisation Error Estimation and Model Selection

Measuring the generalisation error of a model is one of the key components in this thesis, as it determines how the model is expected to perform

on future data. There are several commonly used re-sampling methods for measuring generalisation error.

3.8.1 Holdout Method

In this method, the data is randomly split into training and testing data once at a fixed ratio p , usually at 66.67% or 50%. The model is trained with the training data and then tested on the test data (McLachlan, 1992).

This method builds only one model and therefore requires less computation than other methods. However, this method reserves a large portion of the data to test the model's performance and, therefore, if the dataset is small or imbalanced, there may not be enough input vectors in the training dataset to properly represent the problem. This may lead to an overestimation of the generalisation error.

3.8.2 N-Fold Cross Validation Method

This method randomly splits data into n pieces of near or equal size without replacement, i.e. in each fold, one input vector can only exist in either training data or testing data. The training and testing procedure is repeated n times, each time one of the pieces is used as the testing dataset and the remaining pieces are aggregated as the training data. Different pieces are used as the testing dataset each time and the same piece is never reused as the testing dataset.

The prediction error is calculated for each piece and averaged across n pieces to obtain the final generalisation error (Kohavi, 1995a).

3.8.3 Leave-one-Out Cross Validation

This is an extreme case of n -fold cross validation. Leave-one-Out Cross Validation (LOOCV) is equivalent to n -fold cross validation where n is the number of input vectors in the dataset (Lachenbruch & Mickey, 1968).

This method uses $n-1$ input vectors for training and the remaining input vector for testing. It has the smallest bias (Martens & Dardenne, 1998), making it the most suitable re-sampling method for problems with a small number of input vectors.

LOOCV requires n models to be trained using the largest amount of input vectors, making it very computationally expensive and, therefore, may not be suitable for large datasets.

3.8.4 Monte Carlo Cross Validation (MCCV)

This method is the random repetitive version of the holdout method that repeats holdout v times and allows the input vectors to be reused (Picard & Cook, 1984). The data is split v times with the ratio of p (e.g. 1:10) as in the holdout method. There is no limit to the number of splits, it can be 10, 100, 1000 or more.

The 10-fold cross validation method is widely used in this thesis to measure the generalisation error for medium to large datasets, primarily due to its good balance between computational complexity and level of bias.

As for small datasets, leave-one-out cross validation is used to ensure sufficient numbers of input vectors in the training dataset to avoid over estimation of generalisation error.

3.9 Conclusion

This chapter reviews methods and techniques that are used or highly related to the research in this PhD study. The researches and developments carried out in this PhD study are either improvements made on previous studies reviewed in this chapter or integrate existing methods in a new way to achieve better results.

DyNFIS and MUFIS both improve on DENFIS offline method to allow more emphasis on the problem subspaces. Multi-Model System and the personalised

regression model, as proposed in chapter 6 and 8, integrates regression methods with other methods to allow better representation of the problem subspaces.

In the next chapter, a novel method is proposed to address the issue with unique problem subspaces by improving the DENFIS offline model to allow supervised optimisation on the membership functions and use more complex membership function.

CHAPTER 4 DYNFIS – AN IMPROVED DYNAMIC NEURAL FUZZY INFERENCE SYSTEM FOR LOCAL MODELLING

4.1 Introduction

In this chapter, an improved DENFIS offline model, denoted DyNFIS, is proposed with the following two improvements over the original DENFIS offline model, described as follows:

4.1.1 Improvement 1: Replace Triangular MF with Gaussian MF

The triangular MF used in the original DENFIS was chosen for its low computational requirements, which was important for an online model, but much less so for an offline model as the computational speed is no longer one of the primary issues.

Triangular MF is the simplest form of membership function with just three parameters. It has low computational requirements and provides coverage for the majority of the space. The peak position and the two feet define the triangular MF as shown in Figure 4.1. The degree of membership decline at a fixed rate as the value moves away from the centre and any value outside the triangle has the degree of membership of zero.

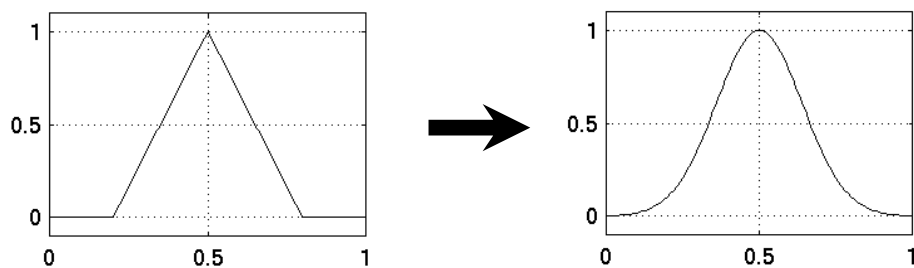


Figure 4.1 Triangular membership function replaced with Gaussian membership function

The Gaussian MF is widely used in many computation methods that involve coverage of problem space such as Evolving Self-Organising Map (Da & Kasabov, 2000), Radial-Basis Function network(Lucks & Oki, 1999). Its coverage of problem space expands infinitely as the degree of membership decreases gradually through the bell curve from the peak at different rate depends on the parameter of the function and it never reaches zero.

The use of different membership functions may not have significant impact on the model's prediction accuracy, but in terms of knowledge discovery, Gaussian membership function makes the rules more accurately represent the model.

The fuzzy rule is often written in simplified form to make it easier to read.

Here is an example of a set of 3 activated rules chosen to make a prediction for input vector x .

Rule 1:

if X_1 is about 0.19 and X_2 is about 0.07 and X_3 is about 0.43
and X_4 is about 0.63 then

$$y = 1.17 + 0.38 * X_1 + 0.56 * X_2 + 0.03 * X_3 + 0.48 * X_4$$

Rule 2:

if X_1 is about 0.21 and X_2 is about 0.14 and X_3 is about 0.50
and X_4 is about 0.69 then

$$y = 1.19 + 0.46 * X_1 + 0.57 * X_2 + 0.09 * X_3 + 0.39 * X_4$$

Rule 3:

if X_1 is about 0.14 and X_2 is about 0.27 and X_3 is about 0.62
and X_4 is about 0.69 then

$$y = 1.21 + 0.35 * X_1 + 0.62 * X_2 - 0.06 * X_3 + 0.49 * X_4$$

The term “about” is a simplified expression for a degree of membership. It simply means two values are similar. When presented using triangular membership function, due to its fixed width, the value outside a fixed range is not “about” certain value. This range is fixed and therefore loses the part of meaning of fuzziness enclosed in the rule. Gaussian MF provides a more accurate represents the term “about” in mathematical form as it captures the fuzziness in our language.

4.1.2 Improvement 2: Supervised Learning with Back-Propagation to Optimise Both Consequent and Membership Functions

DENFIS offline model's supervised learning is limited to the consequent part of the fuzzy rules. The membership functions are fixed once they are created based on an unsupervised clustering method (ECM). It seems only logical to optimise them with a supervised optimisation method, as the offline module does not need to adapt to new data and computational speed is not a critical issue as in the online model. By applying additional supervised learning on the membership function parameters, the prediction accuracy of the model may be further improved. This also allows the rules to better represent the problem as both input and output data are used.

4.2 Algorithm Description

The DyNFIS offline learning process is outlined as follows:

1. Cluster the data to find n cluster centres using Offline Evolving Clustering Method.
2. Create a fuzzy rule for each cluster
 - a. Antecedent of the fuzzy rule is created based on the cluster centre.

- b. Consequence of the fuzzy rule is a linear function trained with the input vectors that belong to the clusters. (Takagi-Sugeno fuzzy rule)
3. For each training input vector, derive the output from m activated rules
4. Using back-propagation to minimise the error by adjusting the membership function and the consequence of the activated rules using equations (4.5) to (4.11)

Consider data that is composed of N data pairs with P input variables and one output variable $\{[x_{i1}, x_{i2}, \dots, x_{ip}], y_i\}$, $i = \{1, 2, \dots, N\}$, $j = \{1, 2, \dots, P\}$ M fuzzy rules are defined initially through a clustering process (ECM), the i^{th} rule has the form of:

R_i : If x_1 is about F_{i1} and x_2 is about F_{i2} ... x_p is about F_{ip} and then $y = f(x)$

$$y = \beta + x_1\beta_1 + x_2\beta_2 + \dots + x_p\beta_p \quad (4.1)$$

F_{ij} are the fuzzy sets defined by the following Gaussian type membership function (MF):

$$\text{Gaussian MF} = \alpha \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right] \quad (4.2)$$

Using the modified centre average defuzzification procedure, the output value of the system can be calculated for an input vector $x_i = [x_1, x_2 \dots x_p]$ as follows:

$$f(x_i) = \frac{\sum_{l=1}^M n_l \prod_{j=1}^p \alpha_{lj} \exp\left(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2}\right)}{\sum_{l=1}^M \prod_{j=1}^p \alpha_{lj} \exp\left(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2}\right)} \quad (4.3)$$

Suppose DyNFIS is given a training input-output data pair $[x_i, t_i]$. The system minimises the following objective function:

$$E = \frac{1}{2} [f(x_i) - t_i]^2 \quad (4.4)$$

The back-propagation (steepest descent) algorithm is used to obtain equation (4.5) - (4.11) for the optimisation of the parameters n_l , α_{lj} , m_{lj} , σ_{lj} and β_l

$$m_{lj}(k+1) = m_{lj}(k) - \frac{\eta_m \Phi(x_i)[f^{(k)}(x_i) - t_i]}{\sigma_{lj}^2(k)} \times [n_l(k) - f^{(k)}(x_i)][x_{ij} - m_{lj}(k)] \quad (4.5)$$

$$n_l(k+1) = n_l(k) - \eta_n \varphi(x_i)[f^{(k)}(x_i) - t_i] \quad (4.6)$$

$$\alpha_{lj}(k+1) = \alpha_{lj}(k) - \frac{\eta_\alpha \varphi(x_i)[f^{(k)}(x_i) - t_i]}{\alpha_{lj}(k)} \times [n_l(k) - f^{(k)}(x_i)] \quad (4.7)$$

$$\sigma_{lj}(k+1) = \sigma_{lj}(k) - \frac{\eta_\sigma \varphi(x_i)[f^{(k)}(x_i) - t_i]}{\sigma_{lj}^2} \times [n_l(k) - f^{(k)}(x_i)][x_{ij} - m_{lj}(k)]^2 \quad (4.8)$$

$$\varphi(x_i) = \frac{\prod_{j=1}^p \alpha_{lj} \exp(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2})}{\sum_{l=1}^M \prod_{j=1}^p \alpha_{lj} \exp(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2})} \quad (4.9)$$

$$\beta_{l0}(k+1) = \beta_l(k) - \eta_\beta \varphi(x_i)[f^{(k)}(x_i) - t_i] \quad (4.10)$$

$$\beta_{lj}(k+1) = \beta_l(k) - \eta_\beta \varphi(x_i)[f^{(k)}(x_i) - t_i]x_{ij} \quad (4.11)$$

where η_m , η_n , η_α , η_σ and η_β are the learning rates for updating the parameters: n_l , α_{lj} , m_{lj} , σ_{lj} and β_l respectively.

In the DyNFIS algorithm, the following indexes are used:

- training data points: $i=1,2,\dots,N$
- input variables: $j=1,2,\dots,P$
- fuzzy rules: $l=1,2,\dots,M$
- training iterations: $k=1,2,\dots$

4.3 Knowledge Extraction

The knowledge extracted from DyNFIS is in the form of easy to understand fuzzy rules, just like the ones extracted from DENFIS, it identifies different groups of input vectors that are unique and should be treated differently. E.g. “IF x_1 is about a_1 , x_2 is about a_2 , x_3 is about a_3 then $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_q x_q$. The term “about” can be more precisely described as the degree of membership.

In DENFIS, the rules are entirely based on the clustering results of either the online or offline ECM. There is no supervised learning applied to these rules to adjust their membership functions’ parameters. DyNFIS, on the other hand, applies back-propagation after the rules are created to adjust its membership functions’ parameters to optimise the accuracy of the prediction. Therefore, the rules in DyNFIS is expected to fit the data better, due to the use of output data in supervised learning and therefore provide more accurate knowledge to the researcher.

4.4 Benchmark Dataset Case Study: Mackey-Glass Dataset

The DyNFIS was applied to the on Mackey-Glass dataset (Mackey & Glass, 1977), which has been widely used as a benchmark in the area of neural networks, fuzzy systems and hybrid systems for time series prediction problems. The dataset was created with the following differential equation:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \quad (4.12)$$

The integer time points for the above equation were obtained using the fourth-order Runge-Kutta method. Here we assume that the time step is 0.1; $x(0) = 1.2$; $\tau = 17$; and $x(t) = 0$ for $t < 0$. (Hornik et al., 1989; Vapnik, 1998)

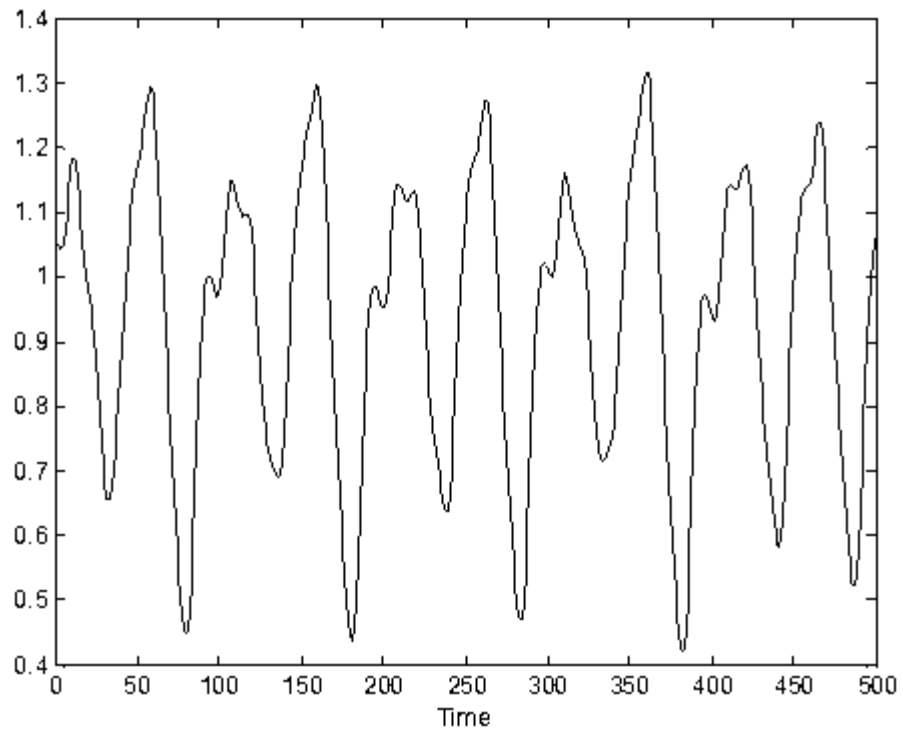


Figure 4.2 Mackey-Glass dataset with 500 input vectors for testing.

For the $t+85$ benchmark dataset, 3000 data points, from $t = 201$ to 3200, were extracted as the training data and 500 data points, from $t = 5001$ to 5500, were extracted as the testing data.

The results are shown in Table 4.1 with other published results for comparison. Non-Dimensional Error Index (NDEI) is used as the measure of quality, which is equivalent to Root Mean Square Error (RMSE) divided by the standard deviation of the training output.

Table 4.1

Prediction accuracy comparison of several offline algorithms on t+85 Mackey-Glass dataset

Methods	Neurons / Rules	Epochs	Training NDEI	Testing NDEI
MLP-BP	60	500	0.021	0.022
ANFIS	81	50	0.032	0.033
ANFIS	81	500	0.024	0.025
DENFIS I (TS)	116	2	0.068	0.068
DENFIS I (TS)	883	2	0.023	0.019
DENFIS II (MLP)	58	60	0.020	0.020
DyNFIS (TS)	91	500	0.017	0.018

Note: Results extracted from (Song, 2001)

As shown in Table 4.1, DyNFIS has no difficulty in solving this difficult problem, the t+85 Mackey-Glass prediction.

DENFIS I (TS), due to lack of optimisation of the fuzzy rules, can finish the training in only two epochs since it only needs to optimise the consequences of the fuzzy rules. This may be ideal for online applications, but in order to achieve better accuracy, it needs to use very large number of fuzzy rules to allow very low level representation of the problem subspaces.

DENFIS II (MLP) uses computationally expensive MLP as its consequence, where many iterations, 10s or 100s, of supervised training is carried out within the MLP optimisation algorithm in each of the epochs in DENFIS. Its performance is still limited due to the un-optimised fuzzy membership function parameters, even with the use of MLP consequence.

Overall, DyNFIS is not much more computationally expensive than other algorithms but was able to achieve better prediction accuracy.

4.5 Application of DyNFIS in Neural Network Forecasting

Competition (NN3) for Time Series Prediction

The proposed algorithm was entered in the NN3 Neural Network Forecasting competition (Crone, 2006). The result was submitted to NN3 under the name of DENFIS due to an error, the prediction was really made using DyNFIS, not DENFIS.

NN3's time-series data are from homogeneous populations of empirical business time series problems.

The 11 time-series data and the predicted values are shown below:

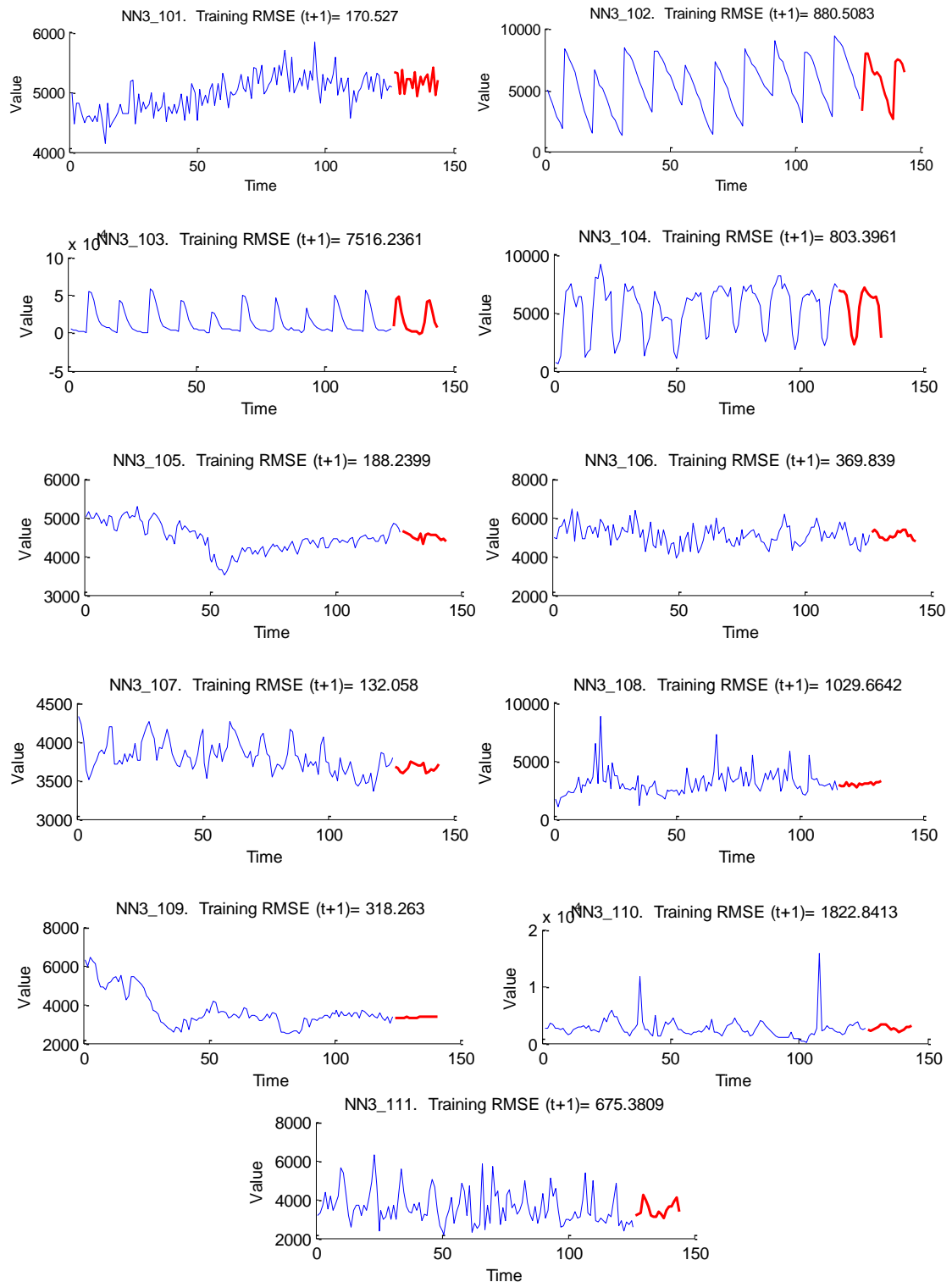


Figure 4.3 The values of the 11 time-series problems in the NN3 competition. Blue lines are the training data and red lines are the predicted values for t+1 to t+18.

The objective of this competition is to predict the next one to eighteen time points (t+1 to t+18). The predictions were made with the following setup:

1. One model was created for each prediction and therefore 88 (11 time-series and 8 time points each) models were created for this competition.
2. For each prediction, three models were created using following three range of data as input:
 - a. $t-2$ to t
 - b. $t-5$ to t
 - c. $t-8$ to t
3. The model that gives lowest RMSE is then used as input for the prediction.
4. DyNFIS default parameters were used for each model. No model training parameter optimisations were applied during training.

Due to time and resource limitations, full optimisation for each model was not carried out and a single set of parameters was used to train all 88 models. The only optimisation done for each model was on the input data where three models were trained with data using “ t to $t-2$ ”, “ t to $t-5$ ” and “ t to $t-8$ ” as their input variables. The model that had the lowest training error was used for final prediction.

DyNFIS does not have an automatic parameter tuning feature, which was available to several better performing methods in NN3 competition, either embedded or wrapped around the algorithm. DyNFIS needs to be optimised manually in order to achieve optimal prediction results. The current method of using a fixed set of features on all time-series data put a significant limit on its potential though it was unavoidable due to the scope and limitation of this PhD study. The implementation of DyNFIS was not designed for distributed computing and, therefore, the use of grid computing facilities was not possible.

On a single computer, full optimisation of the parameters was not practical with the amount of available resources due to the high computational complexity. The DyNFIS model has eight parameters and high precision optimisation is required for some of these parameters. If we optimise the parameters using the

Genetic Algorithm (GA) (Davis & Mitchell, 1991) or Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), the search space is simply too wide.

A rough estimate for GA optimisation, with 100 generations and 24 populations, showed that this process would create 211,200 models for this competition. Each model may take 60 seconds to complete on a single high performing computer and therefore it would take approximately 74 days to complete the predictions without optimising the input data. It was simply not practical at the time of preparing for the NN3 competition.

Under the circumstances, DyNFIS was still able to outperform most of the other prediction algorithms and achieved 10th place among 90 submissions in the 11 time-series competition in NN3. This shows that DyNFIS is a stable and accurate method, even without in-depth parameter optimisation.

4.6 Conclusion

This chapter presents an improved version of DENFIS offline algorithm, DyNFIS, which applies additional back-propagation learning on the membership functions and replaced triangular MF with Gaussian MF.

The algorithm composes a Takagi-Sugeno inference system using m most activated fuzzy rules to derive the output for a given input vector. The proposed system demonstrates superiority when compared with other global models including MLP, ANFIS and the original DENFIS offline system on benchmark data. It also demonstrated its performance in the NN3 competition with minimal optimisation and achieved good result.

This algorithm addresses the following real world data modelling issues.

- Unique problem subspaces

DyNFIS creates fuzzy rules based on clustering and supervised learning on the training data to ensure that unique problem spaces are well represented.

- Outliers

The outliers are expected to be positioned outside the normal clusters since they are expected to be very different from the majority and with ECM clustering ensures that input vectors without any support from other input vectors will not form a cluster. This minimises the influence of outliers for the majority of input vectors since no rule will be created for them.

4.7 Discussion

DyNFIS allows different linear model to be used in each problem subspace and then integrates them together through a fuzzy inference system. Each problem subspace is unique and can be very different from another, it is likely that the linear model is not suitable for some problem subspaces and perhaps non-linear or other models would be better suited.

The next stage of research is to allow different types of models to be used in a fuzzy inference system and allow for more suitable models to be applied based on the characteristic of the problem subspace.

In the next chapter, a novel fuzzy inference system is proposed, which extends DyNFIS to allow multiple fuzzy rule types to be integrated in a single fuzzy inference system.

CHAPTER 5 MUFIS: A NOVEL NEURO-FUZZY INFERENCE SYSTEM USING MULTIPLE TYPES OF FUZZY RULES

In this chapter, a novel fuzzy inference system is proposed, referred to as “MUFIS: A Neuro-Fuzzy Inference System Using Multiple Types of Fuzzy Rules”, which is an extension of the DyNFIS algorithm. This method was published in IEEE International Conference on Fuzzy Systems in 2008 (Hwang et al., 2008). The advantage of this method over DyNFIS is that it allows different types of fuzzy rules to be aggregated for a single prediction.

An implementation of this system using both ZM and TS type fuzzy rules is demonstrated on the Mackey-Glass benchmark dataset (Mackey & Glass, 1977) and then on a real medical dataset for renal function prediction (Levey et al., 1999; Levey et al., 2007).

5.1 Algorithm Description

MUFIS is a dynamic neuro-fuzzy inference system using both Zadeh-Mamdani type fuzzy rules and Takagi-Sugeno type fuzzy rules. Gaussian membership functions are used in each fuzzy rule for the antecedent and either a Gaussian membership function or a linear function is used for the consequence depending on the type of fuzzy inference engine.

In addition, unlike DyNFIS, where the consequent function is derived using input vectors within the cluster, the consequence here is the locally optimised global function.

The general MUFIS algorithm is shown below:

1. Cluster the data to find n cluster centres. (A number of clustering algorithms can be used including but not limited to K-Means (MacQueen, 1967) and the Evolving Clustering Method (ECM) (Kasabov & Song, 2002).

2. Assign either ZM or TS type fuzzy rules to each cluster, based on the cluster's characteristic (described later). The cluster centre is used as the centre of the fuzzy membership functions.
3. Create an FIS system
4. Apply an optimisation algorithm (Steepest descent method / back-propagation), to optimise the fuzzy rules.
5. End of procedure

Consider the data is composed of N data pairs with P input variables and one output variable $\{[x_{i1}, x_{i2}, \dots, x_{ip}], y_i\}$, $i = \{1, 2, \dots, N\}$, $j = \{1, 2, \dots, P\}$ M fuzzy rules are defined initially through the clustering procedure, the l^{th} rule has the form of:

For ZM type fuzzy rules:

R_l : If x_1 is about F_{l1} and x_2 is about F_{l2} ... x_p is about F_{lp} and then $n_l = G_l$

For TS type fuzzy rules:

R_l : If x_1 is about F_{l1} and x_2 is about F_{l2} ... x_p is about F_{lp} and then

$$n_l = \beta + x_1\beta_1 + x_2\beta_2 + \dots + x_p\beta_p$$

F_{lj} are the fuzzy sets defined by the following Gaussian type membership function (MF):

$$\text{Gaussian MF} = \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right] \quad (5.1)$$

G_l is the Gaussian MF for ZM's consequence defined as follows:

$$G_j = \exp\left[-\frac{(x-n)^2}{2\delta^2}\right] \quad (5.2)$$

The coefficients for the TS's linear functions are calculated as in the following equations:

$$\begin{aligned} Q &= (A^T A)^{-1} \\ b &= Q A^T y \end{aligned} \quad (5.3)$$

Where

$$A = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1P} \\ 1 & x_{21} & x_{22} & \dots & x_{2P} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_{N1} & x_{N1} & \dots & x_{NP} \end{pmatrix} \quad (5.4)$$

Using the modified centre average defuzzification procedure, the output value of the system can be calculated for an input vector $x_i = [x_1, x_2 \dots x_p]$ as follows:

For a TS fuzzy inference system

$$f(x_i) = \frac{\sum_{l=1}^M n_l \prod_{j=1}^p \alpha_{lj} \exp\left(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2}\right)}{\sum_{l=1}^M \prod_{j=1}^p \alpha_{lj} \exp\left(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2}\right)} \quad (5.5)$$

For a ZM fuzzy inference system

$$f(x_i) = \frac{\sum_{l=1}^M \frac{n_l}{\delta^2} \prod_{j=1}^p \alpha_{lj} \exp\left(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2}\right)}{\sum_{l=1}^M \frac{1}{\delta^2} \prod_{j=1}^p \alpha_{lj} \exp\left(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2}\right)} \quad (5.6)$$

n_l is the point that has the maximum membership value in the l^{th} output set.

To allow the integration of both ZM and TS fuzzy rules in the same NFI system, a fixed δ for each fuzzy membership function was used without updating it in the training process. This allows for the removal of δ from ZM's calculation and makes ZM's defuzzification function the same as for TS's one. From past experience, a ZM fuzzy inference system's performance is not strongly affected by using a singleton value instead of Gaussian membership function in every fuzzy rule.

Suppose the MUFIS is given a training input-output data pair $[x_i, t_i]$. The system minimizes the following objective function:

$$E = \frac{1}{2} [f(x_i) - t_i]^2 \quad (5.7)$$

The steepest descent algorithm is used to obtain the equations (5.8)-(5.14) for the optimisation of the parameters n_l , α_{lj} , m_{lj} , σ_{lj} and β_l

$$m_{lj}(k+1) = m_{lj}(k) - \frac{\eta_m \Phi(x_i) [f^{(k)}(x_i) - t_i]}{\sigma_{lj}^2(k)} \times [n_l(k) - f^{(k)}(x_i)] [x_{ij} - m_{lj}(k)] \quad (5.8)$$

$$n_l(k+1) = n_l(k) - \eta_n \varphi(x_i) [f^{(k)}(x_i) - t_i] \quad (5.9)$$

$$\alpha_{lj}(k+1) = \alpha_{lj}(k) - \frac{\eta_\alpha \varphi(x_i) [f^{(k)}(x_i) - t_i]}{\alpha_{lj}(k)} \times [n_l(k) - f^{(k)}(x_i)] \quad (5.10)$$

$$\sigma_{lj}(k+1) = \sigma_{lj}(k) - \frac{\eta_\sigma \varphi(x_i) [f^{(k)}(x_i) - t_i]}{\sigma_{lj}^2} \times [n_l(k) - f^{(k)}(x_i)] [x_{ij} - m_{lj}(k)]^2 \quad (5.11)$$

$$\varphi(x_i) = \frac{\prod_{j=1}^p \alpha_{lj} \exp(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2})}{\sum_{l=1}^M \prod_{j=1}^p \alpha_{lj} \exp(-\frac{(x_{ji} - m_{lj})^2}{2\sigma_{lj}^2})} \quad (5.12)$$

$$\beta_{l0}(k+1) = \beta_l(k) - \eta_\beta \varphi(x_i) [f^{(k)}(x_i) - t_i] \quad (5.13)$$

$$\beta_{lj}(k+1) = \beta_l(k) - \eta_\beta \varphi(x_i) [f^{(k)}(x_i) - t_i] x_{ij} \quad (5.14)$$

Where $\eta_m, \eta_n, \eta_\alpha, \eta_\sigma$ and η_β are learning rates for updating the parameters: $n_l, \alpha_{lj}, m_{lj}, \sigma_{lj}$ and β_l respectively.

In the MUFIS algorithm, the following indexes are used:

- training data points: $i=1,2,\dots,N$;
- input variables: $j=1,2,\dots,P$;
- fuzzy rules: $l=1,2,\dots,M$;
- training iterations: $k=1,2,\dots$;

The procedure for assigning the type of fuzzy rule to each cluster is described below:

1. Create a global linear function for all training data points
2. For each cluster, make a copy of the function from the step one and optimise it with WRLSE using the input vectors in the current cluster.
3. Apply the function to the data points in the current cluster to obtain the training prediction output and calculate the root mean square error for TS type fuzzy rules (TS-RMSE).
4. For each cluster, use the mean of the output values as the prediction output for each data point. Calculate the root mean square error for ZM type fuzzy rules (ZM-RMSE).
5. If the ZM-RMSE is larger than the TS-RMSE, then create a TS type fuzzy rule for this cluster; otherwise create a ZM type fuzzy rule.

Some clusters can have a very low number of input vectors or the data can contain high level of noise, but by optimising the global linear function using local input vectors instead of creating a new linear function using only the input vectors in the cluster, it allows more stability in prediction while still putting enough emphasis on the local cluster.

It was noted that some TS rules did not perform well in problem subspaces where variation of the output value was high. There is a notably positive correlation between the number of TS rules under-performing and the level of variation in a problem subspace. It can then be reasonably assumed that by replacing under-performing TS rules with ZM rules, we can achieve better overall accuracy.

The linear function for each cluster in the procedure above is updated using the following formula (Kasabov & Song, 2002) with λ being the forgetting factor.

$$\begin{cases} b(k+1) = b_k + w(k+1)P(k+1)a(k+1)[y(k+1) - a(k+1)^T b(k)] \\ P(k+1) = \frac{1}{\lambda} \left(P(k) - \frac{w(k+1)P(k)a(k+1)a(k+1)^T P(k)}{\lambda + a(k+1)^T P(k)a(k+1)} \right) \\ a \in A \end{cases} \quad (5.15)$$

w is defined as follows:(Kasabov & Song, 2002)

$$w = 1 - \left(\frac{\left(\sum_{i=1}^q (x_i - \bar{y}_i)^2 \right)^{1/2}}{q^{1/2}} \right) \quad (5.16)$$

The steepest descent algorithm (back-propagation) is not expected to make drastic changes to the parameters of the Gaussian MF and therefore the procedure described above provides a rough estimate of each fuzzy rule's performance by simulating the prediction process of each cluster for each type of fuzzy rule with its initial parameters.

5.2 Case Study and Analysis

MUFIS was applied to the Mackey-Glass dataset as described in chapter 4. For comparison purposes, the upper limit of the number of rules is set to 60 and training epochs is set at 200 for MUFIS.

Table 5.1

Prediction results of off-line learning models on Mackey-Glass $t+6$ training and testing data.

Methods	Neurons or Rules	Epochs	Training NDEI	Testing NDEI
MLP-BP	60	50	0.083	0.090
MLP-BP	60	500	0.021	0.022
ANFIS	81	50	0.032	0.033
ANFIS	81	200	0.028	0.029
DyNFIS	55	100	0.017	0.016
MUFIS	51	200	0.015	0.015

Note: Results extracted from (Hwang & Song, 2008; Kasabov & Song, 2002)

In the testing phase, m most activated rules are chosen to derive the final prediction. Since ZM fuzzy rule performs better than TS rules when the data contains high level of variation, ZM's contribution increases for these parts of the problem. This is evident in Figure 5.1, as more ZM rules outperform TS rules when the data contains higher variance.

More training epochs were used for MUFIS training because of the use of WRLSE to create its initial consequent function. This was designed to minimise the impact of potential clusters that were identified due to the noise in data (Noisy Cluster). The initial consequent function was the localised version of the global function which provides a balance between local function and global function.

When a fuzzy rule is created for a "Noisy cluster" and its consequent function created using input vectors only in this cluster, the function may be completely useless and have strong negative impact on the model as a whole.

By using WRLSE to create the consequent function through optimising the global function is using local data, it makes the initial rules slightly less accurate

while minimising the impact of the “Noisy Clusters”. It, therefore, requires more training iterations to train the model.

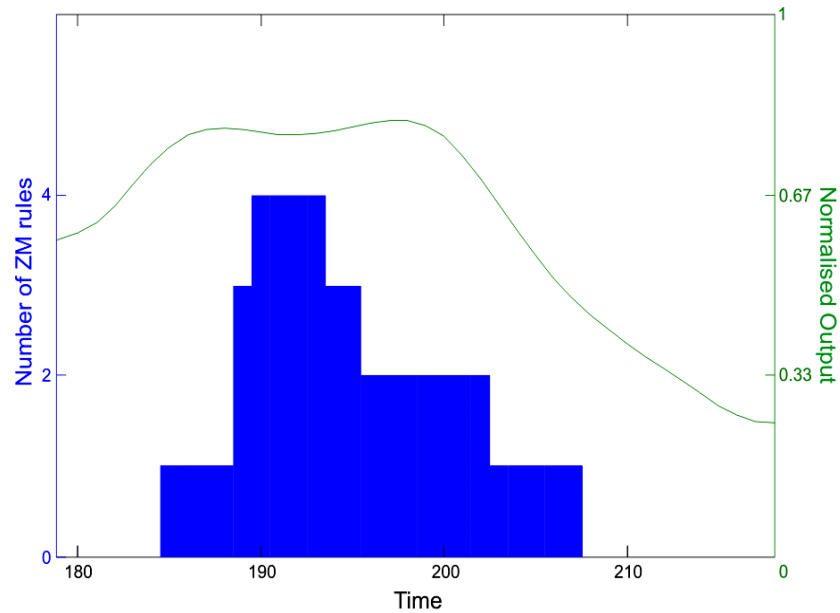


Figure 5.1 Number of ZM rules used in high variation regions of the problem space. Green line: test data’s normalised output values, blue bar: number of ZM fuzzy rules used. This indicates that ZM fuzzy rules are used more often when the data contains high variation.

An example of a MUFIS fuzzy rule set for a given test input vector p is shown below in normalised space:

Rule 1 (ZM) :

If x_{t-18} is about 1.53 and x_{t-12} is about 1.57 and x_{t-6} is about 1.59 and x_t is about 1.79 then
 $y = 1.8065$

Rule 2 (TS):

If x_{t-18} is about 1.42 and x_{t-12} is about 1.69 and x_{t-6} is about

1.68 and x_t is about 1.82 then

$$y = 1.5209 - 0.2982 \times x_{t-18} - 0.1015 \times x_{t-12} - 0.4538 \times x_{t-6} + 0.8854 \times x_t$$

Rule 3 (TS):

If x_{t-18} is about 1.56 and x_{t-12} is about 1.61 and x_{t-6} is about

1.78 and x_t is about 1.79 then

$$y = 1.1792 - 0.0011 \times x_{t-18} - 0.4304 \times x_{t-12} + 0.1312 \times x_{t-6} + 0.6249 \times x_t$$

Rule 4 (TS):

If x_{t-18} is about 1.60 and x_{t-12} is about 1.79 and x_{t-6} is about

1.70 and x_t is about 1.83 then

$$y = 2.3790 - 0.4333 \times x_{t-18} - 0.1029 \times x_{t-12} + 0.0747 \times x_{t-6} - 0.0659 \times x_t$$

In the MUFIS system for this case study with 51 fuzzy rules, 4 of them contribute to the prediction of the test input vector p. One of the fuzzy rules is a ZM fuzzy rule, which indicates that the region where the test input vector p is located may be noisy or contains higher variation and causes TS fuzzy rules to perform worse than ZM fuzzy rules.

For this particular input vector p in Mackey-Glass dataset, the ZM rule specifies that when the input is near this fuzzy rule, the predicted output from this rule will be 1.8065 and the same applies in TS rules with one major difference. The outputs from TS rules are derived on the fly based on the value of the input variables.

In an ideal situation where the dataset provide enough input vectors and covers the problem space adequately, ZM will not be used or constructed often as the consequent function used in TS are able to derive a meaningful function.

However, low quality clusters are often found on real life data modelling problems and it is common to see clusters that has very few input vectors or clusters with input vectors that has no correlation with the output.

In those cases, ZM rule should be used as it provides the most stable prediction for the input vectors in these kinds of clusters.

This predicted output from all activated rules will then be combined using equation(5.6), which does a weighted average of the outputs based on how close the input vector is to the centre of the membership functions of each rules.

5.3 Real world data modelling case study on renal function evaluation

The evaluation of renal function is the foundation for all renal research. The glomerular filtration rate (GFR) is considered the best index of renal function to date. The most accurate way to measure GFR is by the clearance of an administered tracer by the kidney. However, this is also very time consuming and expensive, making it not suitable for everyday clinical practice. Most clinicians opt to estimate the GFR using more easily accessible biological data instead (Levey et al., 1999; Song, Kasabov, Ma, & Marshall, 2006).

Many estimation methods have been proposed in the past to predict the GFR value using common laboratory variables (Bjornsson, Cocchetto, McGowan, Verghese, & Sedor, 1983; Cockcroft & Gault, 1976; Gates, 1985; Hull et al., 1981; Jelliffe, 1971; Jelliffe, 1973; Levey et al., 2007; Mawer, Lukas, Knowles, & Stirland, 1972; M. Walser, 1998; Mackenzie Walser, Drew, & Guldán, 1993). Most of the methods predict creatinine clearance as a measure of GFR instead of GFR directly and therefore can only achieve low accuracy because of the error/noise in the translation between creatinine clearance and GFR. The Cockcroft–Gault (Cockcroft & Gault, 1976) equation is the most widely used equation for predicting creatinine clearance.

The Modified Diet and Renal Disease (MDRD) (Levey et al., 1999) equation was introduced in 1999 and predicts GFR directly using gender, ethnicity, age, diabetes status, cause of renal disease, protein intake and mean arterial pressure as input variables. MDRD was able to achieve much higher accuracy than all previous equations, about half the error of the Cockcroft–Gault equation error in direct comparison.

The dataset used in this case study consisted of 441 GFR measures from 141 patients, i.e. more than one measure for each patient, collected from 12 sites in Australia and New Zealand. GFR was measured as the renal clearance of Cr-EDTA corrected for body surface area (Marshall et al., 2005; Song, Kasabov, Ma et al., 2006).

Comparison of the prediction accuracy of MUFIS and other methods is shown in Table 5.2, including the number of neurons and the test RMSE. The methods include a set of neural network models: MLP and RBF (S. Chen, Cowan, & Grant, 1991; Hornik et al., 1989; Lucks & Oki, 1999; Poggio, 1994), neuro-fuzzy inference models: ANFIS (Jang, 1993), DENFIS (Kasabov & Song, 2002)) and known formulas: Gates, Jelliffe⁷³, MDRD and Walser (Levey et al., 1999; Levey et al., 2007). All experimental results other than MUFIS were extracted from KBNN paper (Song, Kasabov, Ma et al., 2006) based on leave-one-out cross validation experiments. The result from KBNN model itself is excluded from the comparison due to its use of expert knowledge, meaning it cannot be considered as a direct comparison. The results from the other methods presented in the paper, however, can be as they have the same dataset and experimental design.

Table 5.2

RMSE comparison between various models on GFR dataset

Model	Neurons / Rules	RMSE
Gates (Gates, 1985)	-	7.48
Jelliffe73 (Jelliffe, 1973)	-	7.83
MDRD (Levey et al., 1999)	-	7.74
Walser (Mackenzie Walser et al., 1993)	-	7.38
MLP	12	8.44
ANFIS	36	7.49
DENFIS	27	7.29
RBF	32	7.22
MUFIS	21	7.17

Note: the results other than MUFIS were extracted from (Song, Kasabov, Ma et al., 2006) since the dataset and experimental design are the same.

MDRD is the current benchmark on renal function prediction using a linear regression model. It derives a linear function using all available data and then applies it to future data. There are two main issues with this model. First, the MDRD model was developed using data collected in America and when it was applied to New Zealand data, the error increases significantly due to the difference between the two populations. Second, the use of a linear model effectively ignores some potential unique problem subspaces.

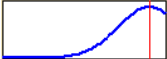

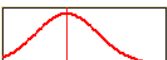


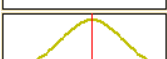
MUFIS addresses the second issue by identifying these unique problem subspaces through a clustering and supervised learning. It allows different models to be applied to them dependent upon their characteristics. The identification of unique problem subspaces is essentially a profiling procedure that allows researchers to identify groups of patients that may be very different from the majority for certain reasons and, therefore, should be studied closely to see if there is anything special about these groups of patients.

5.4 Knowledge Discovery

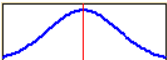
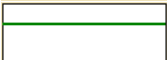



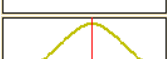
One of the main advantages of MUFIS is its ability to extract meaningful rules that bring new knowledge to the analyst. Being a fuzzy inference system, the model developed with MUFIS is a set of easy to understand IF-THEN rules. Many neural network models are a black box that is difficult, if not impossible, for users to interpret and extract useful knowledge from.

An example of the rules in its original space from the MUFIS GFR model is shown below:

Rule 1: ZM Rule

IF	Age	is about	74	
and	Sex	is	Male	
and	Serum creatinine	is about	0.57	
and	Serum urea	is about	43	
and	Race	is	White	
and	Serum Albumin	is about	35	
THEN	GFR = 16.4170			

Rule 2: TS Rule

IF	Age	is about	49	
and	Sex	is	Female	
and	Serum creatinine	is about	0.38	
and	Serum urea	is about	47	
and	Race	is	White	
and	Serum Albumin	is about	35	
THEN	$\text{GFR} = -0.30 * \text{Age} - 1.48 * \text{Sex} - 42.48 * \text{Serum creatinine} - 0.17 * \text{Serum urea} + 44.8 * \text{Race} + \text{Serum albumin} * 0.25$			

Take patient '▲' as an example. The patient is a 46 year old white female with Serum creatinine of 0.31, Serum urea of 13, and Serum albumin of 37. The patient is positioned in the problem space as shown below:

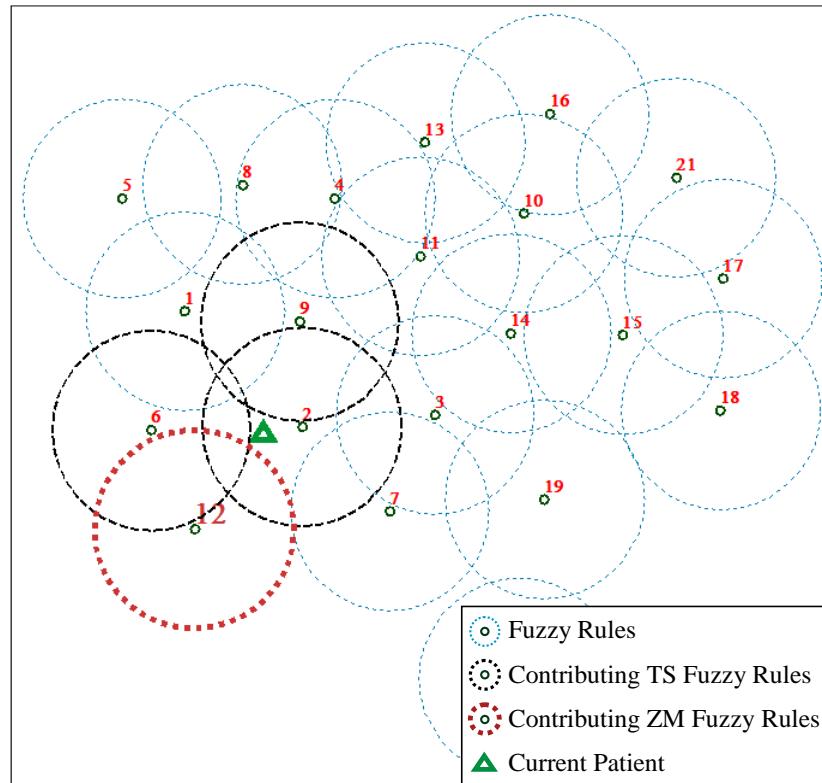


Figure 5.2 The location of fuzzy membership function centres and the patient in the Principle Component Analysis (PCA) space using the first two components. The prediction for the current patient is made using an integration of the three TS fuzzy rules and one ZM fuzzy rule.

As shown in Figure 5.2, MUFIS dynamically integrates multiple types of fuzzy rules based on the position of the current patient in the problem space. This shows that ZM fuzzy rules work better than TS fuzzy rules in some problem subspaces. TS fuzzy rules are suitable for some patients while a mixture of TS and ZM fuzzy rules are suitable for other patients.

Prediction accuracy is not the only measure of model quality in a real world data modelling problem. Sometimes more than one model is able to achieve the best or similar accuracy when applied to the same problem. In this case, the model with easy to understand rules should be given preference, as it allows for further analysis of the problem through analysing the model. MUFIS, due to its easy to read fuzzy rules, has advantages in this respect.

Each of the rules represents a new knowledge, which highlights a problem subspace that is unique and may be worth more in-depth investigation. The antecedent of the rule identifies a point in the entire problem space and the consequence shows the type of rule used. If a ZM rule is used, it indicates this region is more noisy and non-linear. This can be caused by many issues, such as a lack of input vectors, excessive noise, or simply means that linear function is not a suitable model for this problem subspace. The regions can then be inspected further using visualisation tools in 2D or 3D, or other suitable analysis tools to find out if there are abnormalities in this region.

Note that these rules are not the cluster centres as in the first part of the algorithm, as they have been optimised through back-propagation on the training data with global RMSE as its objective function.

It is possible for new input vectors to fall outside the problem space covered by the existing model. This occurs when the original training dataset was incomplete and could not explain the problem fully. When this happens the model will still try to make the prediction using the nearest activated fuzzy rules, though the prediction made for this input vector will be less accurate than the predictions made for input vectors that fall in the areas covered by the original training data.

5.5 Conclusion

A novel fuzzy inference system, MUFIS, was proposed in this chapter, which allows both ZM and TS fuzzy rules to be integrated into a fuzzy inference system. It demonstrated that using a mixture of multiple types of fuzzy rules in a fuzzy inference system can lead to better prediction accuracy in some problems. In addition, the fuzzy rules extracted from the MUFIS model are also more meaningful than the ones from DyNFIS and DENFIS as it now shows the type of fuzzy rules that each problem subspace is better suited to.

The proposed method addresses the following real world data modelling issue:

- Unique problem subspaces

MUFIS creates fuzzy rules based on clustering, suitability of fuzzy rule type and supervised learning on the training data to ensure that unique problem spaces are well represented. Each problem subspace may be assigned with different fuzzy rule types depending on its characteristics.

5.6 Discussion

The concept of integrating different types of fuzzy rules, or local models, in a single fuzzy inference system has shown positive results in benchmark and real world data modelling problems. To allow a simpler and more flexible approach to integrate different types of models, a multi-model system is proposed in the next chapter that aims to allow different types of models to be integrated at a higher level, without changing or modifying the underlying model.

CHAPTER 6 INTEGRATED TEMPORAL AND SPATIAL MULTI-MODEL SYSTEMS

Temporal model refers to a model that concerns only with the change of values over time. This could be a simple linear regression model that creates a trend using recent data only. Since the model concerns only with the change of value over time, I refers to this kind of model as temporal model.

Spatial model look at the problem based on the similarity between recent pattern and historical patterns. It is usually based on Euclidean distance between patterns, hence the name spatial model.

A glance at previous studies in neural networks shows that there is no one model that works better than others in all real world data modelling problems. This is most likely due to the uniqueness of the problems and the way the models are applied. Simple regression models can outperform neural network models and vice versa depending on the problem.

Consider a time-series prediction scenario, where the data is continuously collected. New patterns occasionally emerge, as often seen in problems related to nature, where data is seasonal and repeats existing patterns in general. However, new patterns appear occasionally due to changes in the environment, e.g. global warming and earth quake. It is not possible to know whether the current model's prediction will be accurate on new pattern as it is extrapolating, or is it able to adapt to the new pattern. It is then logical to consider the ideal of integrating multiple types of models that addresses the problem from very different angle to increase the possibility that one of these models will work on the new pattern. This way, the system can switch to, or adjust the contribution level of, the most accurate model on the new pattern.

In this chapter, a multi-model system framework (MMS) that incorporates both spatial and temporal models is proposed to allow two contrasting views on

the same problem. The idea is to have a temporal model that make the prediction based on only recent data and a spatial model that makes the prediction based on historical data only.

The temporal model sees the problem as change of values over time. This kind of model is most suitable for new patterns as the old patterns in historical data is forgotten quickly and replaced with new patterns.

The spatial model, on contrast, look at the similarity between the recent pattern and patterns that have occurred in the past. The prediction is made based on historical patterns that are identified through the distance between the new pattern and historical patterns in the problem space. It most suited for recurring patterns as the model can learn from many historical examples to make the most accurate prediction.

In this chapter, a multi-model framework that utilises both a temporal model and a spatial model is proposed (Song, Kasabov, Hwang, & Chrystall, 2006).

Since patterns shift gradually, the prediction error made by each model from earlier predictions can be used as a parameter to adjust the level of contribution from each model.

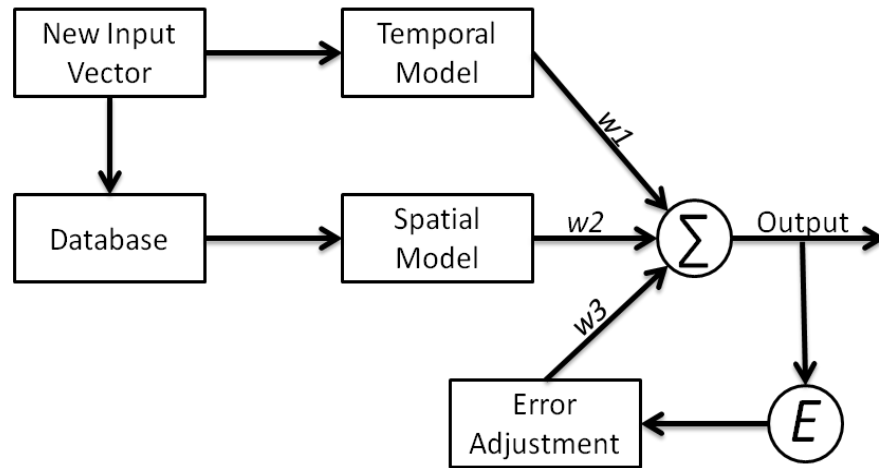


Figure 6.1 Multi-Model System framework, provides multiple views of the problem and integrates the output from each model based on its previous prediction error. Modified from Kasabov's book on evolving connectionist system in 2007 (Kasabov, 2007a).

The advantage of the above system design is that, when the data has a repeated pattern, the spatial model will be able to follow it very well by learning from historical data and the temporal model will perform moderately well, as it is based on recent data only. When a new pattern appears, the spatial model's prediction accuracy drops as there is no previous pattern for it to learn from and a correct prediction and the temporal model will perform better than the temporal model. The contribution from the temporal model will increase until there are enough instances of the new pattern in the historical data for the spatial model to learn from. The design ensures that the system will not be chaotic when new pattern appears while utilising the high accuracy of the spatial model.

6.1 Algorithm

The modified MMS framework consists of three main parts. The first part is a linear regression model: a weighted least square regression estimator, which is a model that aims to solve linear problems over the entire problem space. The second part is a personalised model which aims to solve the problem by matching recent pattern with historical patterns and derive the output from only a few selected patterns. The third part is the contribution weight adjustment module

that adjusts the contribution of each model based on their recent prediction accuracy.

6.1.1 Spatial Model – Transductive Weighted Neuro Fuzzy Inference System (TWNFI)

Any distance based models may be used here. However, TWNFI is chosen as the most suitable model. TWNFI (Song & Kasabov, 2006) is a personalised fuzzy inference system with feature weight optimisation. The author considered it as the most sophisticated personalised model as it has been published in major journal and has shown better prediction accuracy than other methods. It also has the following characteristics, making it the most model to utilise historical data in real world data modelling problems.

1. Personalised model, only relevant data is used
2. Fuzzy inference system with input variable weight optimisation. The method optimises both the fuzzy system and the variable weighting.

For each test input vector, it first selects a subset of input vectors and creates a fuzzy inference system similar to DENFIS. The fuzzy inference system and the input variable weights are optimised together. Due to the change in input variables weights, the subset of input vectors used to construct the model also changes in each iteration. This process is repeated many times until there are no more changes in the subset of data for training as shown in Figure 6.2.

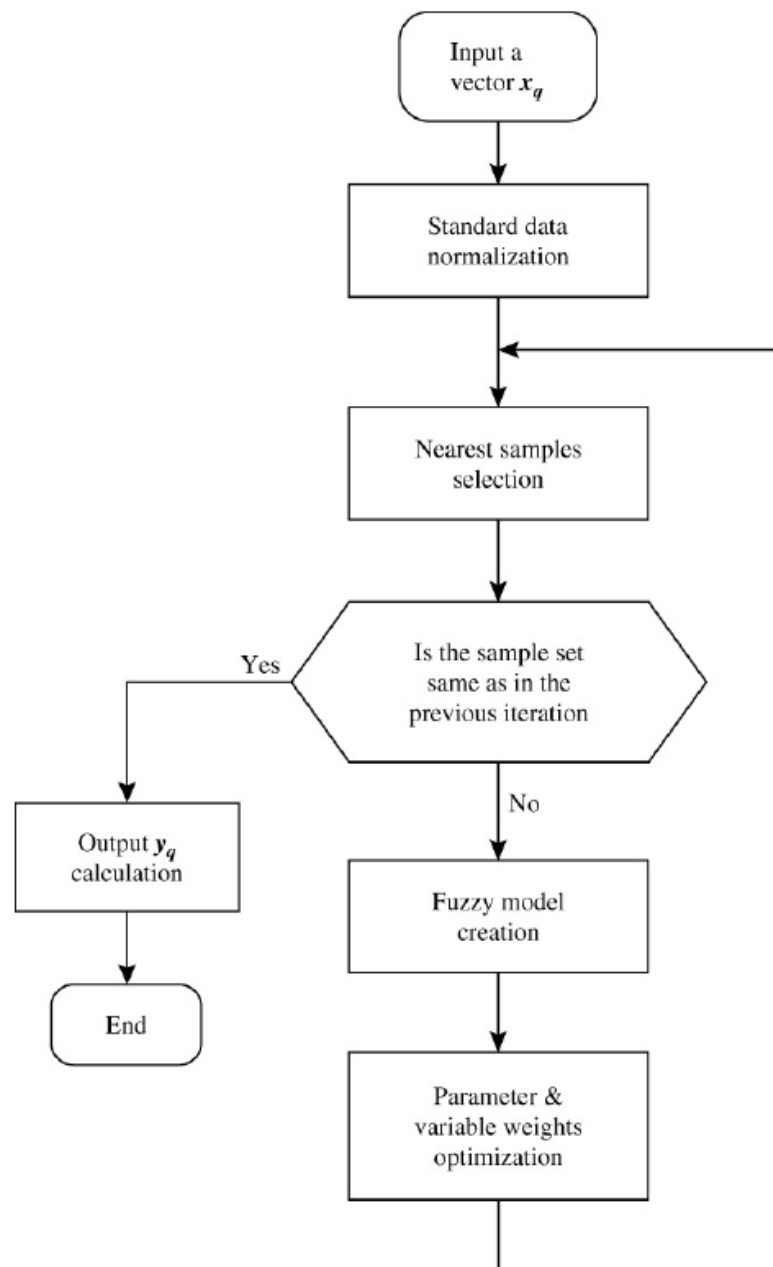


Figure 6.2 A block diagram of the TWNFI algorithm.

This model uses the current pattern, i.e., the past n time points, as the selection criteria for selecting a subset of data from the entire dataset. This searching process is based on the Euclidean distance between the recent pattern and historical patterns. Due to the complexity of the algorithm, the number of input variables must be limited.

6.1.2 Temporal Model – Weighted Least Square Estimator (WLSE) or Weighted Recursive Least Square Estimator (WRLSE)

Any model that is capable of creating a trend from recent data may be suitable temporal model. The idea is to ignore all historical data and concentrate on the recent trend. This allows the model to be unaffected by the change of pattern as it adapts to the new pattern almost immediately. Both WLSE and WRLSE as described in 3.4.1.2 and 3.4.1.3 are considered good candidate for temporal model.

Depends on the characteristic of the data, the amount of data used to derive the trend should be adjusted to represent the trend. For example, if the problem consists of data collected over the period of 10 years and the pattern does not change often, the definition of short term trend may be defined using the data from the past 6 months. On contrast, if the problem is to look the price of a particular stock and its pattern changes almost once per week. Then the short term trend may be defined using data from the past 3-4 days.

6.1.3 Error Adjustment

Since both models make a prediction at each time point, it is necessary to combine the two models to provide a singular output. This is usually done by changing the contribution level or weights between the models.

In this thesis, the assumption that pattern changes from one to another occasionally and gradually and therefore the previous prediction error is considered a good indicator on how well each model is and will be performing in the near future.

The method for aggregating the two models is shown below:

$$V = w_1 * V_{\text{spatial}}(t_0 + T) + w_2 * V_{\text{temporal}}(t_0 + T) - w_3 * E(t_0) \quad (6.1)$$

Where: V is the predicted output of the system

t_0 is the current time

T is the number of time units ahead to be predicted

V_{spatial} is the output of TWNFI model

V_{temporal} is the output of WRLSE model

E is the prediction error at t_0 , $E(t_0) = V(t_0) - D(t_0)$; D is the actual output and V is the predicted output on t_0

w_1 , w_2 and w_3 are weights for V_{spatial} , V_{temporal} and E respectively

w_1 and w_2 are calculated based on the normalised error contribution of each model, the higher the error, the lower the weight.

$$w_1 = 1 - \frac{E(t_0)_{\text{spatial}}}{E(t_0)_{\text{spatial}} + E(t_0)_{\text{temporal}}} \quad (6.2)$$

$$w_2 = 1 - \frac{E(t_0)_{\text{temporal}}}{E(t_0)_{\text{spatial}} + E(t_0)_{\text{temporal}}} \quad (6.3)$$

w_3 is a fixed weight, which allows gradual adjustment of the error if the model is constantly over or under predicting.

The integration method used in this multi-model system is very simple. It is possible to replace it with a more sophisticated learning model to allow the weights between the two models to be adjusted through machine learning.

6.2 Example

Take the following time-series data as an example, the patterns moderately consistent up to the 60th time point and a new pattern emerges after that.

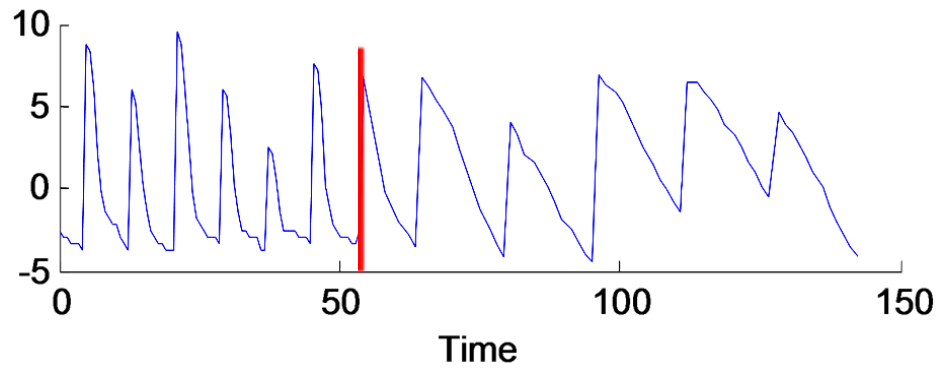


Figure 6.3 An example scenario of time-series pattern that changes over time. Red bar indicates the 53rd time point

Both spatial and temporal model would perform reasonably well prior to the 53rd time point. At the 53rd time point, the new pattern starts, the spatial model's prediction accuracy declines as it will not be able to find similar input vectors in the historical data. The error adjustment module gradually increase the contribution of the temporal model, as it uses only recent data and, is therefore, less affected by the change of pattern. After a few appearances of the new pattern, the spatial model recovers from low prediction accuracy as more appearances of the new pattern become available in the historical data. The error adjustment module gradually increases the contribution level of the spatial model as its prediction accuracy increases.

6.3 Conclusion and Discussion

The proposed system uses both temporal and spatial models to create a contrasting view of the problem, one concentrate on the recent trend and the other concentrate on affects of similar trends in the past and adjust the contribution weight between the two models using their recent prediction error as an indication.

The actual models used for temporal and spatial model are not fixed. Models like DENFIS or eTS are suitable replacement for TWNFI model as they may be less computationally expensive. The temporal model may also be

replaced with simple Least-square estimator or Kalman-filter. It is entirely based on the characteristic of the problem and its requirements.

The currently employed integration method is based on each model's prior prediction error as the assumption is that the pattern will only change gradually. Other integration methods or different weight adjustment criteria may be used. For example, one may use the average error from each model for the last n predictions as criteria to adjust the current prediction to allow a more smooth adjustment of contribution weights.

The model aims to address times-series problems where patterns change over time but does not switch rapidly between patterns.

This algorithm addresses or minimises the following issues.

- Evolving problems (partial)

Each prediction is made with models dynamically created and therefore the proposed method will always take use of the latest dataset.

- Unique problem subspaces

The use of the spatial model (TWNFI) emphasises on the unique problem subspaces of the test input vector.

In the next chapter, this algorithm is applied to a real world time-series case study predicting milk production volume.

CHAPTER 7 THE APPLICATION OF THE MULTI-MODEL SYSTEM TO SOLVE A REAL WORLD TIME-SERIES DATA MODELLING PROBLEM

The goal of the case study was to develop a model that can provide a 4-day-ahead ($t+4$) prediction of milk production volume based on the given data for resource management purposes.

7.1 Data Description

The farm milk collection data consists of the details of milk pickups from 575 farms selected from various regions of New Zealand. Information for each of the selected farms' location and vat capacity was also provided. The data was provided in its original form without any pre-processing.

The following describes some basic understanding of the data:

1. There are two milkings per day. One in the morning and one in the evening.
2. Production volume between day and night milking is slightly different. The average ratio between night milking and day milking is 45/55. The ratio varies slightly in different regions of the country (see next section).

Table 7.1

Description of variables used to describe a milk pickup from a farm.

Variable Name	Description
ID	The unique number for each pickup.
Product	The type of milk that has been picked up, either Ccolostrums or normal milk.
Farm	The unique number used to identify each farm.
PickupDate	The date the pickup occurred.
PickupShift	The type of the shift of the current pickup, either Day or Night
PriorDate	The date of previous pickup.
PriorShift	The type of the shift of the previous pickup, which can be either Day or Night
Actual	The actual pickup volume, disregard exception and adjustments.
EstPerDay	The predicted daily production volume made by Aspire, the company that does the current estimate.
Adjustment	The adjustment made, due to an exception.
Reason	The type of exception that has occurred.

Table 7.2

Description of variables used to describe a farm.

Variable Name	Description
FarmId	The unique number used to identify each farm.
Scheduling Area	Region to which the farm belongs
Zone Code	Zone to which the farm belongs
Active From	When the farm becomes active
Active To	When the farm becomes inactive.
Address	Physical address of the farm
Shareholding	Whether the farm is a company farm or a shareholding farm
VatCap	The capacity of the vat.
Product	The type of milk the farm produces
OffRoad	Off-road distance
oneWay	one-way road or two-way road
Altitude	Altitude
Note	Special Notes/instructions by the farm owner
GPSX	GPS data, X-axis
GPSY	GPS data, Y-axis
Night Shift Ratio	Ratio of farm's milk volume between day and night

7.2 Data Preparation Process

The data provided by Fonterra is in its raw form without any processing and therefore a lot of unwanted data is embedded in the provided data. It was, therefore, necessary to process it before it was used to train the model to ensure only relevant data is used for training.

7.2.1 Colostrum Milk Data Removal

Some farms produce colostrum milk at the beginning of each season. This data should not be mixed with normal seasonal data as they have very different patterns and the goal of this case study does not include the prediction of colostrum milk production. As the milk type is categorised for each pickup, these input vectors are therefore accurately removed.

7.2.2 Season Identification

The data does not contain any accurate information on when the milking seasons begin and end. However, most farms operate based on seasons and it is, therefore, necessary to separate the data into multiple seasons.

In this case study, the seasons are identified using the gap between pickups. A season is considered finished when there have been no pickups for over thirty days.

7.2.3 Winter Milking Farm Season Identification

Some farms are specially contracted to provide milk over winter while the majority of the farms cease operation during this period. The identification of the beginning and the end of a season for winter milking farms is therefore a difficult task.

A farm is identified as a winter milking farms if the season is longer than a year. The data from this farm then needs to be split into multiple seasons using the official season start date, which usually are not be the actual farm season start date. Because of this, the seasonal data from these farms usually have a flat period in the between the official season start date and the real season start date.

7.2.4 Remove Exceptionally Short Seasons

Since the seasons are identified based on the gaps between pickups, some seasons are exceptionally short because of the way the farm works or due to exceptional circumstances. For example, a farm can start milking for a few days and then decide to delay the milking until later if the milk production volume does not increase at the expected rate because the cows are simply not ready yet.

All seasons that are shorter than 45 days are removed, as a normal season is almost always longer than 90 days.

7.2.5 Calculate Day/Night Milking Volume.

Depending on the farms' region, the production volume for each milking is calculated using the correct day/night shift ratio.

Consider the case where the previous pickup was a night shift and the current pickup is done three milking shifts after the previous pickup. i.e. day milking shift (0.55), night milking shift (0.45) and another day milking shift (0.55). If the pickup volume is 1,000 litres, the first day milking volume will be $1,000 \times 0.55 / 1.55 = 354.84$ litres, followed by a night milking: $1,000 \times 0.45 / 1.55 = 290.32$ litres, followed by another day milking: $1,000 \times 0.55 / 1.55 = 354.84$ litres, which totals to 1,000 litres.

7.2.6 Weight Adjustment Using Exception Data

Exception data is used to provide importance or weight to each data point. The input vector that is labelled with exception data are usually outliers. Depending on the cause, the associated pickup volume's importance is reduced. The weight for a normal pickup data point is 1; it is reduced to 0.8 if it is affected by an exception.

Not all exceptions affect the pickup volume. The following types of exceptions are considered important, affecting one or more pickups. The list shows the exceptions that have impact on actual pickup volume and the pickups that are affected.

Let t be the time of the current pickup and $t+1$ be the following pickup.

Table 7.3

The number of pickups affected by different type of exceptions

Type of Exception	Number of Pickups affected
Added Water	t
Already Started Milking	t, t+1
Dumped Milk	t
End of season variance	none
Herd Change	t
Incorrect Estimate	none
flood loss	t
Multi-shed farm	t
Non-standard milking cycle	none
Other (add notes)	none
Out of Schedule collection	t, t+1
Pick up remainder in next shift	t, t+1
Production change	t
Pumping problem	t
Scheduled for another tanker	none
Start of season variance	none
Still Milking	t, t+1
Volume Adjustment (one-off):	t

It was understood that not all exceptions were recorded in the dataset. For example, there were outliers detected with patterns similar to the effect of a still milking exception, where one pickup volume was very low and the following pickup was very high, with the average of the two appearing to be normal. However, these outliers were not marked with any exception.

7.2.7 Outliers Detection and Treatment

Outliers are unexplained exceptions from the normal pickup routine which if undetected could grossly distort predictions of volume production. These outliers are identified and treated using the following method:

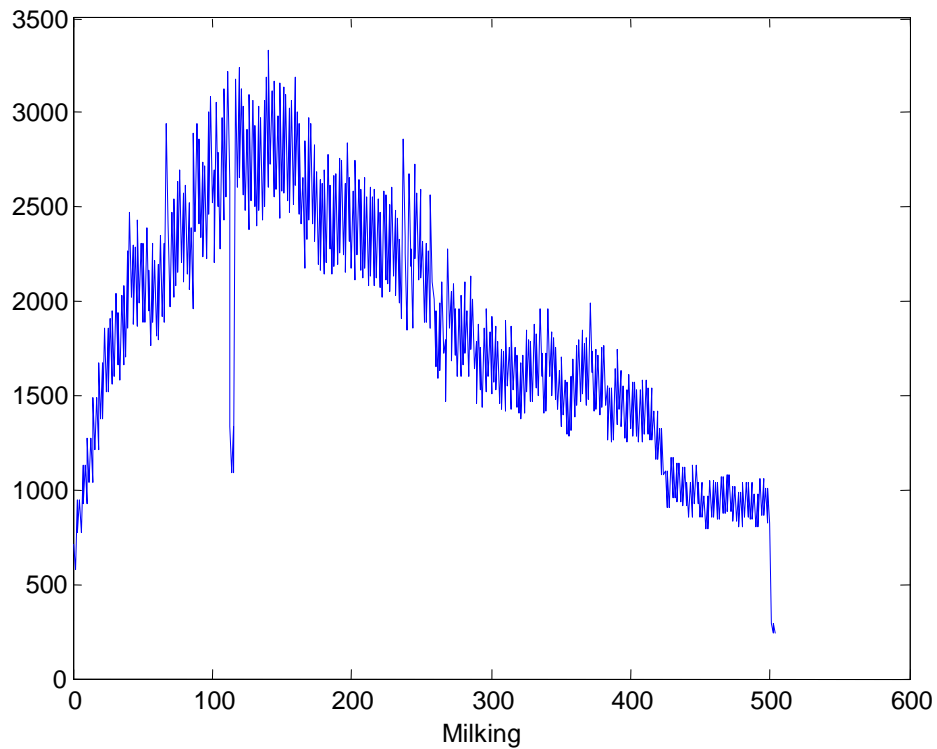


Figure 7.1 Shows an original set of data which was then processed to identify outliers using the process described below.

Proposed and implemented method for outlier detection:

- Step 1. Calculate the five-day moving average of the seasonal milking volume (The result for the time series from Figure 7.1 is shown in Figure 7.2).
- Step 2. Calculate the difference between each shift's volume and its five-day moving average volume (The difference between Figure 7.1 and Figure 7.2 is shown in Figure 7.3)

- Step 3. Identify outliers if a shift's volume is outside the range of four times the mean of the differences (The identified outlier is circled in solid red in Figure 7.3).
- Step 4. Replace the outlier's value with the average value of the pickup volumes of the pickup before and after the outlier. However, if there are two outliers, one followed by another, their values are calculated as the average of the two pickups.
- Step 5. Re-calculate the production volume for each shift that contributes to this outlier using the method in step 1-4.

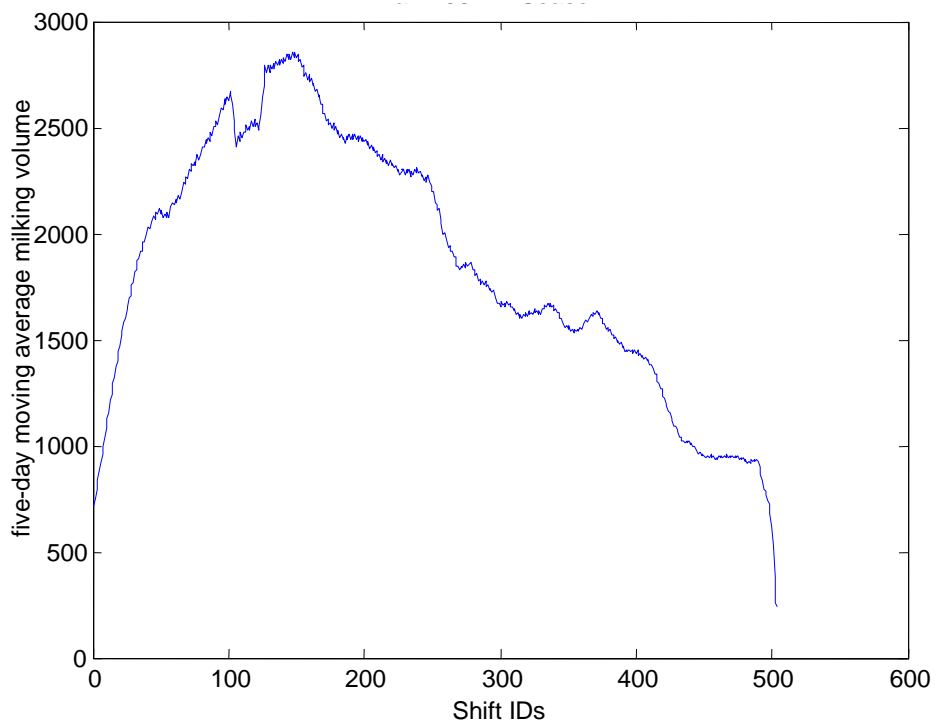


Figure 7.2 Five-day moving average for the season from Figure 7.1.

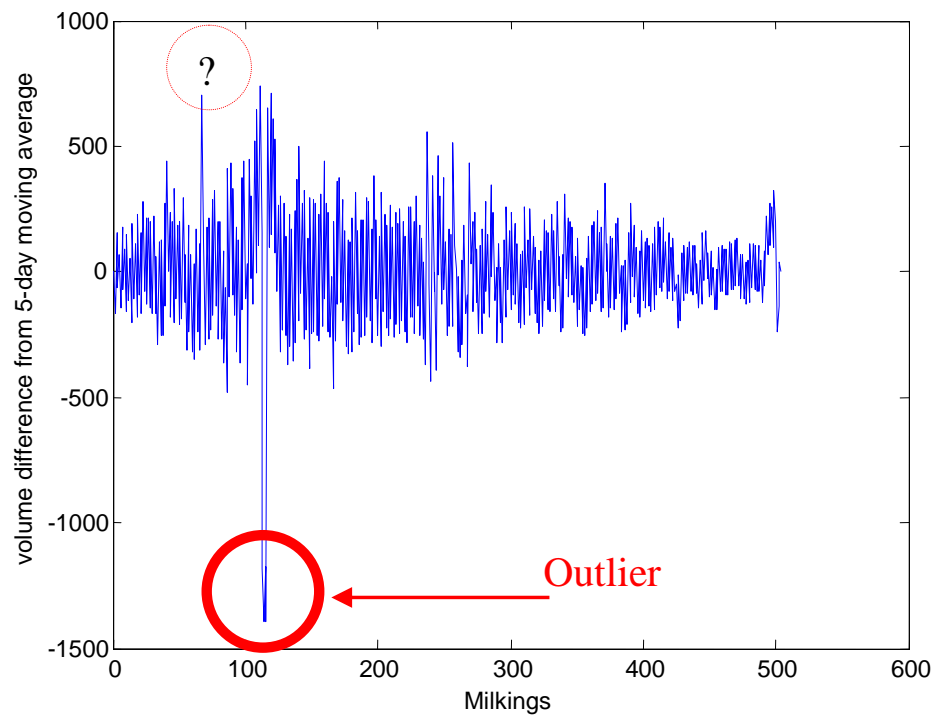


Figure 7.3 The difference between the original milking volume and the five-day moving average volumes (from Figure 7.1 and Figure 7.2)

The weighting for the treated outlier is set to 0.5 compared to 1.0 for the normal data. This is to minimise the impact of artificially generated data as it can be wrong. Because there is enough data available, this approach is not expected to cause any significant problem.

7.2.8 Data Feed to the Model

The pre-processed data is used to train and test the model. The processed data contains data from 575 farms and each farm contains between one and four seasons of data. For each season, the following information is provided in the pre-processed data: Farm ID, season number, unique time ID for each milking, volume of each milking, pickup id, original pickup volume and weight for each input vector.

7.3 Weekly Pattern Analysis

It was expected that there may be weekly patterns resulting from any regular perturbations to the normal weekly schedule. For example, differences in milking times to accommodate weekend sport, different milk staff, or even the music in the sheds. The method we employed was autocorrelation, which measures how well a milk data item matches a time-shifted version of the same data.

This technique is useful only when there is no long term trend in the data and unfortunately this was the case in our dataset where the volume increases at the beginning, and then gradually decreases.

Figure 7.4 shows the daily milking volume of a farm. It was necessary to remove the trend from the data (de-trend) in order to apply the autocorrelation technique on this season to look at the daily variations,

The trend was calculated using a five-day moving average, which appeared as a smoothed line.

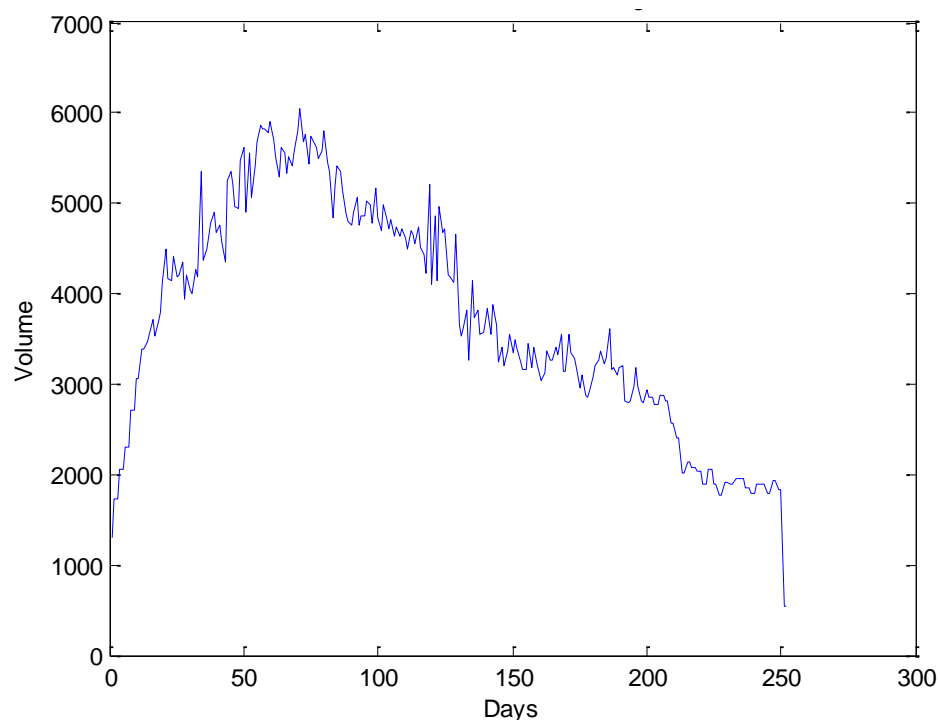


Figure 7.4 Daily milking volume in a season

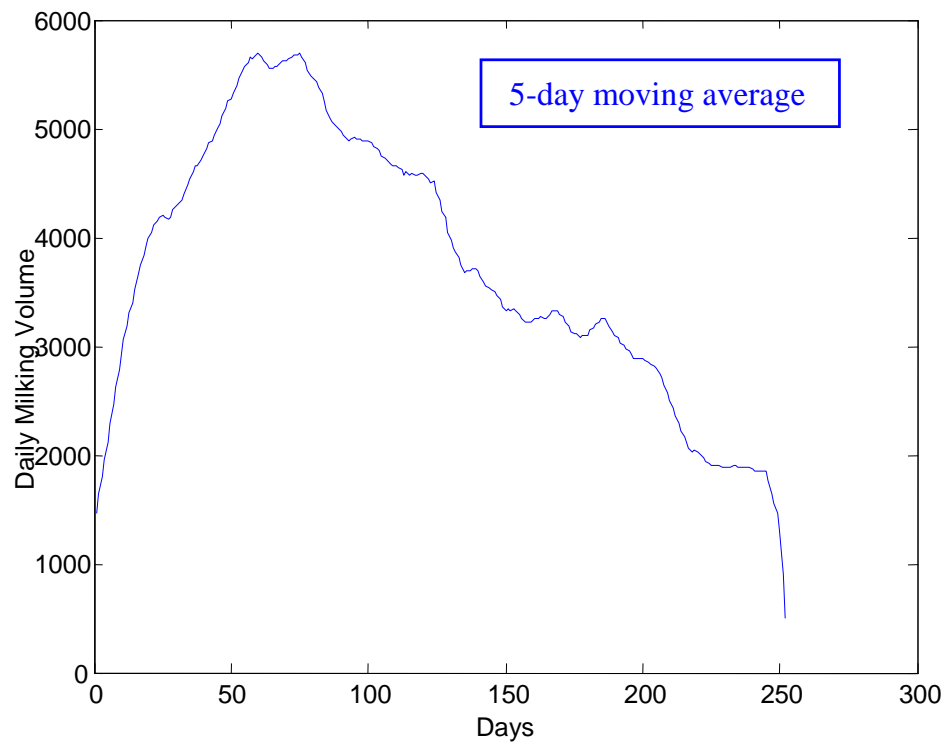


Figure 7.5 Five-day moving average of the daily milk volume in a season

The difference between the original volume and the moving average volume can then be calculated, which is then analysed with autocorrelation as shown in Figure 7.6.

The correlation value at different time lags (up to nine days) is shown in Figure 7.7

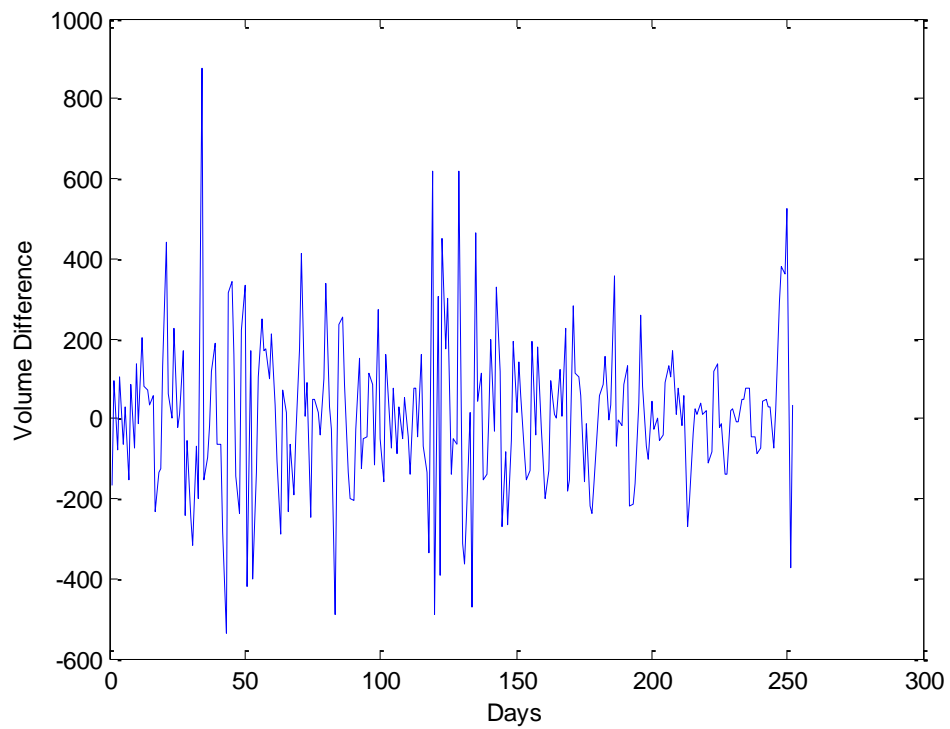


Figure 7.6 Differences between the daily milking volume and the five-day moving average volume

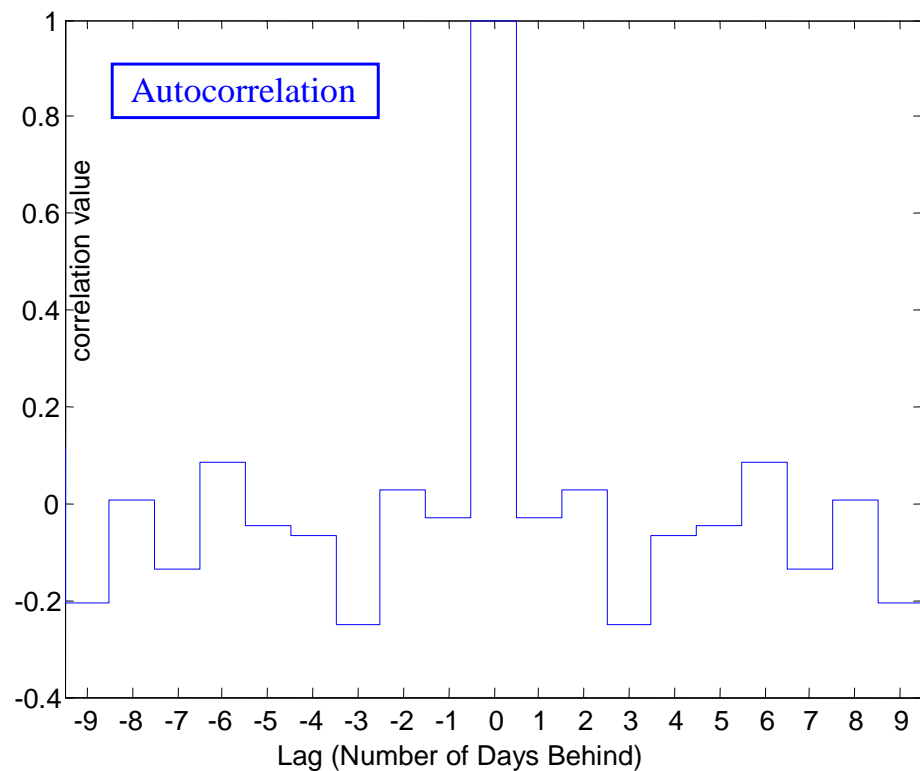


Figure 7.7 Autocorrelation values show the relationship between the “today” milk volume (indicated on the x-axis as 0) and previous days.

Figure 7.7 shows a standard season without very significant weekly patterns, as none of the correlation values for the shifted data are significant. A little negative correlation is manifested with 3 days back for this particular farm.

The autocorrelation of all farms and all seasons is shown in Figure 7.8 which shows the overall trend for the farms. The blue lines show the actual correlation value, and the red line shows the mean and standard deviation at each time lag.

Note that correlation value varies slightly depending on the de-trending method. When using the five-day moving average method, none of the farms have showed a significant weekly pattern. However, it does show that the production volume from the day before positively correlates to the current date's production volume and that in many cases there appear to be moderate correlations with the production data four days either side of the day. There may be a half-a week pattern in milk production, though it is not very significant.

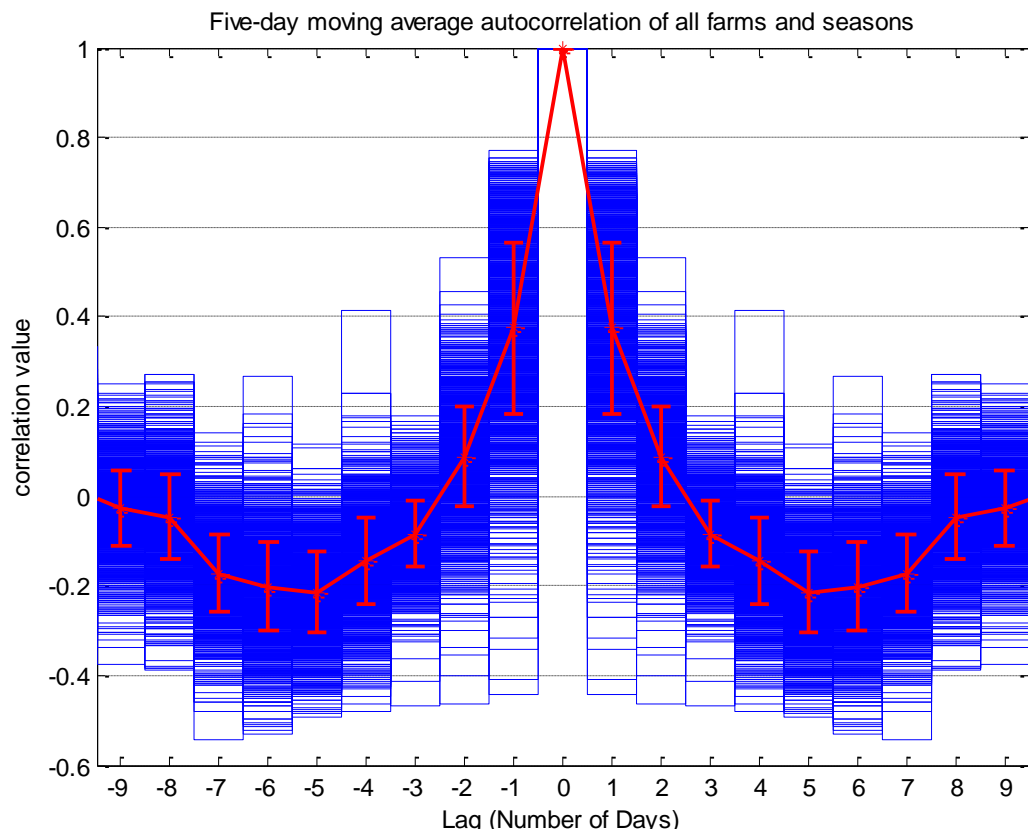


Figure 7.8 Aggregated autocorrelation result of all farms and seasons. No significant average correlation is found, the highest being negative 0.3 at 5 days' back.

7.4 Farm Zone Analysis

The following Figures show the normalised daily milking volume of all farms in one zone in the third season. It is important to note that the milk volume is normalised and farms with less than four seasons of data are ignored in this analysis.

The red I shape bar shows the mean and standard deviation of a specific date. The red '*' between the bar is the mean value. The blue dots show the maximum and minimum values of the specified time lag.

The analysis shows similarity in milk volume trends between farms in a zone, in some cases – across several zones. For example, Zones 1 and 2 as shown in Figure 7.9 and Figure 7.10 show some correlation, while Zones 3 and 4

as shown in Figure 7.11 and Figure 7.12 show less similarity. We can say that farms in a zone do behave similarly in a season.

We may also say that the trends of the zones are similar, since they always increase in the beginning and then gradually drop until the end of season, but some significant difference between some zones can be seen.

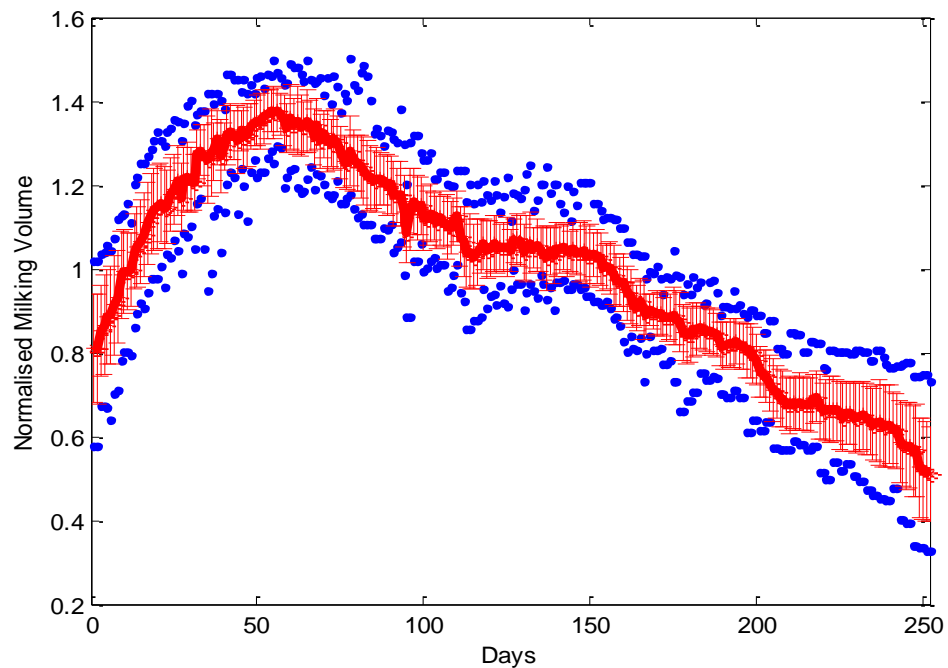


Figure 7.9 Normalised milking volume for Zone 1 (Mean and standard deviation – red; Min-Max volumes – in blue)

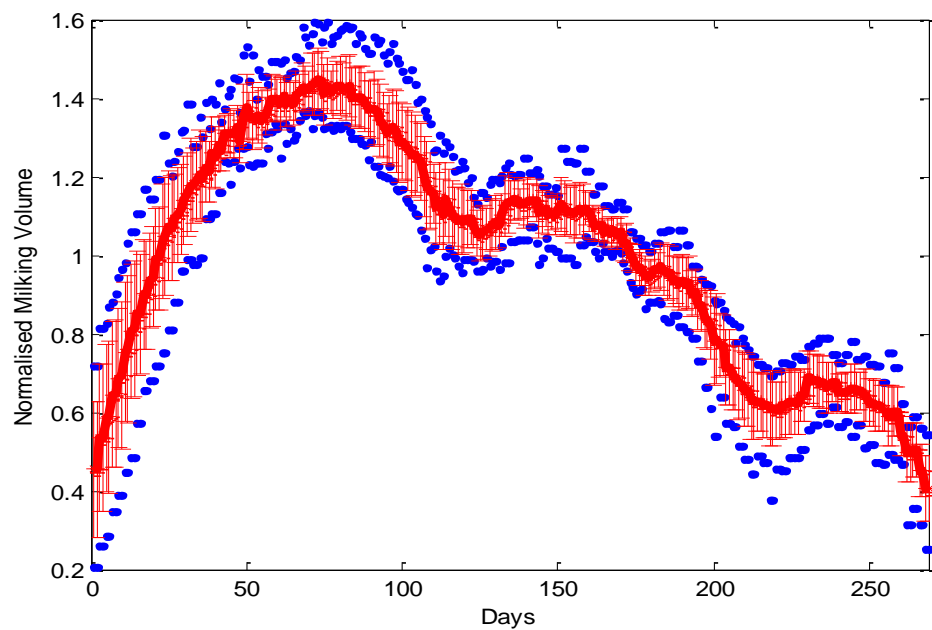


Figure 7.10 Normalised milking volume for Zone 2

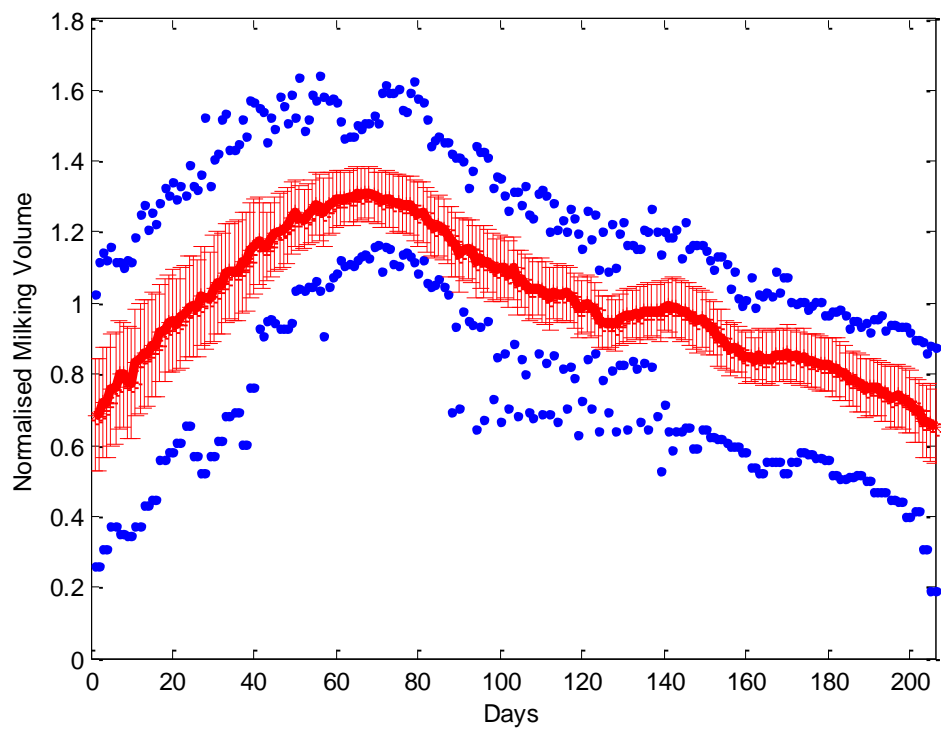


Figure 7.11 Normalised milking volume for Zone 3

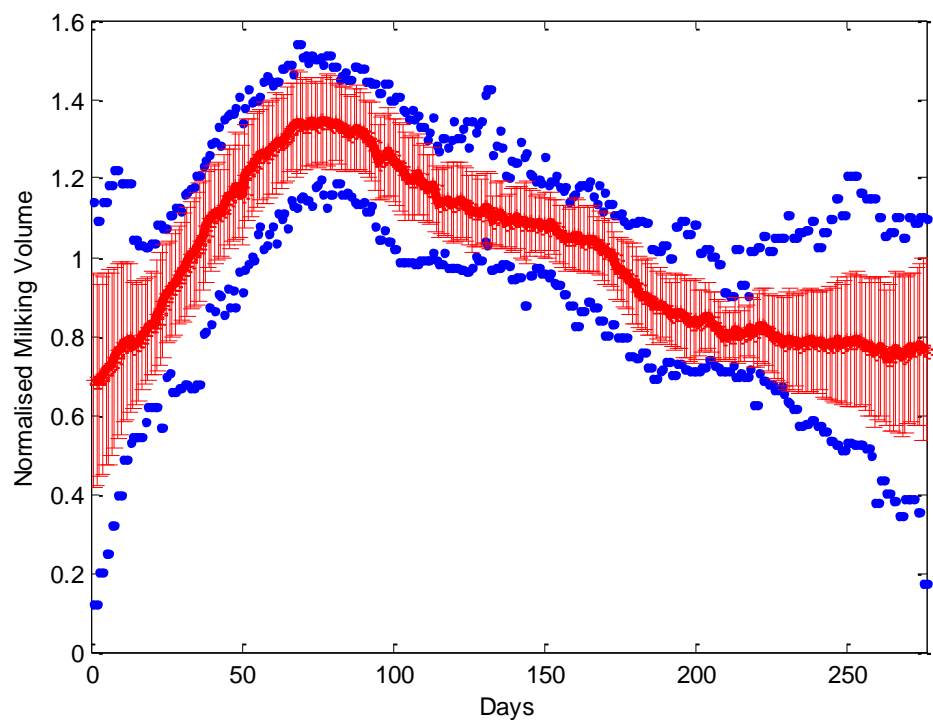


Figure 7.12 Normalised milking volume for Zone 4

There was some similarity between farms in the same zones but the level of similarity is not strong enough to show through normal day to day variations.

One of the main issues was that the farms start their season on different dates and the initial pattern is very difficult to predict and often omitted from the seasonal data. For example, some farms may start milking for a few days then decided to stop due to the condition of the cows or other management issues. The analysis was done for the period where all farms in the zone have correct data and therefore some zones are analysed without the beginning or the end of the period and make them shorter than other zones.

Since variation was high and part of the data was omitted in this analysis, the zone data was not used to train the prediction model.

7.5 Data Smoothing

The volume of milk shows considerable day to day variation (Figure 7.13). Pre-processing is required to remove daily variations and therefore a data smoothing algorithm is applied.

The smoothing method called 'Second-order Linear Smoothing Method' was used, which is commonly used for smoothing time series data. This is illustrated in Figure 7.14.

This smoothing method produces a smoother line, which may be better for the training than the earlier method used in the data processing.

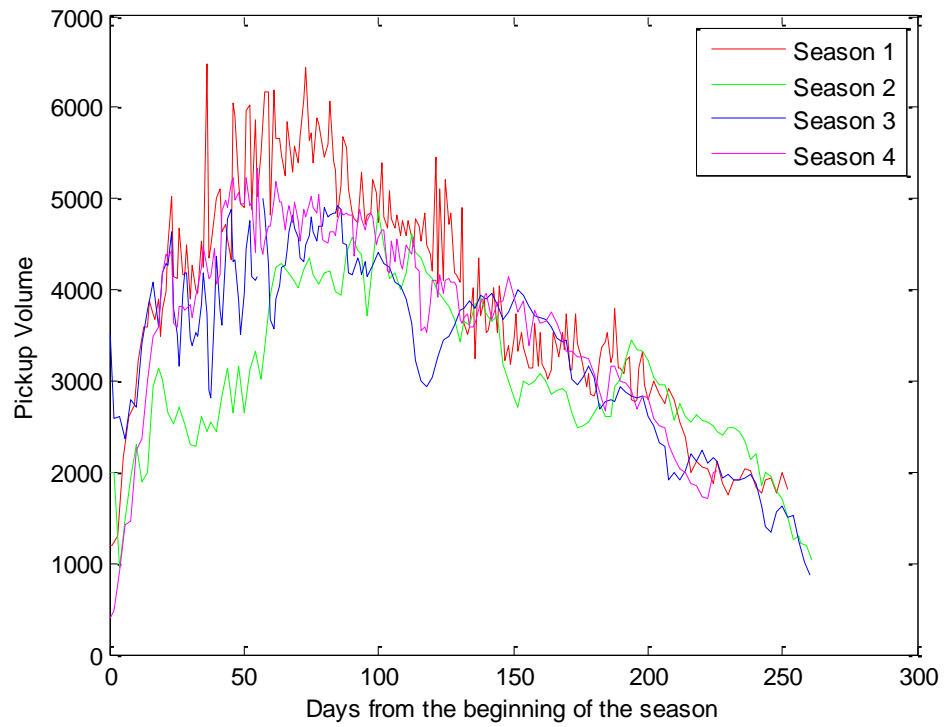


Figure 7.13 Pickup volume across seasons

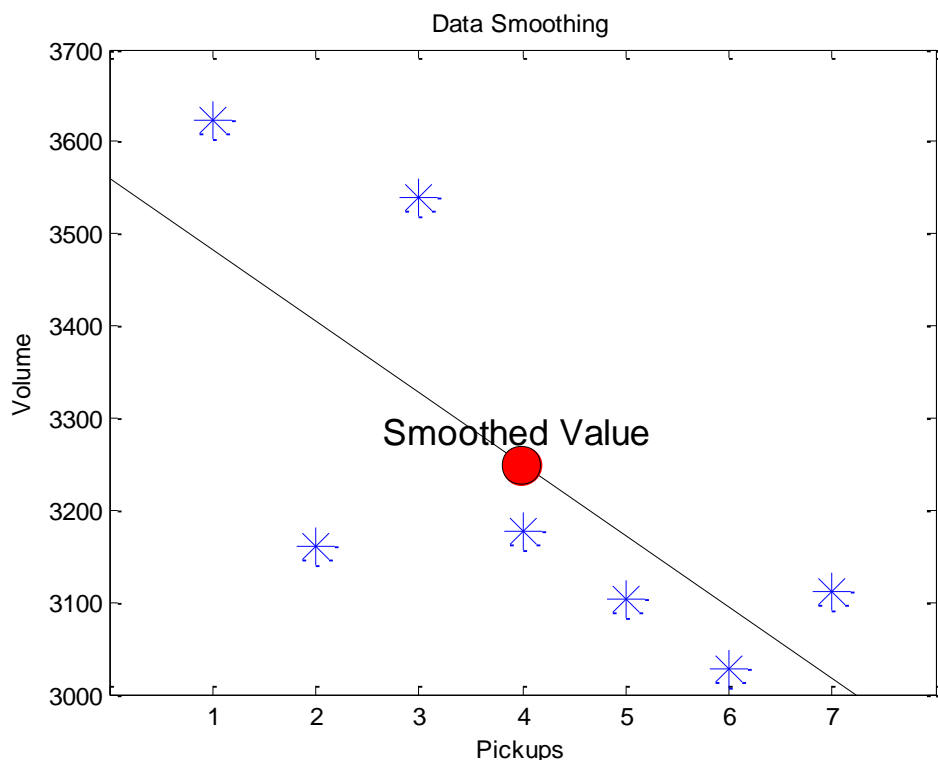


Figure 7.14 Data smoothing illustration

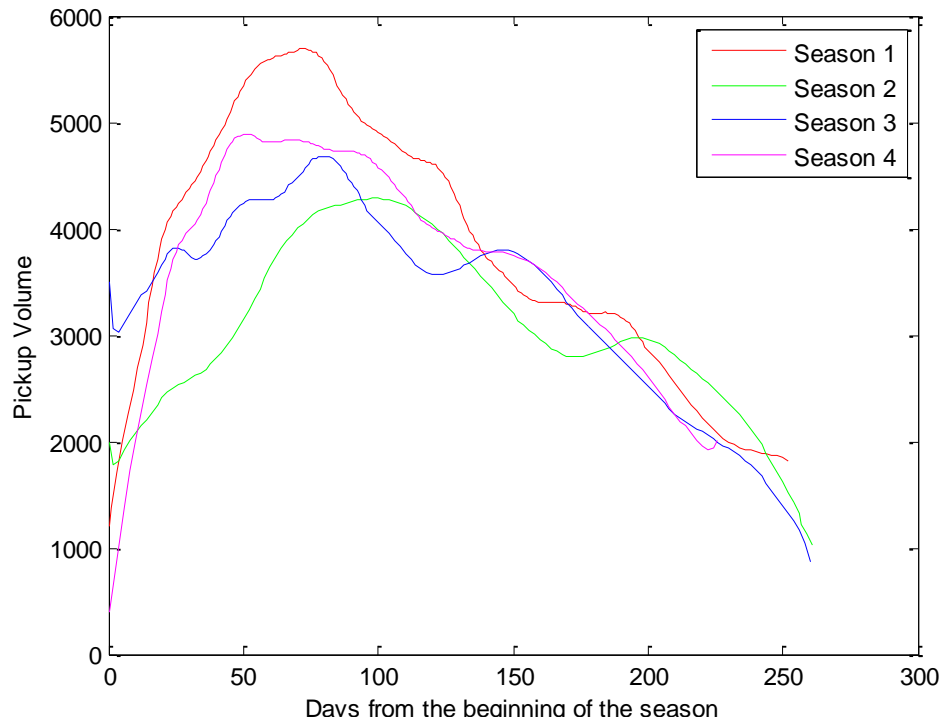


Figure 7.15 Pickup volume data smoothed using the second-order linear smoothing method

Suppose the data set has data x_i , $i = 1, 2, \dots, Nd$.

For obtaining smoothed data y_i a linear function Fun_i is created on data x_j , $j = i-n_1, i-(n_1+1), \dots, i, i+1, \dots, i+n_2$;

$$y_i = [1, n_1+1] * Fun_i \quad (7.1)$$

where: $n_1 = \min(i-1, N_s)$ and $n_2 = \min(Nd-i, N_s)$. N_s is a parameter of the smoothing method which defines the number of pickup data to be used before and after the current date; $N_s = 3$ in Figure 7.15. The min function is required for smoothing values at the beginning and the end of the season where there may not be enough input vectors as specified in N_s .

$$Fun_i = \text{inv}(X_i' * X_i) * X_i' * Y_i \quad (7.2)$$

where “inv” is the function for matrix inversion.

$$X_i = [\text{ones}(n_1 + n_2 + 1, 1), [1:n_1 + n_2 + 1]'] \quad (7.3)$$

where “ones” is a function that creates a matrix of ones.

$$Y_i = x_j = x(i-n_1: i+n_2) \quad (7.4)$$

The 'Second-order' means such a smoothing method is applied twice to provide a higher smoothing effect. The result of this second order smoothing of the raw data in Figure 7.13 is shown in Figure 7.15.

7.6 Multi-Model System (MMS)

The final system is a multi-model system based on the one proposed in CHAPTER 6 that integrates both temporal and spatial models using a model contribution optimisation module as shown below:

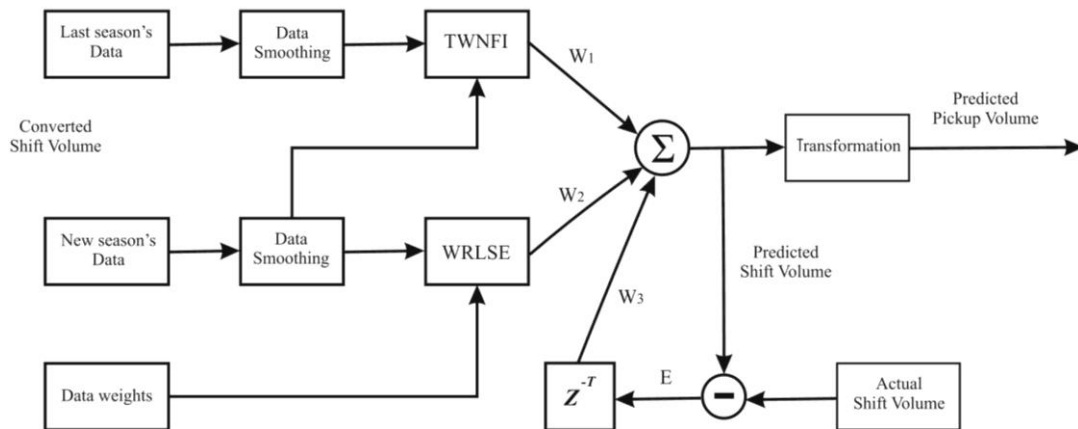


Figure 7.16 The system architecture. The online model is self-improving through correction of the two models contribution weights toward the final output.

7.6.1 Spatial Model - TWNFI

TWNFI is one of the two models combined to constitute the multi-model system that performs the 4-days-ahead prediction. In this system, TWNFI uses eight days' production volumes ($PVD(t-7:t)$), to perform its prediction of the 12th days volume ($PVD(t+4)$).

The TWNFI model is trained with the data from the last two seasons' smoothed data where $PVD(t-7:t)$ is used as the inputs and $PVD(t+4)$ is used as

the output. It is therefore unable to provide a prediction for where data is insufficient, in the beginning of the season prior to the 8th day's volume.

Once the model is trained, it is then applied on the latest eight days production volume data to predict the production volume in four days time. Whenever a new production volume is available from the new season, it is smoothed with the new season's data and then included as part of the training data for the next prediction.

7.6.2 Temporal Model - WRLSE

WRLSE creates a linear function with the first eight days' production volume PVD(1:8) and then modified with the subsequent data online. For every modification the production volume data from the past 8 days are used and such data are smoothed with the method described in 7.5 . The weights, 0.65 – 1.0 with 0.5 intervals are taken for data PVD (t-7:t) respectively and 0.9 is set up as the forgetting factor.

7.6.3 Error Measurements

The following error measurements were used to evaluate the system and they were also used for comparing the multi-model system with a linear regression model

Mean Error (ME)

$$ME = \frac{1}{n} \sum_{i=1}^n (forecast(i) - Actual(i)) \quad (7.5)$$

Mean Absolute Error (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |forecast(i) - Actual(i)| \quad (7.6)$$

Standard deviation of Error (StdE)

$$StdE = \sqrt{\frac{1}{n-1} \sum_{i=1}^n \left[(forecast(i) - Actual(i)) - \frac{1}{n} \sum_{i=1}^n (forecast(i) - Actual(i)) \right]^2} \quad (7.7)$$

In the above formulas, $i = 1, 2, 3, \dots, n$, n is the number of predicted data; forecast (i) is the i^{th} forecasting volume and Actual(i) is the i^{th} actual volume.

The prediction accuracy from each model varies during the season, but MMS was the most accurate model across the entire season.

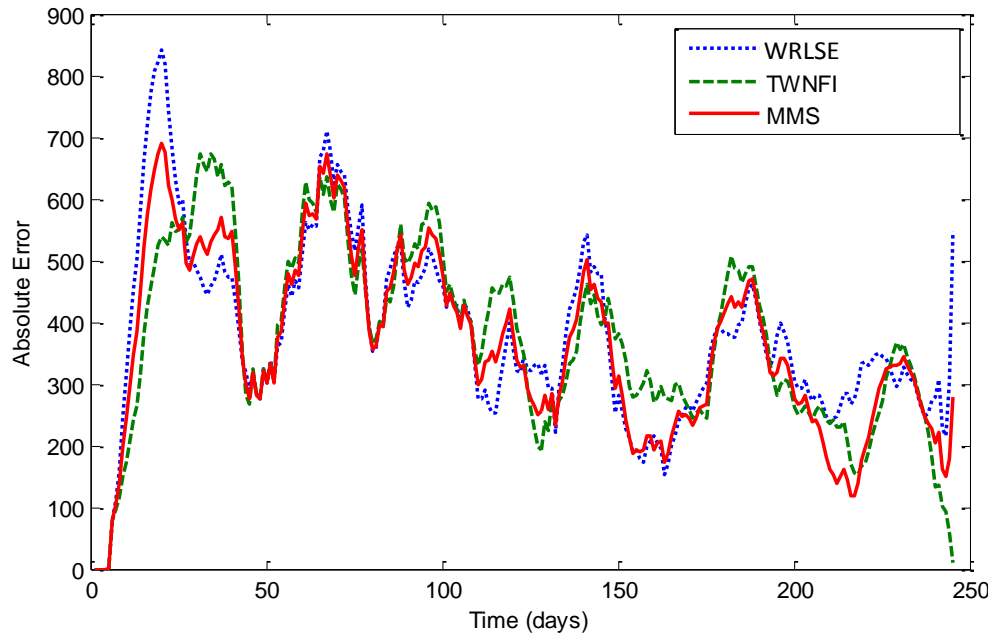


Figure 7.17 Absolute error from each model in MMS across one season. TWNFI and WRLSE performs very differently, each can be significantly better than the other at different time. The MMS framework allows better overall prediction accuracy to be achieved than using either system alone.

7.7 Experimental Results and Comparison

Twelve farms were randomly selected; each of them has at least three seasons' data. For each farm, the data from the first two seasons were used for training and the third season was used for both training and testing.

The MMS system was applied to predict one, two, three and four days. The training data for the TWNFI model consists of two parts: one is the first and second season's data and the other is the data from the first day to the current day ($1: t_0$) in the third season's data. The WRLSE model uses the previous eight days' volumes ($t_0 - 7: t_0$) with their weights. The prediction results with

comparison to a general linear regression model (LR) are shown in Table 7.4, Table 7.5 and Figure 7.18. The average errors of the four-day-ahead predictions on the whole 575 farms' data are listed in Table 7.6. Figure 7.19 shows the actual daily production volumes and 4-day-ahead prediction volumes produced by MMS on the 3rd season of a selected farm. The prediction errors (MAE) on such data are shown in Figure 7.20 (for MMS) and Figure 7.21 (for LR).

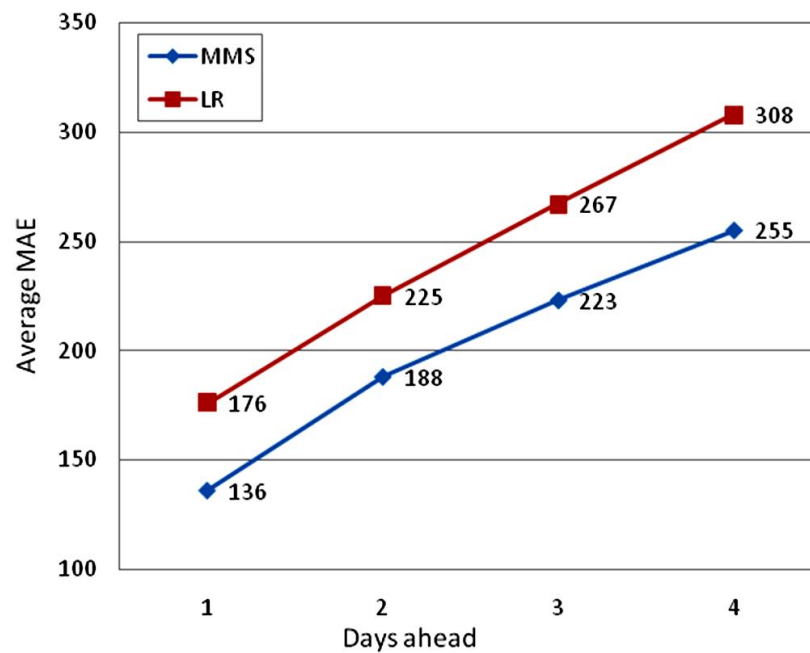


Figure 7.18 A comparison between the MMS and Linear Regression. Average MAE of 1 to 4 day-ahead prediction results on 12 randomly selected farms.

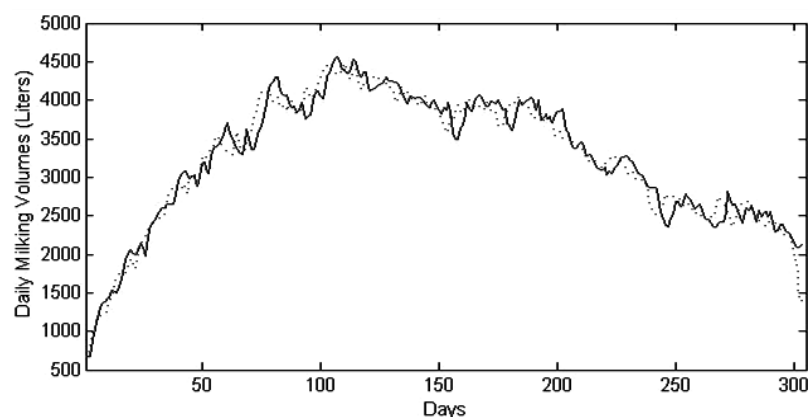


Figure 7.19 Predicted and actual daily production volumes of a random farm (3rd season) (Dotted line: Actual Volumes; Solid line: MMS Predicted Volumes)

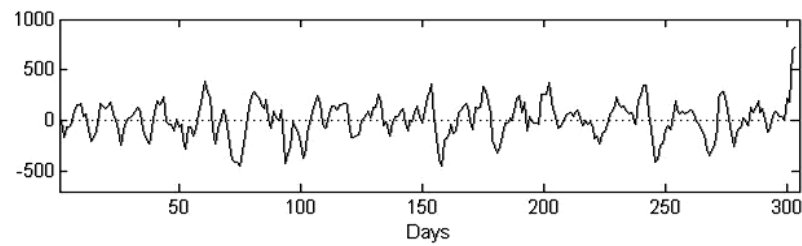


Figure 7.20 ME of predicted volumes by MMS (the 3rd season)

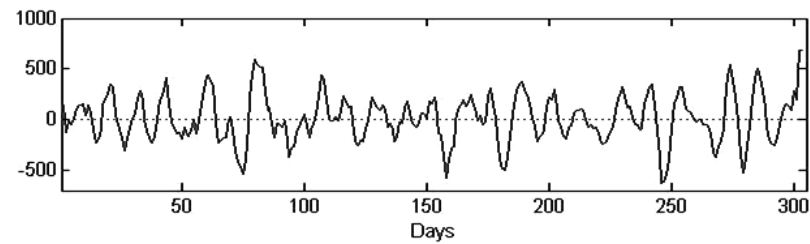


Figure 7.21 ME of predicted volumes by LR (the 3rd season)

Mean Error is used to see whether the model is constantly under or over predicting.

Table 7.4

Comparative analysis of the 1 to 4 days ahead daily prediction error between the proposed MMS and the currently used LR models on 12 farms' data (averaged)

Days	ME		MAE		StdE	
	MMS	LR	MMS	LR	MMS	LR
1	11	9	136	176	219	270
2	18	13	188	225	297	345
3	21	19	223	267	351	413
4	26	24	255	308	402	480

The table above shows that MMS

The overall prediction error is not the only performance measure in this case study. For practical use, the chosen model needs to be stable and must not seriously over or under predicting the output.

The two figures and the table above show the prediction error of both MMS and LR at every time point. The mean error should be around zero, which indicates that the models are not over or under predicting the output values.

Mean Error (ME), indicates whether the model is over or under predicting the output. Positive values mean the model is over predicting and negative values means the model is under predicting. The results show that both models are slightly over predicting the output with MMS a little more so.

Mean Absolute Error (MAE), indicates the how well the model performs. The result shows that MMS performs better than LR.

Standard Deviation of Error (StdE) indicates the variation of error. Lower variation is better since it means less reserve space is needed to ensure all milk gets picked up.

Table 7.5

4-day ahead prediction error on 12 randomly selected farms' data

Farms	ME		MAE		StdE	
	MMS	LR	MMS	LR	MMS	LR
1	103	90	473	578	702	829
2	23	38	356	426	456	537
3	22	16	141	181	227	322
4	22	24	142	188	195	250
5	12	13	103	136	130	167
6	10	6	62	73	84	99
7	7	5	63	63	80	82
8	-2	17	519	603	686	788
9	68	27	320	406	402	518
10	-5	14	360	422	461	550
11	18	19	165	196	215	251
12	8	7	162	185	208	242
Average	26	24	255	308	402	480

Table 7.6

Four days ahead prediction error (average) on 575 farms' data

ME		MAE		StdE	
MMS	LR	MMS	LR	MMS	LR
32	33	331	405	528	650

The results from MMS shows that it is suitable for the task, as it provides consistently better prediction accuracy and adaptability to new data for each individual farm.

Comparing MMS with the currently used LR method on the whole set of 575 farms, MMS is about 22% more accurate than LR. The StdE of the MMS is also lower, which that indicates that MMS is more stable and reliable.

Overall, MMS performs better than LR in respect to the requirements of the case study, making it a better solution.

7.8 Conclusions

This chapter details the data analysis and modelling process for a milk production volume prediction problem. The proposed model results in 22% increase in accuracy than the currently used LR model and constitutes a more stable and reliable predictor. In event of a change in pattern across the season, the multi-model approach allows dynamic weight adjustment based on previous prediction error.

Overall, the proposed MMS model and prototype software system have several advantages when compared to the LR model:

- The MMS model is a more accurate and stable predictor
- The MMS model is adaptive to any new data and new variables.

CHAPTER 8 PERSONALISED REGRESSION MODEL WITH INCREMENTAL FEATURE SELECTION

Global regression models such as linear regression have been widely used in real world biomedical classification problems due to their simple equations and their ability to work well in problems with input noise, which is often the case for many clinical data that involves patient opinion, feelings and behaviour.

One of the issues with global regression models is that unique local problem subspaces may be treated as noise if they do not fit the global pattern. However, in some biomedical problems, it is logical to assume that there are various subgroups of patients who may be disease free or have a disease for different reasons to the majority, but they may not contribute to the solution if there are not enough other patients of the same group.

In the previous chapters, the emphasis has been put on the local “mixture of models” approach to create multiple models, with each model being able to contribute at different levels depending on the location of the test input vector within the global problem space. The above approach may not work in a biomedical problem because the definition of the problem space may not be sound due to the great number of irrelevant or less important variables in a problem with hundreds or thousands of variables.

Feature selection must be applied to create a subset of variables that are relevant to the problem. Even so, the noise in these variables is likely to be high and the variables may be only partially relevant to the problem.

In this chapter, a personalised regression model is proposed that creates a personalised model dynamically for each prediction by optimising the global model to a reasonable degree using input vectors that are relevant to the prediction at hand (Kasabov & Hwang, 2008). This hybrid approach maintains the

stability provided by the global model, while putting emphasis on the problem space nearest to the test input vector. If the variables selected to define the problem space are reasonably good, this approach should be able to increase the accuracy of the global model for the test input vector.

8.1 Algorithm

The algorithm for the personalised local regression model can be briefly described below:

1. Perform univariate analysis on the variables to rank the variables based on their discriminating power using appropriate statistical test for the variable type.
2. Perform incremental feature selection to obtain a best set of variables that allows the global regression model M to achieve the highest accuracy. The quality of the regression model is based on the Area Under ROC curve (AUC) (explained later in this chapter)
3. For each input vector (i)
 - a. Select k neighbours (nbr) from training input vectors using only the selected set of variables from step 2.
 - b. Use the global regression model as the base model and optimise its coefficients using (nbr) to obtain the local regression model (L) for the current input vector (i)
 - c. Perform the prediction on the current input vector (i) with the local regression model (L)

The size of the neighbourhood is a parameter of the algorithm. When the size is small, the solution derived will be very specific to the testing input vector. When the size is large; the solution will be more general and closer to a global model.

In real world data modelling problems with imbalanced dataset, one may want to increase the size of the neighbourhood to obtain at least some coverage

of input vectors from all classes. On contrast, if all classes are equally well presented in the problem space and are reasonably separated, then perhaps only the nearest few input vectors will be sufficient to solve the problem.

8.2 Feature Ranking

As for all distance based algorithms, it is important to define the problem space using only the variables that are relevant to the problem.

It is, however, difficult to know which of the variables should be removed. Univariate analysis can be used to identify how each individual variable can be used to discriminate between two classes but it is possible for two variables with moderate univariate discriminating power to perform better than a single variable with high discriminating power (Guyon & Elisseeff, 2003) as demonstrated in Figure 8.1.

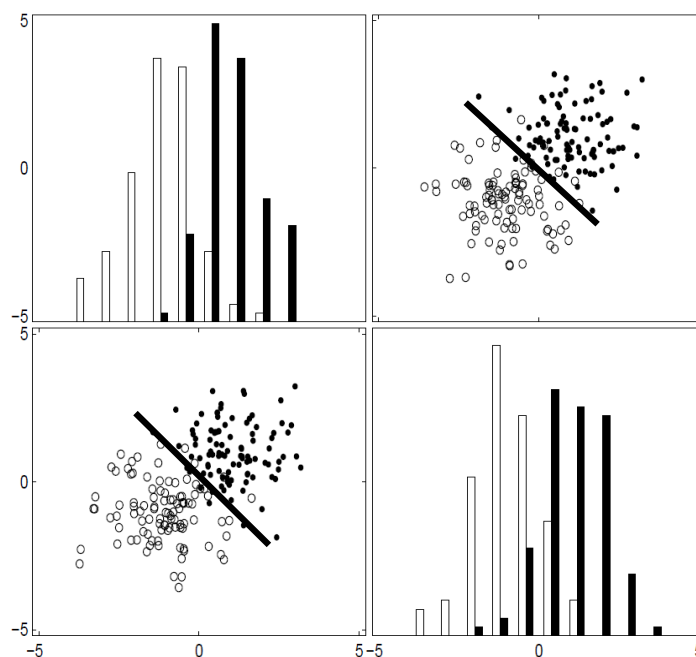


Figure 8.1 Synthetic data with two classes and two variables. Univariate analysis shows both variables have moderate discriminating power but when combined, the discriminating power increases significantly, copied from an overview of feature selection issues (Guyon & Elisseeff, 2003).

The safest method to identify the best combination of variables that obtain the highest discriminating power is to do an exhaustive search of every possible combination of variables and measure how the model performs. This approach is often not practical on problems with a number of variables that is large, e.g. microarray datasets (Shipp et al., 2002).

Univariate analysis nevertheless provides a rough estimate of which variable may have high discriminating power when combined, as the variable should at least be moderately useful in univariate analysis if it is to be potentially useful in multivariate analysis.

Previous studies on the variables should also be taken into account during the variable ranking process, especially when the number of input vectors is not sufficiently large. However, it should not be used as an exclusion criterion as there may be new discoveries on the use of these variables and the variables that were not found to be useful in the previous studies may still contribute to the current study on their own or combined with new variables previously unavailable.

In this implementation, the variables are ranked as follows

1. Sort variables based on their p-values, the lower the p-value the higher the rank and call this Variable Set 1 (Set1)
2. Highlight the variables that were identified as having high discriminating power and put them in Variable Set 2 (Set2), sort these based on their p-values from the previous univariate analysis.
3. Remove variables that exists Set 2 from Set 1 and join Set 2 with Set 1 to form a final variable set. (fset)

8.3 Incremental feature selection

The goal here is to identify a subset of features F that gives the highest accuracy to the global model. The procedure is described as follows:

1. Set the desired baseline accuracy b and increment threshold t , F is empty at this stage.
2. Starting with the highest ranked variable, $f(i)$
 - a. Create a model with combined Variable Set F and $f(i)$.
 - b. If the AUC on the training data is above 0.5 and the current AUC (i) is higher than the best AUC by threshold t , then keep the variable by adding $f(i)$ to F , otherwise move on to the next variable.
 - c. When the last variable is reached, start again with the highest ranking variable that has not yet been selected.
 - d. Repeat a - c until AUC does not increase using every unselected variable.
3. End of procedure. F is the final subset of variables that gives the highest accuracy to the global model.

8.4 Personalised regression Model

The best performing feature subset F provides a level of confidence that the problem space defined by this set of variables is meaningful. The next step is to optimise the global model by adjusting its coefficients to minimise the RMSE on training input vectors that are similar or close to the test input vector (neighbours) using back-propagation optimisation.

The details of the optimisation algorithm are shown below:

Consider the data is composed of n data pairs with m input variables and one output variable $\{[x_{i1}, x_{i2}, \dots, x_{im}], y_i\}$, $i = \{1, 2, \dots, n\}$, $j = \{1, 2, \dots, m\}$

The global linear regression model M is in the form of

$$y_i = a_0 + a_1x_{i1} + a_2x_{i2} + \dots + a_mx_{im} \quad (8.1)$$

The constant and coefficients are updated with equations (8.2) and (8.3) to minimise the following objective function(8.4)

$$a_0(k+1) = a_0(k) - \eta_0 \sum_{i=1}^n (y_i - t_i) \quad (8.2)$$

$$a_j(k+1) = a_j(k) - \eta_j \sum_{i=1}^n x_{ij} (y_i - t_i) \quad (8.3)$$

$$E = \sum_{i=1}^n (f(x_i) - t_i)^2 / 2 \quad (8.4)$$

where

- learning rate: η
- predicted output: t_i
- training iteration: $k=1, 2, \dots$

The end of this procedure provides a local model L , which gives the highest prediction accuracy on the input vectors that are similar to the test input vector. The model L is then used to derive the output for the test input vector. Since every test input vector has different set of neighbours, the local model for each of them is also different.

8.5 Error Measure

In prediction problems as in earlier chapters, the model's accuracy is determined by the difference between the predicted value and actual value; mean absolute error (MAE), root mean square error (RMSE) and non-dimensional error index (NDEI) are the most commonly used methods for measuring error in prediction problems. However, in case of classification problems, the predicted output of the model is risk and the actual output is binary; whether a patient has the disease or not. It is therefore not appropriate to compare models using the difference between the predicted output and actual output. For example, two patients A and B are both diagnosed with lymphoma, patient A has been predicted to have a risk of 15% and patient B has been predicted to have a risk of 20% with a prediction model before they were diagnosed. If a model considers all patients with a risk factor above 30% as

having the disease, then patient A and patient B have the same incorrect prediction: “healthy” and the errors for both patients are the same at 100%. If this is measured with using RMSE, the RMSE for patient A would be much higher than for patient B, which is not meaningful here in classification problems.

In order to compare the prediction accuracy of the model, it is necessary to convert the risk factor to a classification outcome using a threshold. The Receiver Operating Characteristic (ROC) curve (Fawcett, 2004; Hanley & McNeil, 1983) is a commonly used technique that shows the true positive and false positive rate at different thresholds. Researchers or Doctors can look at the curve and then decide where the threshold for the true positive and false positive rates is most appropriate to the problem.

The greater area under the ROC curve means better potential overall true positives and less false positives rate at any conversion ratio and, therefore, a better model. The area under the ROC curve (AUC) is then used for comparing the performance of the models.

Low RMSE can usually be translated to high AUC in most cases but the translation is not very precise. If there are two models that give similar RMSE, it is not possible to identify which one would give higher AUC without actually calculating it based on its predictions.

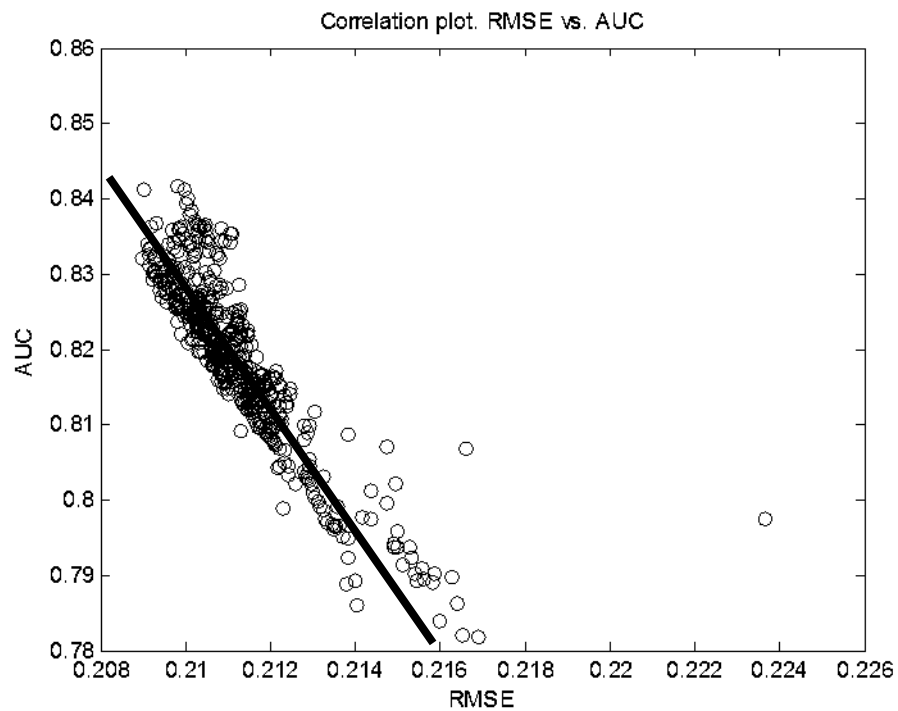


Figure 8.2 Correlation plot between RMSE and AUC during training. Low RMSE can be translated to high AUC but the translation is not 1 to 1. Minor changes in RMSE may not affect AUC and vice versa.

It is not possible to use AUC as the objective function in back-propagation optimisation. AUC is constructed based on the shift of the cut-off threshold between true positive (TP) and false positive (FP) on a set of predicted risks, therefore the changes in prediction does not always immediately translate to changes in AUC, even though they are highly correlated (see Figure 8.2).

However, some other optimisation algorithms may be used to optimise the global function with AUC as its objective function. Genetic Algorithm (GA) and Particle Swarm Optimisation (PSO) are the two optimisation algorithms that may be used. However, these two algorithms are computationally expensive, non-deterministic and the results may not be repeatable, they are, therefore, not used in this method.

8.6 Conclusion

The proposed method is designed especially for problems with high dimensionality, high noise and an average size of data. It can, however, be applied to any classification problem. The emphasis is on defining a sound problem space through extensive feature ranking and selection using the wrapper method. Once the problem space is defined, the global model is then optimised for each test input vector using training input vectors in the neighbourhood.

The proposed method addresses or minimises the following real world data modelling issues:

- Unique problem subspaces

The personalised regression model puts emphasis on the problem space close to each patient.

- Outliers

The final model is optimised for the local space only. Outliers are not likely to be similar to a normal input vector and most likely to be left outside the neighbourhood.

- Imbalanced data

The ROC curve is used as a model fitness measure instead of RMSE, therefore minimising the impact of imbalanced data.

- Irrelevant features

Extensive feature ranking and incremental feature subset selection reduce the chance of using features that do not contribute to the model.

In the next chapter, this method is applied to a real world case study – pregnancy outcome prediction to validate the usefulness of this method with minor modifications, to make use of additional information from the data.

CHAPTER 9 PERSONALISED REGRESSION MODELS FOR PREGNANCY OUTCOME PREDICTION BASED ON SCOPE DATA

9.1 Problem Overview

The overall goal of the SCOPE study (<http://www.scopestudy.net> or see Appendix B for a brief introduction to the study) is to produce a clinically useful screening test for three late pregnancy conditions: preeclampsia (PE), small for gestational-age (SGA) and spontaneous preterm birth (sPTB). The project targets women in their first pregnancy and collects comprehensive clinical and biological data at 15-weeks and 20-weeks. The data source is not limited to the mother; the father and the grandparents are included in the study as well.

“Preeclampsia is a disorder that occurs only during pregnancy and the postpartum period and affects both the mother and the unborn baby. Affecting at least 5-8% of all pregnancies, it is a rapidly progressive condition characterized by high blood pressure and the presence of protein in the urine. Swelling, sudden weight gain, headaches and changes in vision are important symptoms; however, some women with rapidly advancing disease report few symptoms.” (Preeclampsia, 2008)

“SGA refers to a fetus that has failed to achieve a specific biometric or estimated weight threshold by a specific gestational age. Various thresholds (2.5th, 3rd, 5th, 10th, 15th and 25th centiles and 1.0, 1.5 or 2.0 standard deviations below the population average) are used for various fetal measures. The commonly used threshold is the tenth centile for abdominal circumference and estimated birth weight.” (Coomarasamy, Gee, Marlow, & Walkinshaw, 2002)

Preterm birth refers to the delivery of the baby before 37 completed weeks of gestation. Most mortality and morbidity affects very preterm infants. (Tucker & William McGuire, 2005)

Each one of these pregnancy disease is a significant research field on its own and therefore this thesis is limited to the problem of predicting preeclampsia (PE) only using the 15-week clinical dataset.

9.2 Data Description

The project has enrolled 2,512 women from New Zealand and Adelaide, Australia, who are in their first pregnancy. Patients who have had more than two abortions are excluded from the study.

Patient history, paternal data, parental data and 15-week patient clinical data are included in the dataset. The data involves a wide range of variables, from standard pregnancy check up variables to other possibly relevant variables such as diet, exercise, stress and working conditions.

Many neural networks and statistical methods do not work well with unordered categorical variables and table data and therefore these variables are converted into one or more binary, ordinal or continuous variables.

After significant variable merging and purging performed with extreme care by the SCOPE team, which consists of medical doctors, statisticians, bioinformaticians and neural network researchers, the number of variables was reduced to 524, which was then used as the final dataset for modelling purposes. The output for the dataset is either 1 or 0 where 1 indicates the patient has the disease and 0 means the patient did not have the disease.

Each variable is given one of the following levels of importance based on previous studies in the field.

- Level 1: Variable showed consistent increase or decrease in obstetric risk in two or more studies
- Level 2: Variable showed small increase or decrease in obstetric risk in one or more studies.

- Level 3: studies performed but no significance found or the findings were inconsistent across multiple studies.
- Level 4: No previous studies identified.

This information is used as part of the feature selection process to improve the variable rank based on their type. Level 1 being the most important followed by Level 2 and then Levels 3 and 4.

Many variables are also highly correlated due to the design of the study. There can be more than ten questions relating to the same topic. For example, there are several variables describing whether the patient is overweight. These can be weight, height, body mass index, arm circumference, neck circumference and waist size. These variables are designed to measure the same characteristic of the patient. If they are treated as independent variables, this characteristic is literally given more importance than other characteristics, such as cardiovascular fitness and stress. Because of this, the variables are placed into 139 groups based on their intended purpose and only one variable per group can be used in the final model.

9.3 Imbalanced Dataset on Personalised Model

Preeclampsia is a low prevalence disease with only 5% of patients being in the disease group. This translates to about 126 patients having the disease in the entire dataset. With the disease positive input vectors sparsely positioned in the problem space, it is highly likely that some high risk disease regions are not well supported.

This issue causes problems in all types of models, especially in local and personalised models. If the distance threshold is set too small, the clusters or the neighbourhoods may miss out a disease input vector that falls just outside the radius and become a zero-risk region when it should be a low or medium risk region instead. In reality, it is not possible for one person to have zero-risk of these diseases in any case.

One way to minimise the impact of this issue is to enforce a certain number of disease input vectors in the training data by increasing their radius. This minimises the probability of having a high disease risk region labelled as 100% healthy due to a lack of disease input vectors to support it, while reducing the use of non-relevant input vectors in the training of the model.

9.4 Method Application

The personalised regression method proposed in the previous chapter is applied in this case study with minor modifications to the feature selection process to make use the variable groups and variable correlation that are specific to this case study.

Ten-Folds Cross Validation (CV) (Kohavi, 1995b) was used to measure the generalisation error of the model and AUC is used to measure the performance of the model.

In the Ten-Folds Cross Validation method, for each fold, the data is split into a training dataset and a testing dataset with 90% for training and 10% for testing. The test input vector is never reused in the subsequent folds.

The details of the algorithm are described below:

9.4.1 Data Normalisation

The input variables are different ranges and this may leads to over emphasis on the variables with wider range and letting them overpower other variables.

The following data normalisation method is applied.

$$v_i = \log \left(\frac{v_i - \min(v_i)}{\max(v_i) - \min(v_i)} + 1 \right), i = 1, 2, \dots, m \quad (8.5)$$

This procedure first normalises the range of each variable to be between 1 and 2 to ensure equal weight between variables. This is done by shifting the lowest value in the variable to 0 and then divide the all values to the maximum value of the variable after the shift. The normalised values are then log transformed to minimise the affect of potential outliers. m is the number of variables. \max is the function to obtain the maximum value in a vector. \min is the function to obtain minimum value in a vector. The resulting variable will be in the range of 0 and $\log 2$.

9.4.2 Variable Ranking

The variables need to be ranked prior to the incremental feature selection process so it can start with the most important variables. The ranking method takes into account univariate analysis and previous studies.

The details of the variable ranking procedure are described below:

1. Identify variables that have approximately zero standard deviation in either class and have the same value. Remove these variables from the variable list (FLIST). This step removes variables that have no usefulness in model building.
2. Calculate p-values for the variables in the FLIST by applying t-test (Holtsberg, 2000), Chi-Square test (Weisstein) or Fisher's exact test (Trujillo-Ortiz et al., 2004) depending on variable type.
3. Sort FLIST based on the p-values.
4. Calculate correlation between variables to identify a group of variables that are highly correlated or identical ($p > 0.95$). Keep the variables with the lowest p-value and remove the rest from the FLIST.
5. Create the following sets of variables
 - a. Variable Set 1 = Variable level 1 followed by Level 2
 - b. Variable Set 2 = Variables with p-values below 0.05
 - c. Variable Set 3 = Intersect between Variable Set 1 and 2.

- d. Remove variables in Variable Set 3 from Variable Set 1 and 2.
- e. Variable Set 4 = Variable Set 3 + 1 + 2 (in this order)

The rationale for step 5 is to rank variables with low p-value and shown significance in prior studies on the top of the list, followed by variables that shown significance in prior studies and then the rest of the variables ranked by their p-values.

Variable Set 4 is the set of variables that will be passed on to the incremental feature selection algorithm.

9.4.3 Incremental Feature Selection

A threshold based incremental feature selection method is applied to the training dataset to reduce the number of variables in Variable Set 4. The goal is to ensure that only variables that benefit the model by an acceptable margin are selected.

This part of the algorithm defines the problem space and therefore has direct impact on the selection of the subset of training input vectors for a given test input vector, which is the basis of the individualised model or personalised model when applied on clinical patient data.

The details of the algorithm are described below:

1. Create an empty Variable Set 5.
2. Go through variables in Variable Set 4 from the ones ranked highest to lowest, one at a time, add current variable to Variable Set 5 and do the following.
 - a. Build a linear regression model on the training dataset using only variables in Variable Set 5.
 - b. If the AUC is above 50% and increment of AUC higher than 0.5% of the total area, keep this variable in Variable Set 5 and remove this

variable from Variable Set 4. Otherwise, remove this variable from Variable Set 5.

3. Repeat step 2 until the last variable in Variable Set 4 has been processed.
4. Repeat step 2 and 3 until no more variables have been added to Variable Set 5.

The algorithm starts by including the highest ranked variable that is able to achieve an AUC above 50% and then adds the subsequent variables if the combination of the variables can achieve an AUC 0.5% higher than the previous AUC.

9.4.4 Model and Prediction

A personalised model is created by optimising the global model using a subset of input vectors that are near the test input vector.

1. Perform linear regression on the training data with Variable Set 5 to create a global model, M
2. For each test input vector, t , select a subset of training input vectors to be the training data, H_t
 - f. All training input vectors within the distance between the test input vector and the 10% percentile of disease positive input vectors is selected as the training data.
3. Update M using the steepest descent method to minimise error produced by the input vectors in the neighbourhood, to generate model M_t .
4. Apply M_t on t to obtain the prediction.

9.5 Results

With the proposed personalised model, the following results were obtained:

Table 9.1

Result of the proposed personalised model.

Personalised Model PET: AUC 10-Fold CV	
Folds	Test
1	67.19%
2	68.52%
3	70.56%
4	65.38%
5	67.61%
6	75.69%
7	64.29%
8	69.88%
9	75.34%
10	77.70%
Average	70.22%
Std.	4.59%

Since the test input vectors are never reused in the 10-fold cross validation, a prediction has been made for every input vector. Hence the predicted output can be aggregated and plotted as follows:

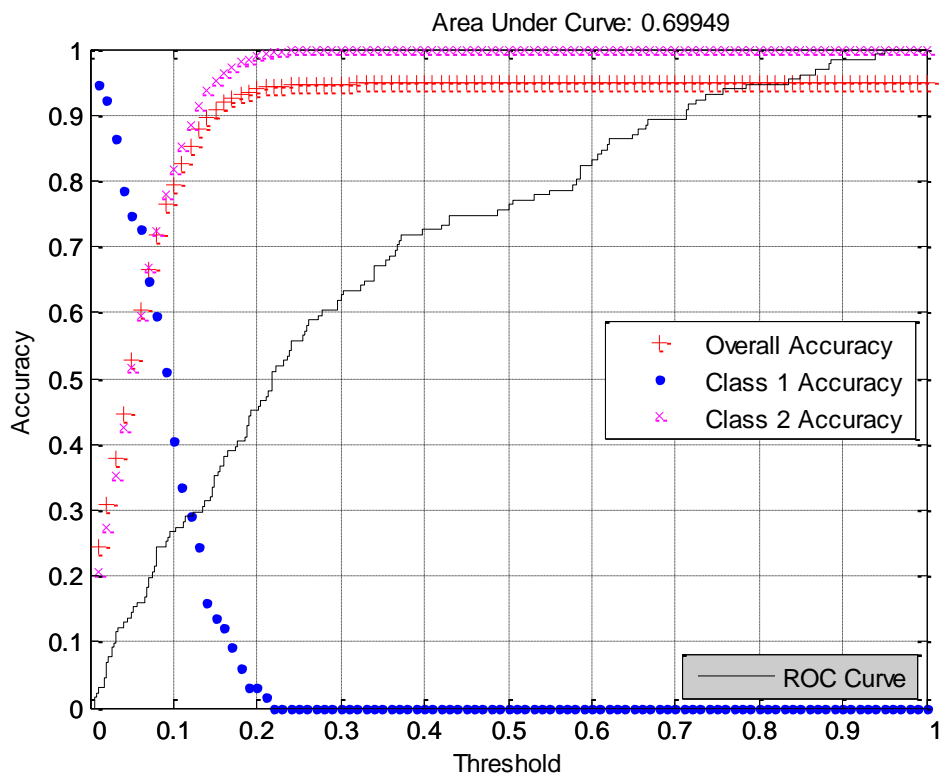


Figure 9.1 Prediction accuracy at different threshold including class 1 accuracy (disease), class 2 accuracy (healthy), overall accuracy, ROC curve and area under the ROC curve.

For comparison purposes The statistics group in the SCOPE team use a logistic regression model with incremental feature selection using Akaike's Information Criterion (AIC) (Akaike, 1974) with an identical 10-fold cross validation.

Table 9.2

Results of the logistic regression model with AIC feature selection.

Logistic Regression with AIC PET: AUC 10-Fold CV		
Folds	Training	Test
1	78.00%	58.50%
2	76.30%	68.00%
3	73.90%	66.00%
4	74.40%	65.20%
5	71.60%	71.50%
6	75.50%	83.80%
7	76.20%	70.50%
8	76.70%	73.90%
9	75.20%	67.90%
10	73.60%	68.10%
Average	75.10%	69.30%
Std.	1.74%	6.22%

The results in Table 9.1 and Table 9.2 shows that the personalised regression model is more accurate than the logistic regression model and the variations in prediction accuracies across the folds are significantly lower. This means the personalised model is, in fact, more stable. This is likely to be caused by the optimisation of the global model with input vectors located in each test input vector's residing problem subspace.

9.6 Chapter Conclusion and Discussion

In this chapter, a linear regression based personalised model with embedded incremental feature selection is proposed.

The algorithm consists of two parts:

1. In the first part:

The best set of features is identified through an incremental feature selection utilising both univariate analysis and variable importance from previous studies. A good set of variables is critical for personalised modelling in the second part, as it defines the problem space for the test input vector and its neighbourhood.

2. In the second part:

A baseline global linear model is created from the training data, which is then optimised based on the subset of training input vectors that are in the vicinity of the test input vector. If the feature set has been defined well in the first part of the algorithm, the selected subset of training input vectors should be highly relevant to the test input vector and therefore allow a better model to be built.

The proposed method, with a simple personalised approach to the problem, was able to achieve slightly better results than the logistic regression approach with AIC method.

The clinical data used in this study contains a lot of noise, possibly due to the fact that many questions cannot be answered with precision. The amount of stress, for example, is subject to personal judgement. It will be ideal to include biological data, such as protein or gene data, from all patients as part of the study, as they may have less noise and bias from human perception.

The blood sample from each patient was collected at their first visit to the clinic. It is, therefore, possible to obtain the Protein or DNA data and include them to the analysis, which may add value to the overall prediction accuracy. It is however, impractical due to the high cost involved. A series of small pilot studies of up to 24 patients using protein data was conducted, but with such a small sample size, it is difficult to reach any conclusive outcome. If such data becomes available from all patients in the future, it would be ideal to include them in the study.

Additional data may also help improving the model's prediction accuracy. An additional clinical dataset of 1,000 patients has been made available just before the submission of this thesis and therefore cannot be included.

CHAPTER 10 CONCLUSION AND FUTURE RESEARCH

Real world data modelling problems are evidently chaotic and unpredictable, consists of issues that are difficult to manage. Most of the issues can limit the performance of a prediction model and they can only be addressed through careful experimental design and an in-depth understanding of the causes of the issues. Four methods were proposed, two improve on existing methods, and the other two are new ways of applying existing methods. Four case studies on real world data modelling problems were carried out.

10.1 DyNFIS – Dynamic Neuro-Fuzzy Inference System

The first proposed method, “DyNFIS” improves the existing method, “DENFIS” - a fuzzy inference system that utilises clustering information, by using a more sophisticated fuzzy membership function and additional supervised learning on the fuzzy rules’ membership functions and leads to better prediction accuracy. This method was first tested on Mackey-Glass benchmark dataset and then entered in the NN3 11-time-series competition where it achieved 10th place among 90 competitors, even without automated parameter optimisation. DyNFIS is therefore considered to be a generalised and stable method for various problems.

In addition to higher accuracy, DyNFIS also allows better knowledge to be extracted in the form of if-then rules. They represented the data more accurately than DENFIS due to the additional supervised learning and the use of Gaussian membership function allows better linguistic representation of the rules to be made.

10.2 MUFIS: A Novel Neuro-Fuzzy Inference System Using Multiple Types of Fuzzy Rules

With the understanding that many problem subspaces are unique and some problems are better suited for one type of model than others, it was only logical to develop the next method to allow different types of fuzzy rules to be used in a single fuzzy inference system. This method, called MUFIS, was able to outperform DyNFIS on the Mackey-Glass benchmark data and also in a real world case study of renal function prediction (GFR).

The current implementation of MUFIS allows both Takagi-Sugeno and Zadeh-Mamdani type fuzzy rules to be used together in a single fuzzy inference system. The analysis of the suitability of fuzzy rule types for each cluster also gives us additional information on problem subspaces. The assignment of the type of rules highlights the characteristics of the problem subspace. Takagi-Sugeno type fuzzy rule is usually assigned to problem subspaces that are linear. Zadeh-Mamdani type fuzzy rule is usually assigned to problem subspaces that are more chaotic, due to the lack of input vectors or the presence of noise.

10.3 Multi-Model System – Temporal and Spatial

To allow multiple models with different points of view of the problem to be integrated, a “mixture of experts” multi-model system seemed like a logical next step. A multi-model system (MMS), using both temporal and spatial models was proposed to allow contrasting views of the problem. The temporal model addresses the problem using only recent data, looking at changes of patterns to make the prediction. The spatial model uses current patterns as an example and searches historical data for similar patterns to make the prediction. By combining the two through a contribution adjustment module that regulate the contribution from each module based on its prior prediction error allows better prediction to be made than any single model alone.

The multi-model system was applied to a real world seasonal time-series case study in milk production volume prediction. In this case study, WRLSE was used as the temporal model and was responsible for providing the prediction using only recent data and the TWNFI was used as spatial model and made the prediction based on historical data similar to the recent data.

The results of the case study showed that the multi-model system performed significantly better than the linear regression model that is currently adopted by Fonterra.

10.4 Personalised Regression Model

One of the issues with local and distance-based models is that they require good definition of the problem space in order to perform properly, which can be difficult in many biological modelling problems as they often involve a large number of noisy variables. In order to apply local or personalised models to this type of problem, an in-depth analysis of the features is necessary.

A personalised regression model with incremental feature selection was proposed in this thesis. This method applies incremental feature selection on variables that were ranked using univariate analysis and results from previous studies. This set of variables was then used to define the problem space and identify the relevant subset of data for each prediction. The global regression model is then optimised with this subset of training data to put a focus on the test input vector's residing problem subspace.

The method described above was applied to a real world case study on pregnancy outcome prediction. It performed slightly better and was more stable than the logistic regression model using AIC incremental feature selection on the problem of predicting the risk of having preeclampsia using only the 15-week clinical data.

10.5 Summary

This PhD study shows that specially designed models may solve or minimise the negative impact caused by issues such as outliers, missing values, evolving processes and unique problem subspaces. However, many of the issues cannot be addressed without an in-depth understanding of the causes of these issues. In a real world data modelling problem, the data analysis and pre-processing may be as important as the modelling if not more so.

The idea of focusing on the unique problem subspaces appeared to be beneficial to the data modelling problems. It allows identification of sub-problems and allows further studies on them.

10.6 Future Research

The research conducted in this thesis is limited by time and resources available during the study and there is much work to be done to further this research and improve the performance of all proposed generic methods. A list of future work is outlined below:

10.6.1 Automated Parameter Tuning

One of the challenges in both DyNFIS and MUFIS is parameter optimisation. The algorithms have several parameters that have significant impact on the performance. For example, the distance threshold affects the number of clusters, which in turn sets the number of fuzzy rules used to solve the problem.

The parameters are currently optimised manually based on the prediction accuracy of the training data.

10.6.2 Online Learning

DyNFIS and MUFIS may both be further developed as online models. This may be done by using online ECM clustering and slight modification of the existing method.

10.6.3 Improvements to the Multi-Model System

The farm milk production prediction case study shows that most farms follow similar patterns. This indicates that the historical data of farms that are similar may also be useful to the prediction.

The current implementation uses both global and personalised model trained with the current farm's current and historical seasonal data only. The logical next step is to add a local model to the system that specifically looks at the relationship between farms, or between farms' seasonal data.

10.6.4 Additional Data in SCOPE Study

Additional clinical data or biological data should be included in the study when it is made available. Larger dataset allows better coverage of the problem space, particularly for disease patients, and better feature selection; both have strong impact on the prediction accuracy of the personalised regression model.

As described in chapter 9, the features used to define the problem space are crucial to the success of a personalised regression model. The inclusion of biological data may provide certain useful variables that lead to better definition of the problem space and, therefore, better accuracy.

10.7 NeuCom

All novel and generic methods proposed in this thesis should be added to the NeuCom software so it can be utilised easily by the research community all

over the world and further validate its performance on other data modelling problems not covered in this thesis.

10.8 Future Publications

DyNFIS, as proposed in Chapter 4, MuFUS, as proposed in Chapter 5, integrated temporal and spatial multi-model systems, as proposed in Chapter 6 and the personalized regression model, as proposed in Chapter 8, are all planned for journals publication in the near future.

REFERENCES

- Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6), 716-723.
- Angelov, P. (2002). *Evolving rule-based models: a tool for design of flexible adaptive systems*: Springer.
- Angelov, P., & Filev, D. (2002). Flexible models with evolving structure, *Proceedings of the First International IEEE Symposium Intelligent Systems, 2002*. (Vol. 2, pp. 28-33).
- Angelov, P., & Filev, D. (2004). An approach to online identification of Takagi-Sugeno fuzzy models. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(1), 484-498.
- Bellman, R. (Ed.). (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press.
- Bhavani, R., & Adam, K. (2004). Extreme re-balancing for SVMs: a case study. *SIGKDD Explor. Newsl.*, 6(1), 60-69.
- Bishop, C., & Svens'en, M. (2003). Bayesian hierarchical mixtures of experts, *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)* (pp. 57-64). San Francisco, CA: Morgan Kaufmann.
- Bjornsson, T. D., Cocchetto, D. M., McGowan, F. X., Verghese, C. P., & Sedor, F. (1983). Nomogram for estimating creatinine clearance. *Clinical Pharmacokinetics*, 8(4), 365-369.
- Boser, B., Guyon, I., & Vapnik, V. (1992). A training algorithm for optimal margin classifiers, *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152). Pittsburgh, Pennsylvania, United States: ACM.

- Cevikalp, H., & Polikar, R. (2008). Local Classifier Weighting by Quadratic Programming. *Neural Networks, IEEE Transactions on*, 19(10), 1832-1838.
- Chen, S., Cowan, C. F. N., & Grant, P. M. (1991). Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Transactions on Neural Networks*, 2(2), 302-309.
- Chen, Y., Wang, G., & Dong, S. (2003). Learning with progressive transductive support vector machine. *Pattern Recognition Letters*, 24(12), 1845-1855.
- Chen, Z., Huang, L., & Murphey, Y. L. (2007). Incremental Learning for Text Document Classification, *Proceedings of the International Joint Conference on Neural Networks, 2007 (IJCNN 2007)* (pp. 2592-2597).
- Chiu, S. L. (1994). Fuzzy Model Identification Based on Cluster Estimation. *IEEE Transactions on Journal of Intelligent & Fuzzy Systems*, 2(3), 267-278.
- Cnossen, J., van der Post, J., Mol, B., Khan, K., Meads, C., & ter Riet, G. (2006). Prediction of pre-eclampsia: a protocol for systematic reviews of test accuracy. *BMC Pregnancy and Childbirth*, 6(1), 29.
- Cockcroft, D. W., & Gault, M. H. (1976). Prediction of creatinine clearance from serum creatinine. *Nephron* 16, 31-41.
- Coomarasamy, A., Gee, H., Marlow, N., & Walkinshaw, S. A. (2002). *The Investigation and Management of the Small-for-Gestational-Age Fetus*, from <http://www.rcog.org.uk/index.asp?PageID=531>
- Crone, S. F. (2006). *NN3 Neural Network Forecasting competition*. Retrieved 06 November, 2008, from <http://www.neural-forecasting-competition.com/NN3/results.htm>
- Da, D., & Kasabov, N. (2000). *ESOM: an algorithm to evolve self-organizing maps from online data streams*. Paper presented at the Neural Networks, 2000.

IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on.

Damousis, I. G., Alexiadis, M. C., Theocharis, J. B., & Dokopoulos, P. S. (2004). A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation. *Energy conversion, IEEE transactions on*, 19(2), 352-361.

Dasarathy, B. V. (1990). *Nearest neighbor norms: NN pattern classification techniques*: IEEE Computer Society.

Davis, L. D., & Mitchell, M. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold.

Domeniconi, C., & Gunopulos, D. (2001). Incremental support vector machine construction, *Proceedings of the IEEE International Conference on Data Mining, 2001 (ICDM 2001)* (pp. 589-592).

Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3, 32-57.

Fawcett, T. (2004). *ROC graphs: Notes and practical considerations for researchers*.

Fontenla-Romero, O., Alonso-Betanzos, A., Castillo, E., Principe, J., & Guijarro-Berdiñas, B. (2002). Local Modeling Using Self-Organizing Maps and Single Layer Neural Networks. In *Artificial Neural Networks — ICANN 2002* (pp. 142-142).

Franco, A., & Nanni, L. (2009). Fusion of classifiers for illumination robust face recognition. *Expert Systems with Applications*, 36(5), 8946-8954.

Freund, Y., & Schapire, R. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, 37, 277-296.

Garson, G. D. (2008). *Data Imputation for Missing Values*, 2009, from <http://faculty.chass.ncsu.edu/garson/PA765/missing.htm>

Gates, G. F. (1985). Creatinine clearance estimation from serum creatinine values: An analysis of three mathematical models of glomerular function. *American Journal of Kidney Diseases*, 5(3), 199-205.

Goh, L., Song, Q., & Kasabov, N. (2004). A novel feature selection method to improve classification of gene expression data, *Proceedings of the second conference on Asia-Pacific bioinformatics* (Vol. 29, pp. 161-166). Dunedin, New Zealand: Australian Computer Society, Inc.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3, 1157-1182.

Hagan, M. T., Demuth, H. B., & Beale, M. (1996). *Neural network design*. Boston PWS Publishing Company.

Hall, M. (1999). *Correlation-based Feature Selection for Machine Learning*. University of Waikato.

Hanley, J. A., & McNeil, B. J. (1983). A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148, 839-843.

Hastie, T., Tibshirani, R., Sherlock, G., Eisen, M., Brown, P., & Botstein, D. (1999). *Imputing missing data for gene expression arrays*: Division of Biostatistics, Stanford University.

Holtsberg, A. (2000). Stixbox: A Statistics Toolbox.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366.

Hull, J. H., Hak, L. J., Koch, G. G., Wargin, W. A., Chi, S. L., & Mattocks, A. M. (1981). Influence of range of renal function and liver disease on predictability of creatinine clearance. *Clinical Pharmacology and Therapeutics*, 29(4), 516-521.

Hwang, Y.-C., & Song, Q. (2008). Dynamic neural fuzzy inference system, *Proceedings of the 15th International Conference on Neuro- Information Processing of the Asia Pacific Neural Network Assembly (ICONIP 2008)*. Auckland, New Zealand: Springer.

Hwang, Y.-C., Song, Q., & Kasabov, N. (2008). MUFIS: A neuro-fuzzy inference system using multiple types of fuzzy rules, *Proceedings of the IEEE International Conference on Fuzzy Systems, 2008 (FUZZ-IEEE 2008)* (pp. 1411-1414).

Hwang, Y.-C., Song, Q., Kasabov, N., Greer, D., Goh, L., & Pang, S. (2009). NeuCom - A Neuro-computing Decision Support Enviroment (<http://www.theneucom.com>) (Version 0.919). Auckland: Knowledge Enginnering and Discovery Research Institute.

Islam, M. M., Xin, Y., Shahriar Nirjon, S. M., Islam, M. A., & Murase, K. (2008). Bagging and Boosting Negatively Correlated Neural Networks. *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, 38(3), 771-784.

Jamshidian, M., & Jennrich, R. I. (1997). Acceleration of the EM Algorithm by using Quasi-Newton Methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 59(3), 569-587.

Jang, J.-S. R. (1993). ANFIS: adaptive-network-based fuzzy inference system. *Systems, Man and Cybernetics, IEEE Transactions on*, 23(3), 665-685.

Jang, J.-S. R., & Sun, C.-T. (1995). Neuro-Fuzzy Modeling and Control. *Proceedings of the IEEE*, 83(3), 378-406.

Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*: Prentice Hall.

Japkowicz, N. (2000a). The class imbalance problem: Significance and strategies, *Proceedings of the Fourteenth Joint Conference on Artificial Intelligence* (pp. 111-117).

Japkowicz, N. (2000b). Learning from Imbalanced Data Sets: A Comparison of Various Strategies, *Proceedings of the AAAI'2000 Workshop on Learning from Imbalanced Data Sets* (pp. 10-15): AAAI Press.

Jelliffe, R. W. (1971). Estimation of creatinine clearance when urine cannot be collected. *Lancet*, 1, 975-976.

Jelliffe, R. W. (1973). Letter: Creatinine clearance: bedside estimate. *Annals of Internal Medicine*, 79(4), 604-605.

Joachims, T. (1999). Transductive inference for text classification using support vector machines, *Proceedings of the Sixteenth International Conference on Machine Learning* (pp. 200-209).

Joachims, T. (2003). Transductive Learning via Spectral Graph Partitioning, *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 290-297).

John, G., Kohavi, R., & Pfleger, K. (1994). *Irrelevant Features and the Subset Selection Problem*. Paper presented at the International Conference on Machine Learning.

Johnson, S. C. (1967). Hierarchical Clustering Schemes. *Psychometrika*, 2(241-254).

Kasabov, N. (1996). *Foundations of neural networks, fuzzy systems and knowledge engineering*. London: MIT Press.

Kasabov, N. (2001). Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning. *IEEE Trans. SMC - part B Cybernetics*, 31(6), 902-918.

Kasabov, N. (2007a). *Evolving Connectionist Systems: The Knowledge Engineering Approach* (2 ed.): Springer.

Kasabov, N. (2007b). Global, local and personalised modeling and pattern discovery in bioinformatics: An integrated approach. *Pattern Recognition Letters*, 28(6), 673-685.

Kasabov, N., & Hwang, Y.-C. (2008). *KEDRI's Local and Personalised modelling on the SCOPE PET clinical data*. Auckland: Knowledge Engineering and Discovery Research Institute.

Kasabov, N., & Pang, S. (2003). Transductive support vector machines and applications in bioinformatics for promoter recognition, *Proceedings of the International Conference on Neural Networks and Signal Processing, 2003* (Vol. 1, pp. 1-6).

Kasabov, N., & Song, Q. (2002). DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *Fuzzy Systems, IEEE Transactions on*, 10(2), 144-154.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks, 1995* (Vol. 4, pp. 1942-1948).

Kim, H.-C., Pang, S., Je, H.-M., Kim, D., & Bang, S.-Y. (2002). Support Vector Machine Ensemble with Bagging. In *Pattern Recognition with Support Vector Machines* (Vol. 2388/2002, pp. 131-141): Springer Berlin.

Kohavi, R. (1995a). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1137-1145). San Mateo, CA: Morgan Kaufmann.

Kohavi, R. (1995b). A study of cross-validation and bootstrap for accuracy estimation and model selection, *Proceedings of the International Joint Conference on Artificial Intelligence 1995* (pp. 1137-1143): Morgan Kaufmann.

Kohavi, R., & John, G. H. (1997). Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97(1-2), 273-324.

Koller, D., & Sahami, M. (1996). Toward Optimal Feature Selection, *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 284-292).

Koskela, T., Varsta, M., Heikkonen, J., & Kaski, K. (1998). Time series prediction using recurrent som with local linear models. *International Journal of Knowledge-based Intelligent Engineering Systems*, 2, 60-68.

Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets: one-sided selection, *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 179-186): Morgan Kaufmann.

Kurnik, R. T., Oliver, J. J., Waterhouse, S. R., Dunn, T., Jayalakshmi, Y., Lesho, M., et al. (1999). Application of the Mixtures of Experts algorithm for signal processing in a noninvasive glucose monitoring system. *Sensors and Actuators B: Chemical*, 60(1), 19-26.

Lachenbruch, P., & Mickey, A. (1968). Estimation of error rates in discriminant analysis. *Technometrics*, 10(1), 1-11.

Lei, Z., Yang, Y., & Wu, Z. (2006). Ensemble of Support Vector Machine for Text-Independent Speaker Recognition. *International Journal of Computer Science and Network Security*, 6(5), 163-167.

Leng, G., McGinnity, T. M., & Prasad, G. (2005). An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network. *Fuzzy Sets and Systems*, 150(2), 211-243.

Levey, A. S., Bosch, J. P., Lewis, J. B., Greene, T., Rogers, N., & Roth, D. (1999). A more accurate method to estimate glomerular filtration rate from serum creatinine: A new prediction equation. *Annals of Internal Medicine*, 130, 461-470.

Levey, A. S., Coresh, J., Greene, T., Marsh, J., Stevens, L. A., Kusek, J. W., et al. (2007). Expressing the Modification of Diet in Renal Disease Study equation for estimating glomerular filtration rate with standardized serum creatinine values. *Clinical Chemistry*, 53(4), 766-772.

Li, J., & Chua, C.-S. (2003). Transductive inference for color-based particle filter tracking, *Proceedings of the International Conference on Image Processing, 2003 (ICIP 2003)* (Vol. 3, pp. 949-952). Nanyang Technol. Univ., Singapore.

Lora, A. T., Santos, J. M. R., Exposito, A. G., Ramos, J. L. M., & Santos, J. C. R. (2007). Electricity Market Price Forecasting Based on Weighted Nearest Neighbors Techniques. *Power Systems, IEEE Transactions on*, 22(3), 1294-1301.

Lucks, M. B., & Oki, N. (1999). A radial basis function network (RBFN) for function approximation, *Proceedings of the 42nd Midwest Symposium on Circuits and Systems*, 1999 (Vol. 2, pp. 1099-1101).

Mackey, M. C., & Glass, L. (1977). Oscillation and chaos in physiological control systems. *Science*, 197, 287-289.

MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 281-297). Berkeley: University of California Press.

- Mamdani, E., & Assilian, S. (1975). An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7(1), 1-15.
- Marshall, M. R., Song, Q., Ma, T. M., MacDonell, S. G., & Kasabov, N. (2005). Evolving connectionist system versus algebraic formulas for prediction of renal function from serum creatinine. *Kidney International*, 67(5), 1944-1954.
- Martens, H. A., & Dardenne, P. (1998). Validation and verification of regression in small data sets. *Chemometrics and Intelligent Laboratory Systems*, 44(1-2), 99-121.
- Mawer, G. E., Lukas, S. B., Knowles, B. R., & Stirland, R. M. (1972). Computer-assisted prescribing of kanamycin for patients with renal insufficiency. *Lancet*, 1, 12-15.
- McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition*. New York: John Wiley & Sons, Inc.
- Minh Ha, N., Abbass, H. A., & McKay, R. I. (2008). Analysis of CCME: Coevolutionary Dynamics, Automatic Problem Decomposition, and Regularization. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(1), 100-109.
- Minsky, M., & Papert, S. (1969). *Perceptrons: an introduction to computational geometry*. MIT Press.
- Mitchell, M. T. (1997). Machine Learning. In: MacGraw-Hill.
- Okamoto, K., Ozawa, S., & Abe, S. (2003). A fast incremental learning algorithm of RBF networks with long-term memory, *Proceedings of the International Joint Conference on Neural Networks 2003* (Vol. 1, pp. 102-107).

Ozawa, S., Pang, S., & Kasabov, N. (2008). Incremental Learning of Chunk Data for Online Pattern Classification Systems. *Neural Networks, IEEE Transactions on*, 19(6), 1061-1074.

Pang, S. (2004). SVM classification tree algorithm with application to face membership authentication, *Proceedings of the IEEE International Joint Conference on Neural Networks, 2004* (Vol. 1, pp. 436).

Picard, R. R., & Cook, R. D. (1984). Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387), 575-583.

Poggio, F. (1994). Regularization theory, radial basis functions and networks. *From Statistics to Neural Networks: Theory and Pattern Recognition Applications, NATO ASI Series*, 83 - 104.

Preeclampsia, F. (2008). *About Preeclampsia*, from <http://www.preeclampsia.org/about.asp>

Qinbao, S., Martin, S., Xiangru, C., & Jun, L. (2008). Can k-NN imputation improve the performance of C4.5 with small software project data sets? A comparative evaluation. *J. Syst. Softw.*, 81(12), 2361-2370.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*: Morgan Kaufmann Publishers Inc.

Riedmiller, M., & Braun, H. (1992). RPROP- A fast adaptive learning algorithm, *Proceedings of the Seventh International Symposium on Computer and Information Sciences*. Ankara, Turkey.

Rodrigo, A. G., & Learn, G. H. (2000). *Computational and evolutionary analysis of HIV molecular sequences* (1 ed.): Springer.

Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain,. *Psychological Review*, 65(6), 386-408.

Ross, S. (2003). Peirce's Criterion for the Elimination of Suspect Experimental Data. *J. Engr. Technology*.

Ruey, S. T. (1988). Outliers, level shifts, and variance changes in time series. *Journal of Forecasting*, 7(1), 1-20.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 1, pp. 318-362). Cambridge: The MIT Press.

Schafer, J. L. (1999). Multiple imputation: a primer *Statistical Methods in Medical Research*, 8(1), 3-15.

Scheffer, J. (2002). Dealing with Missing Data. *Research Letters in the Information and Mathematical Sciences*, 3, 153-160.

Schliebs, S., Platel, M. D., & Kasabov, N. (2008). Integrated Feature and Parameter Optimization for an Evolving Spiking Neural Network. In M.Koeppen, N.Kasabov, G.Coghill & M.Ishikawa (Eds.), *Proceedings of the 15th International Conference on Neuro- Information Processing of the Asia Pacific Neural Network Assembly (ICONIP 2008)*. Auckland, NZ: Springer LNCS.

Scott, E. F., & Christian, L. (1990). The cascade-correlation learning architecture. In *Advances in neural information processing systems 2* (pp. 524-532): Morgan Kaufmann Publishers Inc.

Sharkey, A. J. C., Chandroth, G. O., & Sharkey, N. E. (2000). A Multi-Net System for the Fault Diagnosis of a Diesel Engine. *Neural Computing & Applications*, 9(2), 152-160.

Sheng-Sung, Y., Chia-Lu, H., & Chien-Min, L. (2006). HBP: improvement in BP algorithm for an adaptive MLP decision feedback equalizer. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 53(3), 240-244.

Shipp, M., Ross, K., Tamayo, P., Weng, A., Kutok, J., Aguiar, R., et al. (2002). Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8, 68 - 74.

Song, Q. (2001). *Evolving connectionist systems and applications for dynamic process modelling and control*. University of Otago, Dunedin.

Song, Q., & Kasabov, N. (2004). TWRBF - Transductive RBF Neural Network with Weighted Data Normalization. *Lecture Notes in Computer Science*, 3316, pp. 633-640.

Song, Q., & Kasabov, N. (2006). TWNFI -- a transductive neuro-fuzzy inference system with weighted data normalization for personalized modeling. *Neural Networks*, 19(10), 1591-1596.

Song, Q., Kasabov, N., Hwang, Y.-C., & Chrystall, B. (2006). *Fonterra/AUT Neural Network Prototype for Forecasting Volumes, Stage II Prototype Model Design*. Auckland: Knowledge Engineering and Discovery Research Institute.

Song, Q., Kasabov, N., Ma, T., & Marshall, M. R. (2006). Integrating regression formulas and kernel functions into locally adaptive knowledge-based neural networks: A case study on renal function evaluation. *Artificial Intelligence in Medicine*, 36(3), 235-244.

Song, Q., & Kasabov, N. K. (2005). NFI: a neuro-fuzzy inference method for transductive reasoning. *Fuzzy Systems, IEEE Transactions on*, 13(6), 799-808.

Song, Q., Ma, T. M., & Kasabov, N. (2006). TTLSC – Transductive Total Least Square Model for Classification and Its Application in Medicine, *Advanced Data Mining and Applications, Lecture Notes in Computer Science* (Vol. 4093/2006, pp. 197-204): Springer Berlin / Heidelberg.

Soucy, P., & Mineau, G. W. (2001). A simple KNN algorithm for text categorization, *Proceedings of the International Conference on Data Mining 2001 (ICDM 2001)* (pp. 647-648). San Jose, CA, USA.

Sugiyama, M., Okabe, Y., & Ogawa, H. (2004). On the Influence of Input Noise on a Generalization Error Estimator, *Proceedings of the Artificial Intelligence and Applications conference, 2004 (AIA 2004)* (pp. 218 - 223). Innsbruck, Austria.

Swets, J. A. (1988). Measuring the Accuracy of Diagnostic Systems. *Science*, 240, 1285-1293.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modelling and control. *IEEE Trans. Systems Man Cybernet*, 15, 116–132.

Tax, D., & Juszczak, P. (2002). Kernel Whitening for One-Class Classification. In *Pattern Recognition with Support Vector Machines* (pp. 855-873).

Tax, D. M. J., van Breukelen, M., Duin, R. P. W., & Josef, K. (2000). Combining multiple classifiers by averaging or by multiplying? *Pattern Recognition*, 33(9), 1475-1485.

Taylor, J. R. (1997). *An Introduction to Error Analysis*. California: University Science Books.

Tibshirani, T., Friedman, R., & Hastie, J. H. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*: Springer Verlag.

Tin Kam, H., Hull, J. J., & Srihari, S. N. (1994). Decision combination in multiple classifier systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(1), 66-75.

Trujillo-Ortiz, A., Hernandez-Walls, R., Castro-Perez, A., Rodriguez-Cardozo, L., Ramos-Delgado, N. A., & Garcia-Sanchez, R. (2004). Fisherextest:Fisher's Exact Probability Test.

Tucker, J., & William McGuire. (2005). ABC of preterm birth: Epidemiology of preterm birth. *Student BMJ*, 13, 133-176.

Übeyli, E. D. (2005). A Mixture of Experts Network Structure for Breast Cancer Diagnosis. *J. Med. Syst.*, 29(5), 569-579.

Ueffing, N., Haffari, G., & Sarkar, A. (2007). Transductive learning for statistical machine translation, *Proceedings of the Association for Computational Linguistics (ACL) Second Workshop on Statistical Machine Translation (WMT07)* (pp. 25-32). Prague, Czech Republic.

Vapnik, V. N. (1998). *Statistical Learning Theory*. New York: John Wiley and Sons.

Vernieuwe, H., Verhoest, N. E. C., De Baets, B., Hoeben, R., & De Troch, F. P. (2007). Cluster-based fuzzy models for groundwater flow in the unsaturated zone. *Advances in Water Resources*, 30(4), 701-714.

Walser, M. (1998). Assessing renal function from creatinine measurements in adults with chronic renal failure. *American Journal of Kidney Diseases*, 32(1), 23-31.

Walser, M., Drew, H. H., & Guldán, J. L. (1993). Prediction of glomerular filtration rate from serum creatinine concentration in advanced chronic renal failure. *Kidney Int*, 44(5), 1145-1148.

Wang, L.-X. (1994). *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*: Prentice Hall.

Watts, M. J. (2009). A Decade of Kasabov's Evolving Connectionist Systems: A Review. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 39(3), 253-269.

Weisstein, E. *Chi-Squared Test*. Retrieved 28 January, 2009, from <http://mathworld.wolfram.com/Chi-SquaredTest.html>

Wenhua, Z., & Jian, M. (2004). A novel incremental SVM learning algorithm, *Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design, 2004*. (Vol. 1, pp. 658-662).

Widiputra, H., Pears, R., Serguieva, A., & Kasabov, N. (2008). Dynamic Interaction Networks in Modelling and Predicting the Behaviour of Multiple Interactive Stock Markets. *Journal of Intelligent System in Accounting, Finance and Management*.

Woods, K., Kegelmeyer, W. P., Jr., & Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(4), 405-410.

Xin, Y., & Yong, L. (1998). Making use of population information in evolutionary artificial neural networks. *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, 28(3), 417-425.

Yamada, T., Yamashita, K., Ishii, N., & Iwata, K. (2006a). Text Classification by Combining Different Distance Functions with Weights, *Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006*. (pp. 85-90).

Yamada, T., Yamashita, K., Ishii, N., & Iwata, K. (2006b). *Text Classification by Combining Different Distance Functions with Weights*.

Yao, X., & Liu, Y. (1996). Ensemble structure of evolutionary artificial neural networks, *Proceedings of the IEEE International Conference on Evolutionary Computation, 1996* (pp. 659-664).

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8, 338-353.

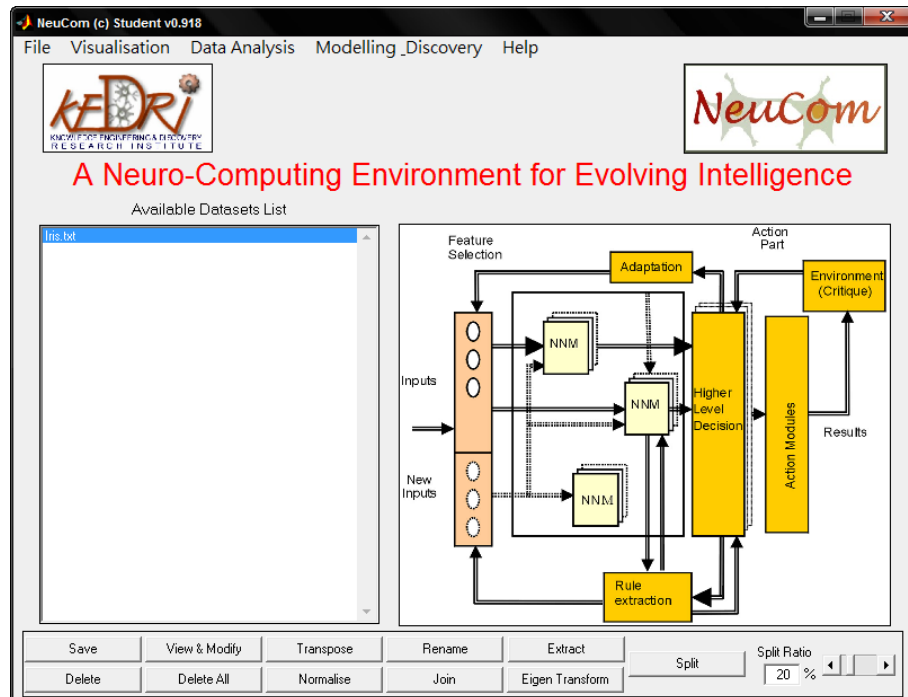
Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Systems Man Cybernet*, 28–44.

Zhou, Z.-H., & Jiang, Y. (2003). Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble. *Information Technology in Biomedicine, IEEE Transactions on*, 7(1), 37-42.

LIST OF PUBLICATIONS

1. Song, Q., Kasabov, N., Hwang, Y.-C., & Chrystall, B. (2006). Fonterra/AUT Neural Network Prototype for Forecasting Volumes, Stage II Prototype Model Design. Auckland: Knowledge Engineering and Discovery Research Institute. **(Technical report)**
2. Hwang, Y.-C., & Song, Q. (2008). Dynamic neural fuzzy inference system, Proc. ICONIP 2008. LNCS. Auckland, New Zealand: Springer. **(Conference Proceeding)**
3. Hwang, Y.-C., Song, Q., & Kasabov, N. (2008). MUFIS: A neuro-fuzzy inference system using multiple types of fuzzy rules, Proceedings of the IEEE International Conference on Fuzzy Systems, 2008 (FUZZ-IEEE 2008) (pp. 1411-1414). **(Conference Proceeding)**
4. Kasabov, N., & Hwang, Y.-C. (2008). KEDRI's Local and Personalised modelling on the SCOPE Preeclampsia prediction. Auckland: Knowledge Engineering and Discovery Research Institute. **(Technical Report)**
5. Hwang, Y.-C., Hwang, Y.-C., & Su, C.-H. (2008). Experience Co-Creation on Ubiquitous Cultural e-Service Provision: A Case of Taiwan's Hakka Culture, Fourth International Conference on Networked Computing and Advanced Information Management, 2008. NCM '08. Vol. 2, pp. 438-443. **(Conference Proceeding)**

APPENDIX A NEUCOM - A NEURO-COMPUTING DECISION SUPPORT ENVIRONMENT

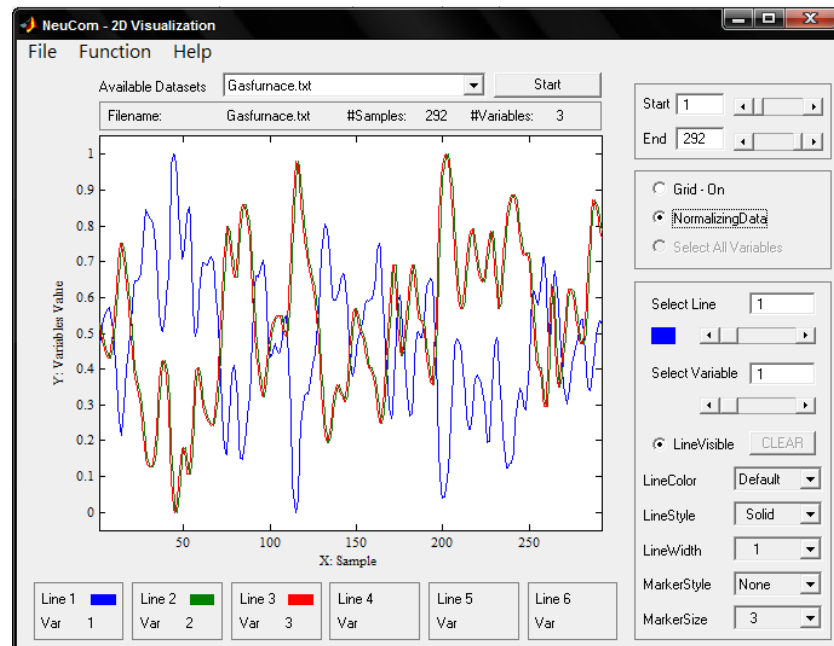


NeuCom is a generic data processing platform that can facilitate researchers in understanding their data (Hwang et al., 2009). This software is freely available at <http://www.theneucom.com> for non-commercial use.

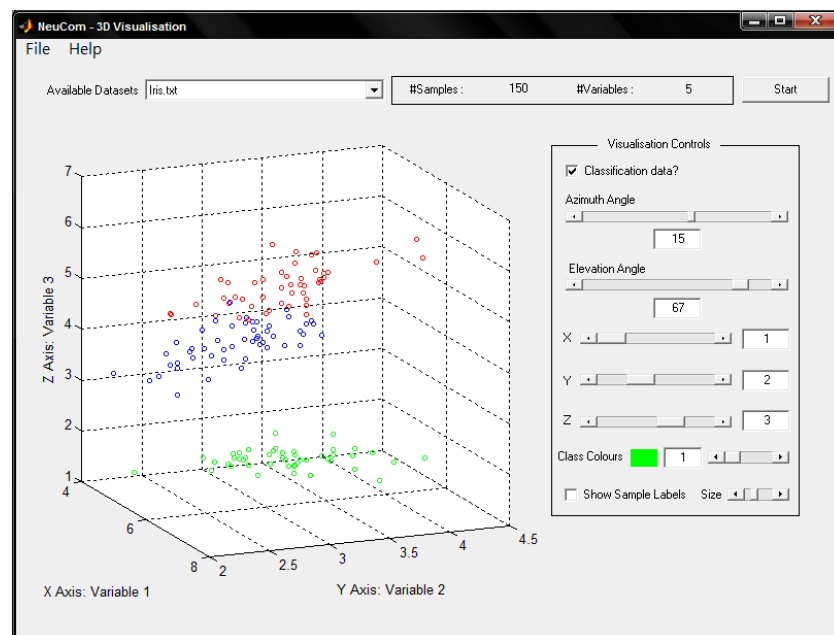
NeuCom was originally designed to facilitate the comparison between prediction methods based on Evolving Connectionist System principle and other prediction methods. It was later expanded into a full fledge data analysis and modelling software package.

NeuCom provides many useful tools and methods that are commonly used by researchers to perform the following tasks:

1. Data Visualisation
 - 2D Visualisation



- 3D Visualisation



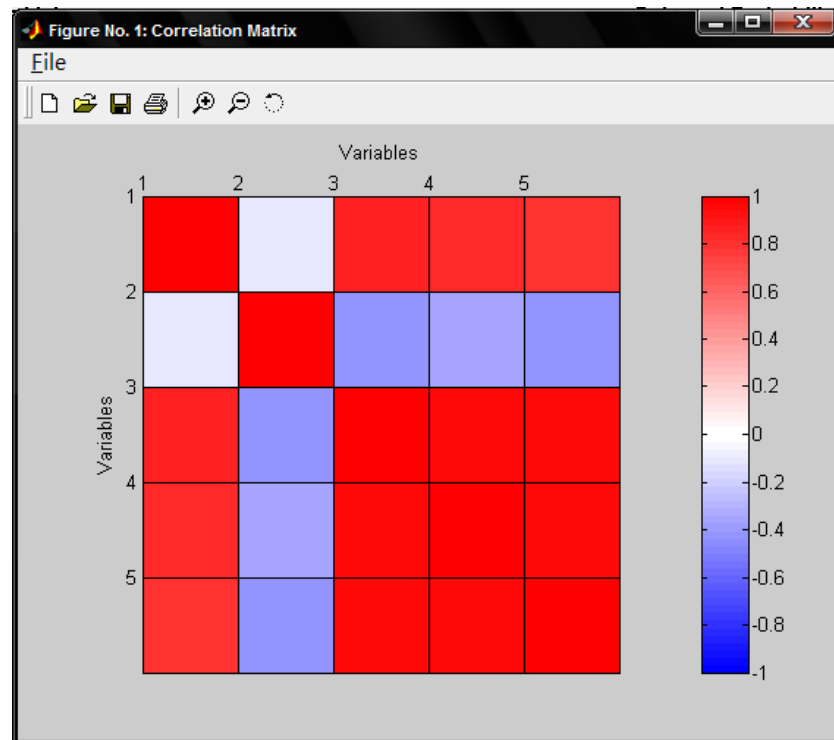
- Surface Plot

2. Data Transformation

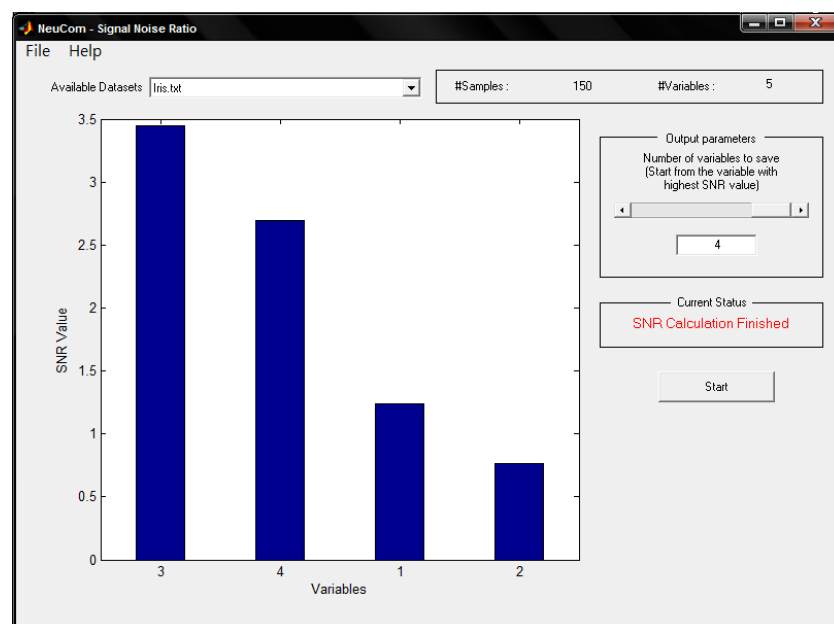
- Principle Component Analysis
- Linear Discriminant Analysis

3. Feature Selection and Analysis

- Correlation Coefficient Analysis

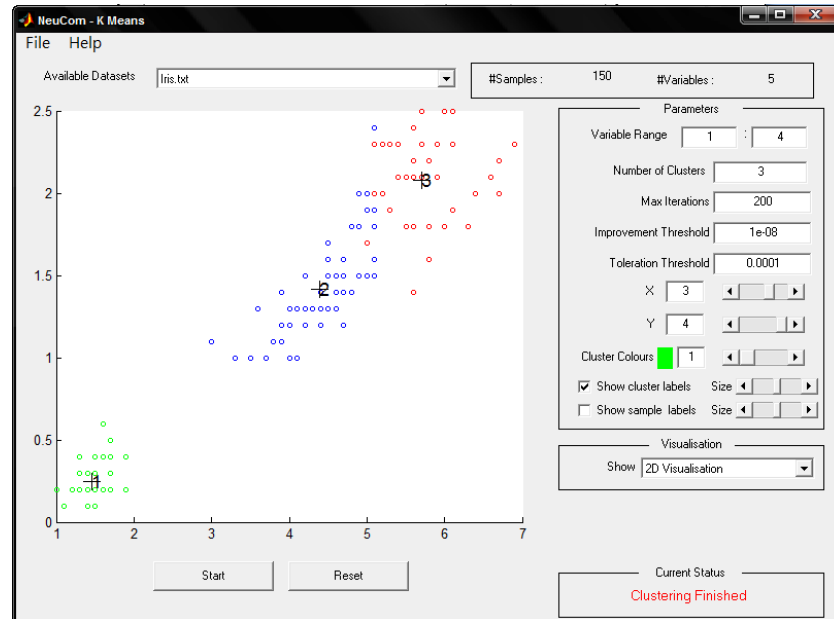


- Signal-To-Noise Ratio

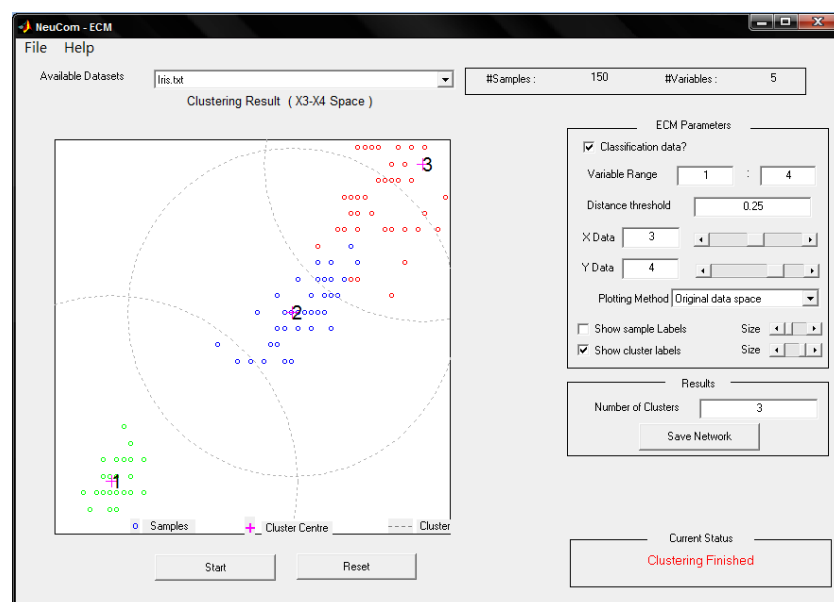


4. Clustering

- K-Means



- Evolving Clustering Method

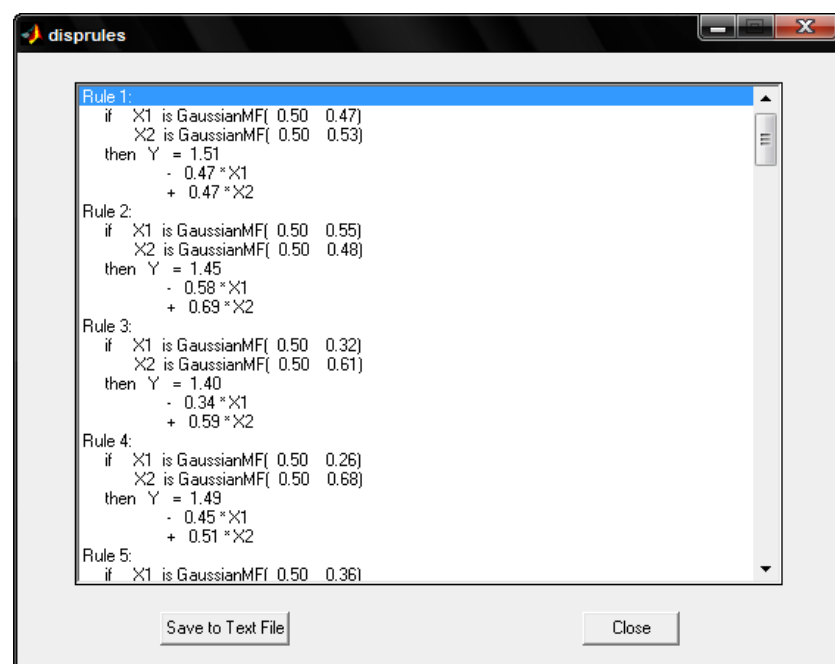
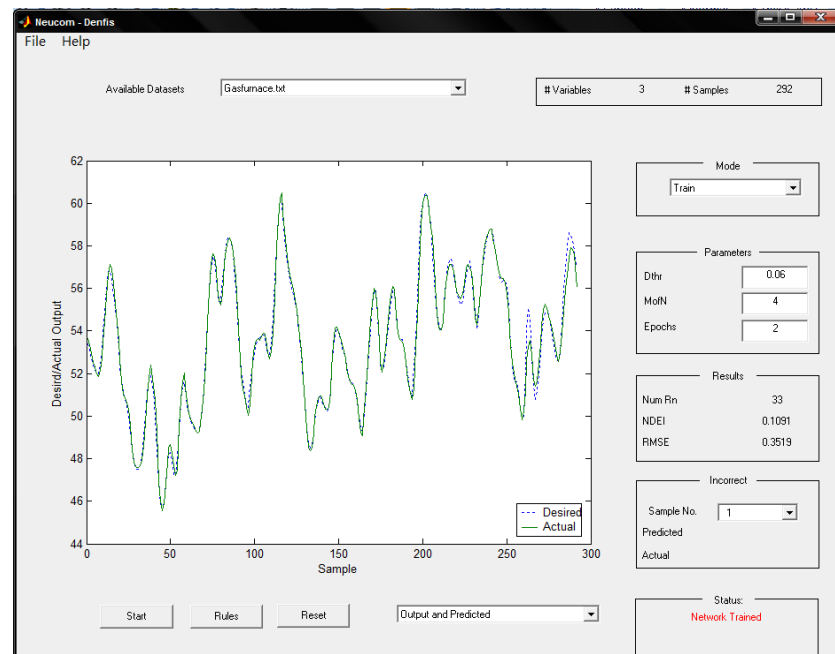


- Bi-Clustering

5. Modelling

- Linear Regression
- Support Vector Machine
- Radial Basis Function Network

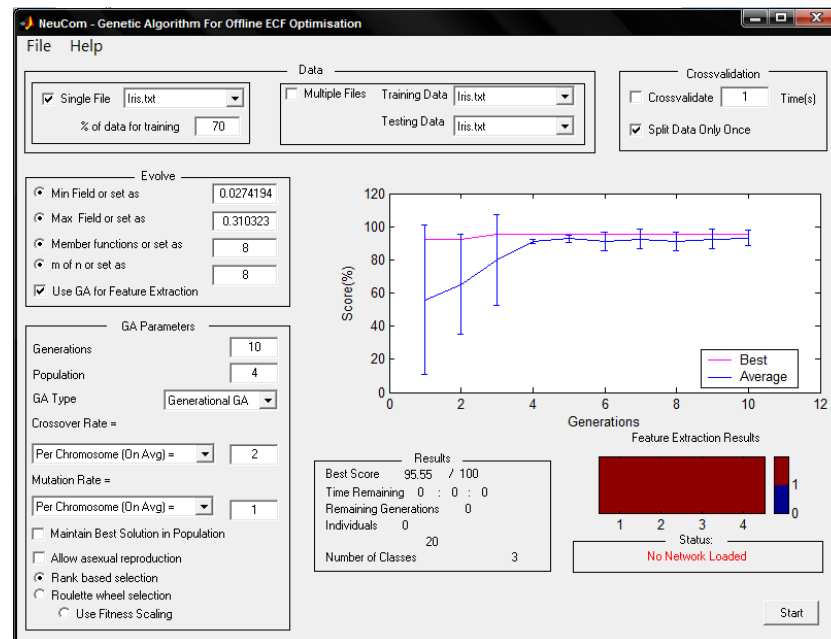
- Evolving Classification Function
- Evolving Clustering Method for Classification
- K-Nearest Neighbour
- Dynamic Evolving Neuro-Fuzzy Inference System



- Evolving Fuzzy Neural Network

6. Optimisation

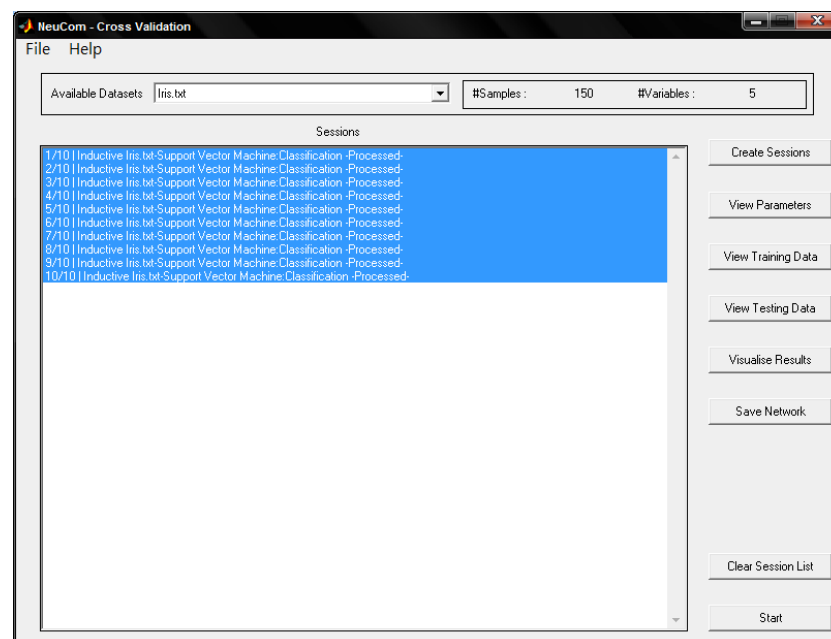
- Genetic Algorithm

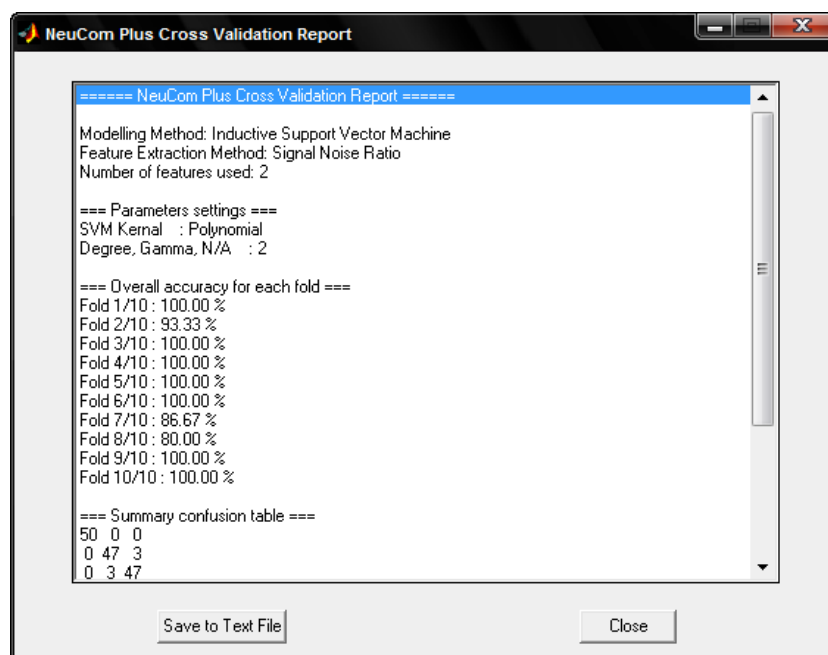


- Evolutionary Strategy

7. Cross Validation

- K-Fold Cross Validation with feature selection (Filter Method)





8. Other tools for data manipulation

- Split
- Transpose
- Normalise
- Eigen transform

APPENDIX B INTRODUCTION TO SCOPE STUDY

(from official scope study website <http://www.scopestudy.net>)

Introduction

Preeclampsia, fetal growth restriction (undernourished baby) and spontaneous preterm birth are the major complications of late pregnancy. They are leading causes of illness and death in mothers and newborn babies. In the developed world, in almost half the cases either the mother and/or baby require admission to an intensive care unit. Every year, an estimated \$41 billion is spent on healthcare costs related to these pregnancy diseases.

Pregnancy Problems

Preeclampsia is a severe high blood pressure condition where the mother can develop kidney or liver problems, stroke and seizures. It affects 5% of first time mothers. Each year, the number of maternal deaths from preeclampsia is equivalent to the loss of 170 jumbo jets of pregnant women. A quarter of the babies born to mothers with preeclampsia are growth restricted and a third are premature.

Fetal growth restriction is usually due to placental problems leading to inadequate nutrition of the baby and overall affects 1 in 10 pregnancies.

One of the greatest risks to a baby's health is the premature birth. Premature babies are 10-times less likely to survive. Two-thirds of all premature births are caused by **spontaneous premature labour**.

All three conditions can have lifelong consequences for the child. The child may have problems with brain development that can result in mild learning difficulties through to severe disabilities. Being born growth restricted predisposes the child to high blood pressure, heart attacks and diabetes as an

adult. The social consequences and lifelong economic costs resulting from these conditions are enormous. Prevention of these health problems is of paramount importance to future mothers, fathers and children.

Prenatal Care Today

Currently, there is no screening test that accurately predicts which first time mothers will develop these late pregnancy diseases. Prenatal care consists of a series of consultations during pregnancy with a doctor or midwife. One of the main reasons for these checks is to detect early signs of these pregnancy complications. Unfortunately, these problems often present suddenly themselves. The standard intervals between prenatal visits may result in delays in diagnosis with an increased chance of severe complications. If at risk, first time mothers were able to be identified in early pregnancy, known therapies could prevent almost a third of cases.

Predict to Prevent

Identification of first time mothers at risk for these conditions is the first step to effective intervention and prevention. Through SCOPE, we expect to develop an early pregnancy screening test that will offer first time mothers accurate risk assessment for each disease. The intensity of prenatal care could then be matched to each woman's personal risk profile and preventative therapies offered to those at high risk. The majority of women at very low risk could be reassured and medical intervention in their pregnancy care minimized.

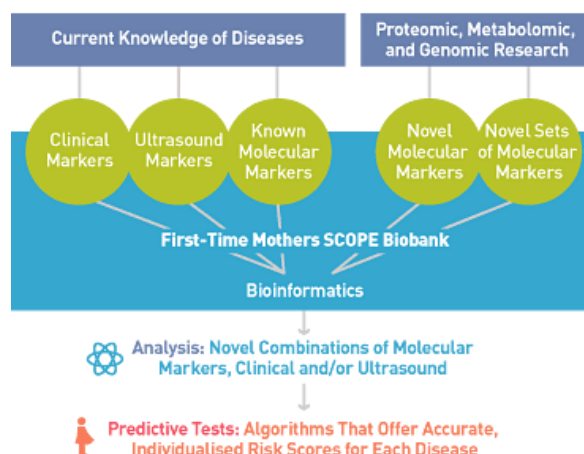
Aims

The SCOPE study is establishing a unique, international pregnancy biobank that will serve as a platform to:

- Identify novel molecular markers that predict early pregnancy women who will subsequently develop late pregnancy complications.
- Test and validate combinations of key clinical, known and novel molecular markers to predict each disease.
- Develop predictive tests that offer first time mothers an accurate, personalised risk rating for each disease.

Research

The SCOPE study arises from the knowledge that there are a number of potential clinical and molecular markers (certain proteins, fats and small molecules in blood) for these complications. None of these candidate markers are useful as individual predictive tests, but combinations of markers are likely to result in clinically useful screening tests. Further, recent advances in proteomic and metabolomic technologies and bioinformatics (advanced mathematics) allow us to discover and map differences in molecules circulating in the blood of women who later develop these conditions. This has created the opportunity to develop effective methods of predicting these diseases, with the potential to dramatically improve maternal and infant health worldwide.



Various types of data used to develop the predictive tests. (From

www.thescopestudy.net)

Clinical Dataset

The clinical data collected for SCOPE study contains the following topics:

- Demography History
- Maternal History
- Family History
- Current Pregnancy
- 15-week Clinical Examination
- 15-week Lifestyle Questionnaires
- Partner Data
- 20-week Clinical Examination
- 20-week Lifestyle Questionnaires
- Ultrasound
- Outcome

The questions covered in each topic are highly comprehensive and significant amount of work was done to minimise missing values to ensure high quality database. This comes to over 400 questions for each patient.