Aston University

If you have discovered material in AURA which is unlawful e.g. breaches copyright, (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please read our Takedown Policy and contact the service immediately

Uncertainty and Topographic Visualisations

An Application of Breast Cancer Prognosis

MINGMANAS SIVARAKSA

Doctor of Philosophy

ASTON UNIVERSITY

December 2008

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without proper acknowledgement.

Aston University

Uncertainty and Topographic Visualisations

An Application of Breast Cancer Prognosis

Mingmanas Sivaraksa

Doctor of Philosophy, 2008

Thesis Summary

This thesis is a study of low-dimensional visualisation methods for data visualisation under uncertainty of the input data. It focuses on the two main feed-forward neural network algorithms which are NeuroScale and Generative Topographic Mapping (GTM) by trying to make both algorithms able to accommodate the uncertainty.

The two models are shown not to work well under high levels of noise within the data and need to be modified. The modification of both models, NeuroScale and GTM, are verified by using synthetic data to show their ability to accommodate the noise.

The thesis is interested in the controversy surrounding the non-uniqueness of predictive gene lists (PGL) of predicting prognosis outcome of breast cancer patients as available in DNA microarray experiments. Many of these studies have ignored the uncertainty issue resulting in random correlations of sparse model selection in high dimensional spaces. The visualisation techniques are used to confirm that the patients involved in such medical studies are *intrinsically unclassifiable* on the basis of provided PGL evidence. This additional category of 'unclassifiable' should be accommodated within medical decision support systems if serious errors and unnecessary adjuvant therapy are to be avoided.

Keywords: Topographic Visualisation, NeuroScale, Generative Topographic Mapping, Uncertainty, Breast Cancer Prognosis, Gene Expressions, Microarray, Kullback-Leibler distance

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Prof. David Lowe, for his patience and understanding. David has been a very good supervisor by giving me very useful advice, suggestions and also encouragement. I am very grateful that he could provide some of his busy schedule for discussion on my research. Also to Dr. Davide D'Alimonte who helped me on my first year of my research and gave many useful guidance.

Importantly, I would like to express my gratitude to Biopattern for funding my research work. Thank to Randa Herzalla who helped with some parts of committee classifiers and to all members of the NCRG group, especially Prof. Ian Nabney, Dr. Dan Cornford for useful discussions and to Vicky Bond for dealing with all administrative jobs. Also to all my friend in the group who have been helping me with all technical problems. Special thank to Rajeswari Matam and Thomas Bermudez for helping me while I was having my health problems.

I would like to thank to all my Thai friends, especially Kontaros Kaomuagnoi, for nice Thai food and moral support during my time here in Birmingham. Finally, I would like to thank my family for their support and encouragement.

Declaration

I declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research. Parts of this work have been appeared as:

- M. Sivaraksa and D. Lowe. Predictive gene lists for breast cancer prognosis: A topographic visualisation study. BMC medical genomics, 1(1):8, 2008.[chapter 3.]
- M. Sivaraksa and D. Lowe. Unclassifiability in medical prognosis: example using biopattern gene markers. In Third International Conference on Computational Intelligence in Medicine and Healthcare, Plymouth, UK, 2007.[Part of chapter 6.]
- M. Sivaraksa and D. Lowe. Gene expression predictors of breast outcome: A visualisation study. pages 53-56, Tampere Finland, June 2006.[Part of chapter 2.]
- M. Sivaraksa and D. Lowe. Microarray Visualisation Using NeuroScale Models Poster session presented at: European Summer School in Biomedical Informatics., Balatonfured, Hungary, July 2006. [Part of chapter 5.]
- D. D'Alimonte, D. Lowe, I. T. Nabney, and M. Sivaraksa. Visualising uncertain data. In 2nd European Conference on Emergent Aspects in Clinical Data Analysis, EACDA05, Pisa, Italy, 2005. [Part of chapter 4 and 5.]

Contents

1 Introduction **13**
- 1.1 Context . . . 13
  - 1.1.1 Topographic Visualisation . . . 14
  - 1.1.2 Systems Biology . . . 15
  - 1.1.3 Gene Expression . . . 16
  - 1.1.4 Microarray Technology . . . 16
- 1.2 Uncertainty . . . 20
  - 1.2.1 Types of uncertainty . . . 21
  - 1.2.2 Related techniques for uncertainty modelling in microarray data 22
  - 1.2.3 Importance of uncertainty . . . 23
- 1.3 Model uncertainty . . . 24
  - 1.3.1 Bayesian . . . 24
  - 1.3.2 Bootstrap . . . 25
  - 1.3.3 Network Ensemble . . . 25
  - 1.3.4 Predictive Error Bar . . . 26
- 1.4 Biomarkers from Gene Expression Profiles . . . 28
  - 1.4.1 The van't Veer data set . . . 29
  - 1.4.2 Why data visualisation helps . . . 31
- 1.5 Plan of the thesis . . . 32

2 Standard Visualisation Techniques **34**
- 2.1 NeuroScale . . . 35
- 2.2 Isometric Mapping . . . 36
- 2.3 Locally Linear Embedding . . . 38
- 2.4 Probabilistic PCA . . . 40
- 2.5 Generative Topographic Mapping . . . 42
- 2.6 Stochastic Neighbour Embedding . . . 44
- 2.7 Parametric Embedding . . . 46
- 2.8 Other techniques . . . 47
  - 2.8.1 Computational Complexity . . . 47
- 2.9 Discussion . . . 48

3 Prognosis Gene List and Patient Visualisation **53**
- 3.1 Introduction . . . 53
  - 3.1.1 The van't Veer data problem . . . 53
  - 3.1.2 The previous study of non uniqueness gene list . . . 55
- 3.2 Experiments on the van't Veer data . . . 56

3.2.1 Classifier Comparison . . . 56
3.2.2 NeuroScale Projection . . . 57
3.2.3 Locally Linear Embedding . . . 60
3.2.4 Generative Topographic Mapping . . . 64
3.2.5 Stochastic Neighbour Embedding . . . 68
3.3 Discussion . . . 71
3.3.1 Comparison across models . . . 71
3.3.2 Comparison of PGLs across patient groups . . . 73
3.4 The validation set . . . 73
3.5 Conclusion . . . 80

4 Modified GTM **82**
4.1 Introduction . . . 82
4.2 The Modified GTM Model . . . 82
4.3 Illustrations of modified GTM . . . 85
4.3.1 Data-dependent Variance . . . 88
4.3.2 Comparisons of the synthetic data results . . . 94
4.3.3 The real data sample with attached uncertainty . . . 95
4.3.4 The Comparison between the Standard GTM and the Modified GTM . . . 98
4.4 Conclusion . . . 102

5 Probabilistic Approaches to NeuroScale **103**
5.1 Heuristic Approaches . . . 103
5.1.1 Additional input noise . . . 104
5.1.2 Modified cost function . . . 104
5.2 The Probabilistic Approach . . . 105
5.2.1 Background . . . 105
5.2.2 Uncertainty weighted dissimilarity . . . 108
5.2.3 The probabilistic approach . . . 109
5.2.4 Modification of the Shadow Target Algorithm . . . 111
5.3 Illustrations of Modified NeuroScale . . . 113
5.3.1 Synthetic Example . . . 113
5.3.2 Novel data projection . . . 115
5.3.3 Synthetic Example of a Curved Manifold . . . 119
5.3.4 Synthetic Example on a 2-dimensional plane with non-uniform added noise . . . 121
5.3.5 The real data sample with attached uncertainty . . . 124
5.4 Conclusion . . . 130

6 Cancer Prognosis Case Study **131**
6.1 Uncertainty Measures . . . 131
6.1.1 Predictive Error Bar . . . 132
6.1.2 The committee network . . . 134
6.1.3 Prognosis Indicators with uncertainty . . . 135
6.2 Modified NeuroScale applied to the van't Veer data set . . . 135
6.2.1 Single predictive error bar . . . 136

6.2.2 Modified NeuroScale results . . . 136
6.2.3 Committee predictive error bar results . . . 140
6.2.4 Modified NeuroScale results using the van't Veer data set and committee-predictive uncertainty. . . . 142
6.3 Modified GTM applied to the van't Veer data set . . . 144
6.3.1 Single predictive error bar . . . 144
6.3.2 Modified GTM results using the single predictive error bar . . . 144
6.3.3 Committee predictive error bar results . . . 146
6.3.4 The modified GTM results using the van't Veer data set and committee-predictive uncertainty . . . 146
6.4 Projection of New Patients . . . 150
6.4.1 The projection of the new patients using modified NeuroScale . 150
6.4.2 The projection of the new patients by the modified GTM . . . 154
6.5 Comparison between the probabilistic NeuroScale and the probabilistic GTM . . . 155
6.6 Conclusion . . . 157

7 Conclusion **158**
7.1 Visualisation for Biomedical Data . . . 158
7.2 Probabilistic Visualisation . . . 159
7.3 Intrinsic Uncertainty . . . 161
7.4 Directions of Future Research . . . 162

A The Two Gene Lists **171**
A.1 A List of the original 70 gene list, List A. . . . 171
A.2 A List of the alternative 70 gene list, List B. . . . 171

B Classification Results of the standard models **172**
B.1 The misclassification matrix from the NeuroScale projection using List A 172
B.2 The misclassification matrix from the NeuroScale projection using List A with only high confidence patients . . . 172
B.3 The misclassification matrix from the NeuroScale projection using List B 173
B.4 The misclassification matrix from the NeuroScale projection using List B with only high confidence patients . . . 173
B.5 The misclassification matrix from the LLE projection using List A with $K = 5$ . . . 173
B.6 The misclassification matrix from the LLE projection using List A with $K = 5$ using only high confidence patients . . . 173
B.7 The misclassification matrix from the LLE projection using List B with $K = 5$ . . . 174
B.8 The misclassification matrix from the LLE projection using List B with $K = 5$ using only high confidence patients . . . 174
B.9 The misclassification matrix from the LLE projection using List A with $K = 20$ . . . 174
B.10 The misclassification matrix from the LLE projection using List A with $K = 20$ using only high confidence patients . . . 175

B.11 The misclassification matrix from the LLE projection using List B with $K = 20$ 175
B.12 The misclassification matrix from the LLE projection using List B with $K = 20$ using only high confidence patients 175
B.13 The misclassification matrix from the GTM projection using List A . . 175
B.14 The misclassification matrix from the GTM projection using List A using only high confidence patients 176
B.15 The misclassification matrix from the GTM projection using List B . . 176
B.16 The misclassification matrix from the GTM projection using List B using only high confidence patients 176
B.17 The misclassification matrix from the SNE projection using List A with $\sigma = \log(5)$ 177
B.18 The misclassification matrix from the SNE projection using List A with $\sigma = \log(5)$ using only high confidence patients 177
B.19 The misclassification matrix from the SNE projection using List B with $\sigma = \log(5)$ 177
B.20 The misclassification matrix from the SNE projection using List B with $\sigma = \log(5)$ using only high confidence patients 177
B.21 The misclassification matrix from the SNE projection using List A with $\sigma = \log(20)$ 178
B.22 The misclassification matrix from the SNE projection using List A with $\sigma = \log(20)$ using only high confidence patients 178
B.23 The misclassification matrix from the SNE projection using List B with $\sigma = \log(20)$ 178
B.24 The misclassification matrix from the SNE projection using List B with $\sigma = \log(20)$ using only high confidence patients 179
B.25 The misclassification matrix from the NeuroScale projection on the new 234 patients using List A 179
B.26 The misclassification matrix from the NeuroScale projection on the new 234 patients using List A with only high confidence patients 179
B.27 The misclassification matrix from the NeuroScale projection on the new 234 patients using List B 180
B.28 The misclassification matrix from the NeuroScale projection on the new 234 patients using List B with only high confidence patients 180
B.29 The misclassification matrix from the GTM projection on the new 234 patients using List A 180
B.30 The misclassification matrix from the GTM projection on the new 234 patients using List A with only high confidence patients 181
B.31 The misclassification matrix from the GTM projection on the new 234 patients using List B 181
B.32 The misclassification matrix from the GTM projection on the new 234 patients using List B with only high confidence patients 181

C Traditional Visualisation Approaches **182**
C.1 Principal Component Analysis 182
C.2 Multidimensional Scaling 184
C.2.1 Classical multidimensional scaling 184

C.2.2 Non-metric multidimensional scaling 185
C.2.3 MDS Example 186

D Standard Shadow Targets 190

List of Figures

1.1 The process of data visualisation. 14
1.2 The manufacturing process of microarray Data. 18
1.3 A microarray chip with errors in some spots. 19
1.4 The simulation of an RBF approximation of $y = \sin(x)$ with added noise ϵ. 24

2.1 The NeuroScale architecture. 36
2.2 The comparison between the geodesic distance and the Euclidean distance. 37
2.3 The Locally Linear Embedding algorithm. 39
2.4 The GTM architecture. 42
2.5 The example of using LLE algorithm to recover the structure of S-curve. 49
2.6 The barbel example where local methods fail to preserve the original data 51

3.1 The standard NeuroScale projections of Lists A (a) and B (b). 59
3.2 The LLE results with $K = 5$ using PGL A and B. 61
3.3 The LLE results with $K = 20$ using PGL A and B. 62
3.4 The standard GTM results using PGL A and B. 66
3.5 The standard GTM results using PGL A and B with latent shape of 56. 67
3.6 The SNE results with $\sigma = log(5)$ patients using PGL A and B. 69
3.7 The SNE results with $\sigma = log(20)$ patients using PGL A and B. 70
3.8 The NeuroScale visualisation projection of the new 234 patients. 75
3.9 The GTM visualisation projection of the new 234 patients. 77
3.10 The LLE visualisation projection of the new 234 patients. 79

4.1 The visualisation of the 3 Gaussian clusters using modified GTM with different variances. 86
4.2 The comparative visualisation of the 3 Gaussian clusters between modified GTM and standard GTM with different variances. 87
4.3 The visualisation using modified GTM which shows the poor scaling of σ_n^2. 88
4.4 The modified GTM results of a synthetic example with different K values. 89
4.5 The resulting visualisation of the standard GTM of another synthetic data. 91
4.6 The modified GTM result of another synthetic data with 16 latent points and RBF centres 92
4.7 The modified GTM result of another synthetic data with 64 latent points and 64 RBF centres 93
4.8 The resulting visualisation using modified GTM with fixed K=0.1. . . . 96

4.9 The microarray visualisation of the microarray data set using modified GTM and estimating K to find the appropriate K to suit $\sigma^2 = K\sigma^{*2}$. The value of K is estimated by using the maximum log likelihood. . . . 97
4.10 The resulting visualisation using the standard GTM 97
4.11 The resulting visualisation of the standard GTM model, focusing on a grey circle region that we will use to investigate. 98
4.12 Genes 'SCO0271', 'SCO5208', 'SCO4897' and 'SCO4260' are labelled in the standard GTM visualisation. 98
4.13 The expressions of the selected genes 99
4.14 The resulting visualisations of the modified GTM of those genes that previously grouped together in the standard model. 100
4.15 The gene expression profiles of gene 'SCO4897' and its neighbours in the modified GTM. 100
4.16 The gene expression profiles of gene 'SCO5208' and its neighbours in the modified GTM. 101
4.17 The gene expression profiles of gene 'SCO4260' and its neighbours in the modified GTM. 101
4.18 The gene expression profiles of gene 'SCO0271' and its neighbours in the modified GTM. 101

5.1 The standard NeuroScale projection of a synthetic example generated from three different Gaussian centres. 114
5.2 The modified input matrix of the standard NeuroScale projection of the synthetic example. 114
5.3 Different probabilistic NeuroScale projections of the synthetic example. 116
5.4 Different NeuroScale projections of novel data projection in the synthetic example. 118
5.5 Another original synthetic example in a curve manifold. 120
5.6 Another synthetic projection of different NeuroScale approaches with non-uniform noise variance. 122
5.7 The resulting visualisation using modified NeuroScale. 125
5.8 The resulting visualisation using standard NeuroScale. 126
5.9 The resulting visualisation of the standard NeuroScale model, focusing on the blue circle region that we will use to investigate. 127
5.10 The expressions of the selected genes. 127
5.11 The resulting visualisations of modified NeuroScale of those genes that previously grouped together in the standard model. 128
5.12 The gene expression profiles of gene 'SCO5777' and its neighbours in the modified NeuroScale. 128
5.13 The gene expression profiles of gene 'SCO5775' and its neighbours in the modified NeuroScale. 129
5.14 The gene expression profiles of gene 'SCO4449' and its neighbours in the modified NeuroScale. 129
5.15 The gene expression profiles of gene 'SCO5487' and its neighbours in the modified NeuroScale. 129

6.1 The diagram of methodology of training the predictive error bars for the standard projection. 132
6.2 The predictive output of the classifier from the standard NeuroScale projection. 137
6.3 The Modified NeuroScale results using single network uncertainty. . . . 139
6.4 The predictive output of the committee classifier from the standard NeuroScale projection. 140
6.5 The modified NeuroScale using the single predictive error bar. 143
6.6 The predictive outputs of the classifier from the standard GTM projection. 145
6.7 The modified GTM visualisation using the single-model predictive error bar. 147
6.8 The predictive outputs of the classifier from the standard GTM projection. 148
6.9 The modified GTM visualisation using the committee average predictive error bar. 149
6.10 The modified NeuroScale projection of the new patients trained in Figure 6.3. 151
6.11 The modified NeuroScale projection of the new patients trained in Figure 6.5 153
6.12 The modified GTM projection of the new patients trained in Figure Figure 6.7. 154
6.13 The modified GTM projection of the new patients trained in Figure Figure 6.9. 156

C.1 A reconstructed map of British cities using classical MDS. 187
C.2 A possible reconstructed map of British cities using Non metric MDS. . 188

List of Tables

3.1 The misclassification results of all models. 71
3.2 Misclassification results of GTM and NeuroScale of the new patient set. 76

4.1 A classification accuracy table of the synthetic examples. 94
4.2 A table of means and variances of the error of different GTM approaches. 95

5.1 A table of averaged STRESS from different level of variances with different NeuroScale approaches. 123
5.2 A table of median average of the STRESS measure using different NeuroScale approaches. 124

C.1 A table of distances between various cities with units in miles. 187
C.2 A table of reconstructed distances using Classical MDS between various cities with units in miles. 188
C.3 A table of reconstructed distances using non-matric MDS between various cities. 189

Chapter 1

Introduction

1.1 Context

This thesis is about finding the best way to represent data in a simpler form for interpretation by aiming to preserve the 'topology' of the data. The representation of data in visual projections in a form which allows people to understand the data easier [84] can be referred to as "visualisation". One single data sample can consist of many entities describing that particular sample, leading to the high dimensional nature of data. It is often difficult to investigate all the entities to understand the structure of the data in the original space. 'Data visualisation' is an essential tool for aiding the users to understand and making further data analysis and evaluation. For human visual perception, the projection from the high dimensional space which holds all the data entities to the reduced dimensional form facilitates the understanding of the data and assists the interpretation by using the brain's ability to assimilate patterns in data. Clearly we can not perceive visualised images in more than three dimensions and can understand better two-dimensional representations [59].

Figure 1.1 shows the process of data visualisation from the given information. The first stage is where mathematical models are required to *project* or *transform* from more complicated data to a data format which is easier to represent using image representation techniques in the second stage. Many people refer to data visualisation as just one of the two stages of the whole process of data visualisation. To differentiate between two stages, it is common to refer to this first stage as 'data projection' [84] or

'embedding techniques' [36]. This thesis will mainly focus on this first stage of data visualisation. The transformed features obtained from this stage can then be used easily with any graphic representation techniques in the second stage.

Many existing techniques ignore the fact that real-world data is usually noisy and imprecise. Moreover, when mathematical model involved in the data modeling also increase the uncertainty. This type of uncertainty is called model uncertainty which is also very important. Ignoring this information can alter the final projective representation and lead to incorrect understanding and assumptions of the data. This thesis uses examples from systems biology in which uncertainty is quite common but very sensitive. Projecting the data down to low dimensions without caution can affect the results and lead to incorrect interpretation, which is very dangerous especially if it is part of a life-critical application [69, 95]. The thesis uses the representations of genomic microarray data as a real-world example because of its high levels of noise in the data, its very high dimensional nature, and the recent importance of DNA microarrays for the extraction of biomarkers for cancer, for example [90].

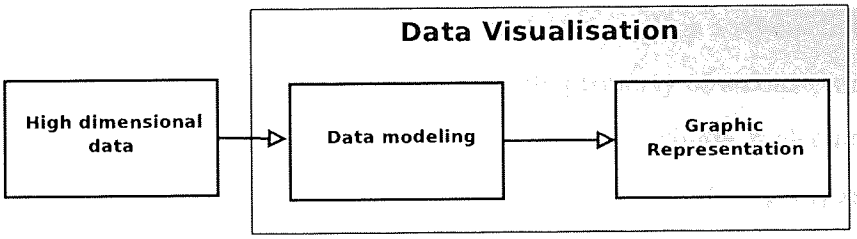


Figure 1.1: The process of data visualisation.

1.1.1 Topographic Visualisation

Tipping [84] mentioned that in this information age where large amounts of information can be easily obtained, using all the available information makes understanding and interpretation difficult. Topographic visualisation is a data projection method which yields the *preservation* of the structure of the data between the data space and the reduced dimensional space. A topographic projection preserves the geometric structures of the data by mainly retaining the relative similarities between data in the high and low dimensional spaces. Therefore, the visualised image of the data in the two or three

dimensional representation is very helpful. Alternatively, the topographic method may only consider the maintenance of neighbourhood relationships as being important, such that points that originally lie close together are likewise preserved in the map, and this is referred to as *topological ordering*.

Nevertheless, information obtained from real-world data is usually noisy and will eventually distort the resulting projections of the data that hence do not represent the true structure of the data. Even though many studies have been focusing on the noise in supervised classification and regression, not enough attention has been paid to uncertainty in unsupervised methods, such as data visualisation [15, 77, 78]. The thesis will investigate this problem motivated by a real-world example in medical *systems biology* where data have high levels of uncertainty but high accuracy of data interpretation is also crucial.

1.1.2 Systems Biology

Systems biology is the study of functional properties of living organisms by looking at a big system of many interactions between small constituents. It is a recent field in molecular biology which traditionally looks at only a single property of a constituent without understanding the interactions between them. Studying a single molecular property independently does not lead to understanding about the true living organisms since within their biological systems, there are interactions and dynamics involved which creates dependency between constituents. The developments of molecular biology include producing new high throughput techniques and analysis of functional behaviour of the interactions of multiple constituents within cells [10]. Therefore mechanisms for helping scientists to observe and understand the system are required. The technology exists to allow scientists to observe everything that happens in the system, nevertheless, in order to completely understand a system the theoretical background and thorough observations are also required [10]. Systems biology helps interpret the biological development by not only understanding the interactions and dynamics in the system but also by trying to build the mathematical model to 'fit' the observed data and with the aim eventually to predict future outcomes. Clearly, large systems such as in systems biology, require a comprehensive study for qualitative and accurate interpretation of

systems. Reliable visualisation tools help the biologists to understand the structures and make correct decisions. One large problem in systems biology is *microarray technology* which has a wide range of benefits in both medical and biological domains which helps identify genetic structure and pathways of diseases [10].

1.1.3 Gene Expression

Gene expression is the process by which a gene transcribes from DNA into RNA. The transcription process is a process of making RNA from one strand of the DNA molecule. Then RNA gained from the transcription leaves the nucleus and is translated to the required protein [48]. Any gene which is active in this way at a particular time is said to be expressed. It is clear that gene expression plays the key role in regulating all the functions. This gene expression information is important because the role of a gene is determined by the protein it produces. The improper expression of genes can be detected compared to the standard one. Furthermore, different environmental conditions can cause different expressions of genes. This adaptability to the environment of genes can cause disease if genes express in an improper manner. It is important to know in which environment genes create unwanted behaviour or, perhaps, in which environment genes behave in a proper manner in order to develop the biological environment for each gene. The technology that can investigate multiple genes simultaneously is microarray technology, which is described in the next section.

1.1.4 Microarray Technology

The microarray is a powerful recent technology [99, 39] that allows the investigation of the 'gene expression' level of thousands of genes at the same time. The reason for its popularity is that microarrays changed the way researchers work. Instead of working on a gene-by-gene basis, scientists can study large numbers of genes at once [48]. Therefore, improper gene expression, which can cause genetic health problems can be identified more easily. One of the most common diseases that uses this technology is cancer [96]. Nevertheless, with microarray technology, rather than giving better accuracy in the data, much higher error rates are produced than traditional methods [10]. Most publications on microarray experiments do not discuss the implications

of uncertainty in the data and consequences for the results.

The microarray is physically a slide where genes under investigation are spotted in a defined position of a regular grid [13, 39, 64]. The microarray slide can be made by nylon meshes, silicon, or nitrocellulose. There are many applications of microarrays but one common way that will also be used in this thesis is to use the relative intensities of the cell in two different conditions which are the sample cells, such as cancer, and normal cells in order to investigate genes that have different expressions compared to the other conditions. The procedure of extracting the intensities from two cells is described below.

The two cells need to be identified differently. RNA is extracted from the two previously prepared cells and labelled with two different fluorescent dyes. The red-fluorescent dye or Cy5 is for the diseased cell and the green-fluorescent dye or Cy3 is for the normal or control cell. Both of them are combined to a pre-arranged DNA microarray [49]. If the sequence is complementary to the DNA spot on the array, known as a probe, those RNAs will hybridise to that spot. The RNA which can find the complementary pair is measured by being excited by two different types of laser for different dyes, red for Cy5 and green for Cy3. The intensity of the emitted light from each dye is measured by a detector which records its intensity. The scanned data are then transformed into two digital images of the array. The colours which are used to identify in this image are similar to the previous laser colours, red for Cy5 and green for Cy3. This final image can used as an RGB image to represent the levels of intensity of the spots. Figure 1.2 summarises all the microarray manufacturing processes as described above.

In the real-world of the microarray experiments, errors can be produced at any stage of the experiments. For example, the intensity of light can be recorded from a different spot during the scanning process. Moreover, the intensity in some spots may be high because of extra light contributing to the background of noise of the scanned array image. In addition the microarray chips always contain distorted spots or spots with irregular shapes similar to Figure 1.3. Some spots combine together to make a single spot and the spots do not have definite, fixed or expected positions. As a result, processing the images to get a correct estimation of the true expression values

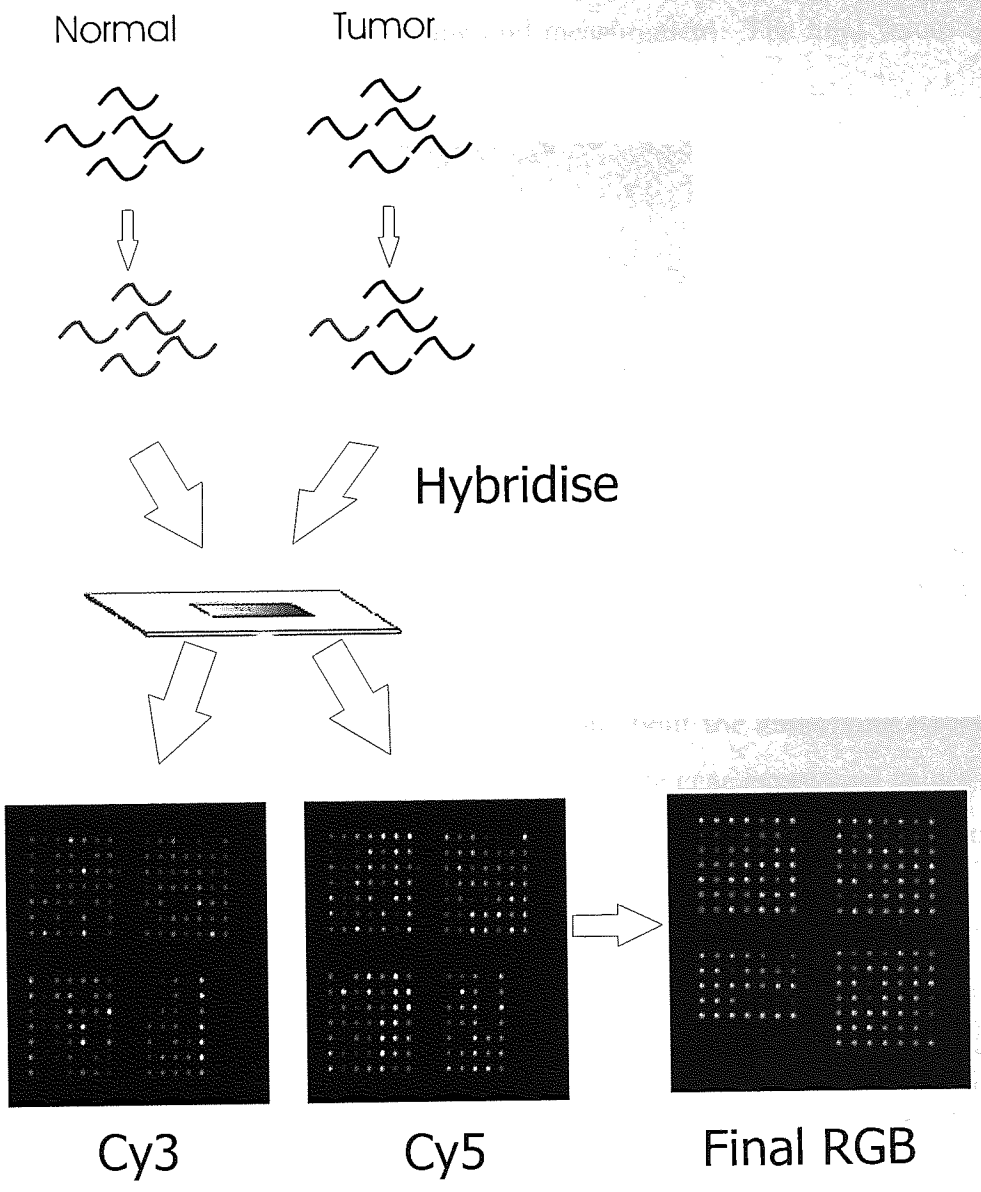


Figure 1.2: The manufacturing process of microarray Data. Firstly, cells from two different conditions are chosen and then labelled differently by using different colours. Next, hybridise to the prepared chip. Then, use two different colour lasers to activate and measure the intensities from different spots and create two grey scale images. Finally, put both images in different colours and superimpose to create the final RGB image.

which is measured from an intensity at a wavelength becomes extremely hard. Many normalisation techniques can help reduce some errors within the microarray chips, however, using normalisation techniques alone can not remove all types of error that occur during the process of manufacturing and measurement. The final values always include some errors and some missing spots.

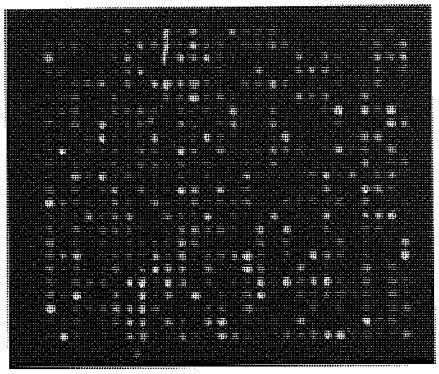


Figure 1.3: A microarray chip with errors in some spots. Some spots are too big which may interfere the results of the neighbouring spots, such as the one in the top left corner of this chip.

In order to make a more appropriate assumption about the underlying structure of the data, uncertainty measures should be attached to the gene expression values before any further data processing. Using data visualisation with uncertain data information can give incorrect data structures, therefore it can mislead the interpretation of the projected microarray data.

Errors in Microarray Data

There are many different sources of variation in microarray data experiments. Errors in microarray can be as simple as created by human, such as swapping dye from different channel or replica. Since microarray process involves many biological process from extracting mRNA in which the control condition is very sensitive to get the correct results on the gene chip. The production of the microarray can also create errors, mainly due to machine error such as quality of printed probe. The hybridisation and scanning process can also created variation in the final results of the microarray experiments [99].

Some sources of error can be reduced by using data normalisation. By first, us-

ing spatial normalisation to reduce the effects during the hybridisation which occurred when the chip was not washed evenly. Then, background correction will be done. This correction will reduce the effects of any signal which is measured as optical intensities which is produced in the background and irrespective of the true signal. Further more, dye-effect corrections will be performed on the microarray measurement. This normalisation will reduce the different properties in two dyes of the dual-channel array [99]. Finally within replicate and across-conditions normalisation will be done. Within replicate normalisation refers to the normalisation of the data in the same slide. For this normalisation, define $M = \log(Cy3/Cy5)$ and $A = \frac{\log(Cy3*Cy5)}{2}$. The 2-dimension plot of M against A is produced. This normalisation shifts the tend line of this plot, produced by value M down to zero along with the values of A, under the assumption that most genes have not been differentially expressed. Across-conditions normalisation is to make the mean or the median of the different arrays in different conditions into the fixed quantity. This process assumes that the average intensity from all the different conditions used in the normalisation are similar.

1.2 Uncertainty

Data quality is a very important topic in data assessment [91], especially in a medical domain where small errors can cause incorrect diagnosis resulting in dangerous outcomes. It is essential to know whether the data used is of good or poor quality. Expressing an estimate of the quality in terms of the *uncertainty* of the data is meaningful and should not be neglected [65]. The scientific community is focused mainly on the development and invention of new instruments to reduce errors in measurements. Representing the quality of the data in terms of the uncertainty of the data is, therefore, the right direction towards reducing error and increasing the degree of belief to make the result more reliable. However, it is not always possible to minimise or nullify all the errors. In some cases a small amount of error is acceptable. If the erroneous results are used to make further assumptions or analysis, it can eventually lead to incorrect conclusions. Acknowledging the nature of errors and investigating the errors by using a knowledge of the uncertainty, and by taking into account the range of the errors, an

appropriate assessment of data can eventually be obtained from such experiments.

Uncertainty is the concept of 'a lack of precise information of a certain value or event' [32]. We can express this extra uncertainty information on the resulting value obtained from the scientific experiments. Most of the time scientists can give a degree of belief to the resulting value from a certain experiment as additional information.

Of all the frameworks used to model uncertainty, the most principled and developed is probability theory. At a trivial level, estimates of variance (σ^2) can be used to represent the likely value, or error bars to bound the values in a certain region by limiting the upper and lower bound of the data can also be obtained. One approach for qualifying the quality of the microarray data is from the BlueFuse software [1] which provides a direct numerical 'score' for 'confidence' however the detailed explanation of this approach is not very clear. It tries to estimate the uncertainty level by analysing the final image of the gene chips. Alternatively, if the data uncertainty in not explicitly available and the model uncertainty can be estimated, this uncertainty should also be used.

1.2.1 Types of uncertainty

Two main types of uncertainty are usually associated with data analysis. The intrinsic noise derived from the data themselves and the model uncertainty. Noise can be added to the data either at the input from sensor noise or measurement noise or at a later stage after the transformation which is called predicted target noise, ϵ_n. Target noise is illustrated in Figure 1.4. The target noise is the noise that affects the model of the system resulting in the input data predicting the output inaccurately. The observed target can be expressed as:

$$t_n = f(x_n, W) + \epsilon_n \tag{1.1}$$

where the target variable t is assumed to be given by a deterministic function $f(x_n, W)$, x_n is the observed data and W is a parameter of the model class used with additive noise. Using the noisy target can result in poor model accuracy when trying to reconstruct the function from the input data. This noise may be reduced by having more targets from the same input, x_n. In addition to the target, noise also can affect the input. The intrinsic uncertainty of the data propagates through the model used to

describe the functional relationships in the data.

The model uncertainty, σ_w^2 is affected by unknown model characterisation such as the different training of different final configuration parameters. The intrinsic uncertainty of the data, expressed by σ_ν^2. The common assumption of the total uncertainty is a combination of uncertainties $\sigma_t^2 = \sigma_w^2 + \sigma_\nu^2$. There are many different approaches for estimating the errors of models. Many techniques associated with this problem include classical techniques such as, Bayesian techniques [8] and Bootstrap [32, 23], or neural-network related, such as Network Ensembles [63] or using predictive error bars [52].

1.2.2 Related techniques for uncertainty modelling in microarray data

In microarray data analysis, there are concerns over uncertainty issues in the data. One of the traditional methods for expressing uncertainty in microarray data is using the standard error from different repeat measurements from different experiments. However, more recently researchers have proposed new ways of attaching uncertainty to the microarray data [1, 67, 3, 17] by using Bayesian statistical techniques from only a single replicate. BlueGnome's technology platform, BlueFuse [1], is one of the most widely used software environments for microarray data measurement which introduces a 'confidence' level to attach to the final gene expression results.

Bluefuse automatically generates information from microarray images. Moreover, it calculates the signal intensities of dual channels, fluorescently labelled, microarray images based on image processing. The operation is fully automated; saving time, removing human subjectivity and raising experimental repeatability. BlueFuse uses statistical modelling techniques to "learn" from the array data and so does not require skilled human operators during the process as in earlier techniques [1]. This software combines mathematical models and uses Bayesian methods to extract more justifiable biological measurements from microarray experiments. Precise details of what quantity is calculated and how, are not publicly available. Some information can be obtained from the BlueGnome website [1]. In the development of the BlueFuse software, a great deal of 'prior knowledge' about the process used to generate microarray data was

utilised. In practice this prior knowledge varies from a high degree of confidence, such as detector performance, to that which is less predictable, such as washing effects. The power of Bayesian systems lies in their ability to rigorously combine all grades of prior knowledge with the data to deliver improved results [29]. BlueFuse uses this variability of results to create a single 'Confidence Estimate' for each log-ratio of observed dual channel expression values. The observed dual channels are normally between the test and controlled channels. This estimate lies between 0 and 1. A Confidence Estimate close to 1 indicates a high level of confidence in the log-ratio calculation [1] while a confidence estimate close to 0 indicates a low level of confidence in the log-ratio calculation. This Bayesian technique helps to improve the traditional approaches which only set a threshold to divide the results between those that are either 'certainly present' or 'certainly absent' [1]. We intend to use this extra knowledge as part of this thesis work.

Another interesting method is using the PUMA (Propagating Uncertainty in Microarray Analysis) method which uses within-sample testing as an important source of uncertainty estimation [67]. Similar to the Bluefuse software, PUMA uses Bayesian inference to estimate the propagation of uncertainty from the probe-level by estimating the model using a gamma distribution of the positive probe intensities and the posterior distribution of expression levels. It also suggests that all stages of uncertainty analysis of different levels should be combined to one single probabilistic model. Current visualisation methods are unable to take into account the influence of uncertainty measurements. Part of this thesis will investigate the consequences of incorporating this extra information of uncertainty in expression values in data visualisation methods.

1.2.3 Importance of uncertainty

Ignoring uncertainty information means all available data have the same level of assumed quality. Mathematical models estimated from such data will treat every data point equivalently. A single outlier can alter the overall structure. A simple example is given in Figure 1.4 which is the simulation of the noisy sine function $t = \sin(x) + \epsilon$ with one outlier added to the point 70. The noisy data t is plotted in blue dots. The underlying function $\sin(x)$ is plotted in black and the model approximated by using

a Radial Basis Function neural network is plotted in a blue line. The result shows that one single outlier point can distort the model in the surrounding region. As a result any new data points that fall in that region will be corrupted. Hence it is very important to take into account input/output uncertainty.

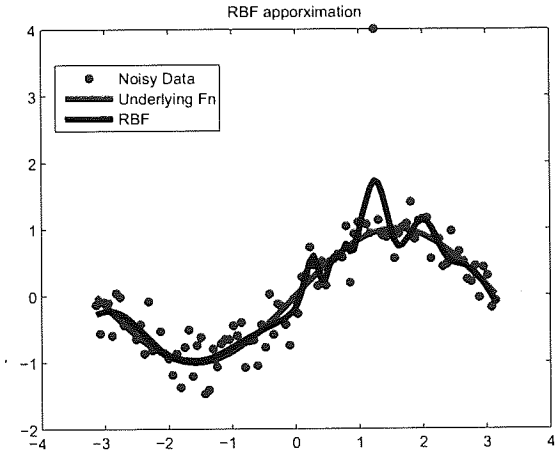


Figure 1.4: The simulation of RBF approximation of $y = \sin(x)$ with added noise ϵ. The function approximated by the RBF is plotted in a blue line.

This example illustrates that when estimates are made without taking into account the obvious outlier, a part of the model can be distorted. Hence incorporating this uncertainty information for model estimation is crucial.

1.3 Model uncertainty

In different data modelling approaches, such as classification and regression models, incorporating of uncertainty has been considered extensively. Many techniques associated with this problem include classical techniques such as, Bayesian techniques [8] and Bootstrap [32, 23], or neural network related, such as Network Ensembles [63] or using predictive error bars [52].

1.3.1 Bayesian

The first approach is using Bayesian approximation by integrating all the parameters to obtain suitable probability of each value [8].

Noise can affect the data from the input state. Further the target noise for regression problems can disturb the data after the transformation(σ_t^2), and prior information of the input data (σ_I^2).

$$\sigma_y^2 = \sigma_t^2 + \sigma_W^2 + \sigma_I^2, \tag{1.2}$$

where σ_I^2 is the variance of the input data. The first term can be derived by the Bayes rule. The results are given by

$$\sigma_t^2 = \frac{1}{\beta} + G^T A^{-1} G. \tag{1.3}$$

The second term is from the width of the posterior distribution of the model uncertainty, σ_W^2 and the last term can be approximated by

$$\sigma_I^2 = h^T \Sigma_i h, \tag{1.4}$$

where $h = \partial y(x, W))/\partial x$ is the input gradient of the neural network output measured at the noisy input data, and Σ_i is full covariance of the input noise which is assumed to be known.

1.3.2 Bootstrap

The bootstrap method is a method for estimating the standard error of statistical parameters. This technique uses different set of samples in each training set and the output of the network is obtained by

$$\hat{y}(x) = \frac{1}{M} \sum_{l=1}^{M} y(x, W_l). \tag{1.5}$$

The confidence interval can be estimated by

$$\sigma_W^2(y) = \frac{1}{M-1} \sum_{l=1}^{M} [y(x, W_l) - \hat{y}(x)]^2. \tag{1.6}$$

1.3.3 Network Ensemble

The bootstrap method is similar to using different neural networks which is known as the network ensembles [63].

The general ensemble function is

$$f_{GEM} = \sum_{i=1}^{M} \alpha_i f_i = f(x) + \sum_{i=1}^{M} \alpha_i m_i, \tag{1.7}$$

which m_i is the *misfit* function that is the deviation of the output data of each model i , y_i, to the target, $m_i = f(x) - y_i(x)$.

The combined output of the network is

$$y_{COMM} = \sum_{i=1}^{M} \alpha_i y_i. \tag{1.8}$$

The model coefficient, α_i satisfies the constraint $\sum_{i=1}^{N} \alpha_i$. The mean square error is the sum of the combination of the error from different networks,

$$MSE = \sum_{i,j} \alpha_i \alpha_j C_{ij}, \tag{1.9}$$

where $C_{ij} = E[m_i, m_j]$ is a correlation between two models. If each models is assumed independent then α_i given by

$$\alpha_i = \frac{\sigma^{-2}}{\sum_{l=1}^{M} \sigma_l^{-2}}, \tag{1.10}$$

where $\sigma_i^2 = C_{ii}$.

From the committee the outcome for each patient can be determined by (1.8). To gain uncertainty information for the new data, error estimation is required. The σ_i^2 estimation from the above method is derived literally from the error compared to the target vector.

$$\sigma_i^2 = \langle \|\overline{y} - \overline{t}\| | M_i \rangle. \tag{1.11}$$

However, this method does not provide the predictive property for the new data. But, this can be used with the predictive error bar methods.

1.3.4 Predictive Error Bar

Another approach is to use another neural network layer for predictive error bars. In [52], they use another layer of RBF networks with the same hidden unit to estimate another set of error bars. The first network estimates the optimum output which is $\langle t(x)|x \rangle$ which gives the local variance of $\|t(x) - \langle t(x)|x \rangle\|^2$. This value can be used for training another network for the uncertainty prediction. The optimum output of

this network is an approximation of variance for a given input, which can be used for estimating the uncertainty $\sigma^2(x) = \langle \|t(x) - \langle t(x)|x\rangle\|^2|x\rangle$. Both networks share the same input, the same first layer and hidden units of the neural networks but the output layers are separated with different sets of parameters. This model will be discussed in more detail in Chapter 6.

The optimisation process contains two stages. The first stage is to optimise the weights $W1$ for the traditional output of the network given the target values for regression. In the second stage, the network is the same but another neural network is attached to the hidden unit of the previous network (the first stage network). Weights for this network are optimised to achieve the error prediction trained, by using the variance of the previous output of the network. This method is obviously faster and does not need any assumptions of Gaussian distributions as in the simple Bayesian approach. If the uncertainty is assumed to be Gaussian, the likelihood can also be written down. However, any error can be used for the network training to obtain the error bars for the new network input.

Similarly, [62] used this same method but using multilayer perceptrons instead of RBF networks and obtained the variance by using the maximum likelihood approach which gives results similar to the Bayesian approach:

$$\sigma_t^2 = \sum_{l=1}^{N} [t(\mathbf{x}_l) - y(\mathbf{x}_l, w)]^2, \tag{1.12}$$

where N is the number of training examples and $t(x)$ is the desired target. The output $y(x, w)$ is the output of the first network.

Besides the low processing time of this method, the second stage neural network for predicting error bars can be used with new data efficiently. However, the disadvantage is that the extra layer may induce an extra weight uncertainty.

Similarly, for the visualisation model, the uncertainty of the data will induce model uncertainty and creates a poor projection of not only the data with large noise but also the surrounding data. However, for unsupervised techniques, such as visualisation, uncertainty has not been looked at so extensively. Only a few studies have been conducted, for example in the missing data problems, which can be regarded as data with very low uncertainty level in GTM model [9, 55]. This will be discussed further in chapter 4. In this thesis, the effect of incorporating uncertainty information which

may be available from data measurements, or estimated from imprecise models will be studied on visualisation techniques

1.4 Biomarkers from Gene Expression Profiles

In order to extract biomarkers from gene expression profiles, there are many different techniques using machine learning approaches. The simplest method is by detecting biomarkers using supervised classification. This method identyfies gene *signature* which most associate with the disease outcome. In this method, it is important that some validation set is needed to be set aside to test the performance of chosen biomarkers. To avoid overfitting of the biomarker extraction which is strongly dependent on the specific supervised training set available, an unsupervised technique provides another approach. This method aims to search for genes with similar structures of gene expression across a number of samples. This method is aiming to search for genes with similar structure of gene expression across number of samples. It is believed that genes with similar function will create the same outcome to the patients [87]. Data visualisation is regarded as an unsupervised data investigation. Other recent unsupervised data investigation of microarray includes, the Self Organising Map (SOM) to investigate yeast [81] and human cancers [30] and [92], the latter in combination with the k-means algorithm. Analogously, Principal Component Analysis (PCA) has been used to investigate yeast [46] and to identify tissue-specific expression of human genes [57].

Furthermore, many publications have suggested using expression of multiple genes related to the life cycle of breast cancer patients [21, 93, 90, 80], lung cancer patients [5], combination of cancer patients [66] and of patients with other genetic diseases [20, 19].

One of the high profile examples of a biological study into breast cancer markers from high dimensional microarray data is that of the research team of Laura van't Veer [90]

This study was motivated by personalised treatment for breast cancer patients. Breast cancer is the second most common cancer (10.6% of all cancer patients) after lung cancer and is the most common in women [100]. It is also the disease that carries significant psychological impact for women [79]. Traditional biomarkers used

for prognosis include tumour size, lymph node status and estrogen receptors. But these fail to correctly predict the outcome and prognosis for individual patients who need a tailored treatment plan to undergo cytolytic therapy, such as chemotherapy, which has unpleasant side effects as it can affect not only the cancer cell but also the normal surrounding cells [4]. The conventional treatment region allows 70-80% of cancer patients to unnecessarily undergo these therapies [26, 90, 89]. They would likely survive without treatment. Many publications are trying to improve this problem by using alternative methods. Gene variation gives a more thorough understanding into each individual patient. Not all patients with the same symptoms require the same treatment, and it depends also on their genetic profile which impacts on how the body processes each medication [4, 64]. Personalised medicine is a treatment paradigm where one size does not fit all [94] which overcomes the traditional approach in which the average treatment is advised for all patients [27, 64]. The emergence of using gene expressions for tailoring the treatments for each individual patient's need has become more popular in cancer research [61].

The approach that is now becoming more common is to detect breast cancer susceptible genes such as BRCA1 and BRCA2 which are high risk germline mutations for breast cancer [34]. Although successful as indicators, they are only found in at most 5 percent of breast cancer patients [88].

In addition, there is also a study using a gene expression profile together with traditional clinical information such as lymph node status [37] as biomarkers. However, none of these publications take into account any uncertainty measure. The aim of these gene set studies is to emphasise personalised care for breast cancer patients by using 'informative' genes to predict the development of distant metastasis in the patients.

1.4.1 The van't Veer data set

This thesis will focus mainly on the data obtained from the van't Veer study. The van't Veer study suggests a new methodology for predicting which breast cancer patients are prone to developing distant metastases, which means the cancer spreads beyond the breast area to other distant part of the body. From this stage, the rate of mortality in breast cancer patients is very high. Therefore preventing patients reaching this

stage means preventing the death in the patients. The adjuvent therapies, such as chemotherapy, helps impede the tumours to develop. However, not all sporadic tumours will develop into this stage. Current pathology diagnosis, such as family historical background, tumour sizes, lymph node status, fails to correctly classify the patients who need or who need not undergo these treatments. The van't Veer study suggested that using only gene expression profiles from patients, the predictability outperforms those clinical parameters that are currently used.

The van't Veer data set uses mainly 78 patients: 34 of whom developed distant metastasis within 5 years, which will be called poor prognosis patients. The remaining 44 patients are disease-free after a period of at least 5 year, which will be called good prognosis patients. All of them are lymph node negative and aged under 55 years old. RNAs are extracted from each patient and used for deriving complementary RNA(cRNA). The reference cRNA was made by extracting similar amounts of 5μg from each of the sporadic carcinomas. The hybridisations for each tumour were carried out by using a fluorescent dye reversal technique on microarrays containing approximately 25,000 human genes synthesised by inkjet technology. Only 5,000 genes were significant that is giving at least a twofold difference and P-value of less than 0.01 in more than five tumours. In the original paper, some of the histopathological data, such as oestrogen receptor(α) were presented for comparison [90].

In the van't Veer study, significant genes are then extracted by calculating the correlation coefficients which give the significant outcome of the disease from 5,000 genes to 231 genes. Furthermore, to find the optimum genes for predicting metastasis, this paper suggests building a "prognosis classifier" by using a subset of 5 genes from the top-ranked of the correlation coefficient magnitude trained by using the leave-one-out cross-validation. A subset of 5 genes from the top-ranked list are then added up recursively until all the 231 genes were used. The number of genes that optimally gives the highest accuracy is selected. Seventy genes were found to be the most optimum informative genes for predicting the distant metastasis.

During the selection of informative genes for prediction, a huge reduction in gene numbers are made. In the van't Veer data set, there is an approximate 95% reduction of the number of genes from the significantly expressed genes to significantly correlated

genes, from about 5,000 to 231 genes, under the assumption that all gene expressions are true values without any distortion or occurrence of uncertainty in the data. Obviously if any values are imprecise with some errors in the experiments, the resulting set of genes will very likely be different. In addition using a different training set for gene extraction will result in a different subset of genes as discussed in [24, 56]. However, high levels of noise in the microarray data may induce random correlation while trying to select features. Using a small subset selection of genes from very high dimensional gene samples can easily have random correlation between genes, especially if the data set has a very small sample size, such as 78 patients in the van't Veer study. Attaching uncertainty levels to gene expression data is important and will help to avoid misleading information [95] in order to achieve higher overall accuracy, with reduced risk to the patient. The most relevant uncertainty information for this data set comes from the small sample patients that used for investigation in the study, not the data uncertainty of the selected genes.

1.4.2 Why data visualisation helps

Dimensionality reduction techniques are required for visualising microarray data. Visual representation of the biological data can help in making a faster inspection and revealing structure hidden in complex gene expression data [59]. The dendrogram is one of the traditional approaches to perform microarray data clustering. Some studies extend the use of a traditional dendrogram for easier use [33]. However it usually produces a suboptimal local clustering solution and is not effective as a spatial visualisation tool to reflect relative dissimilarities. Many other algorithms for reduced dimensionality representation have previously been used to visualise microarray data. For instance, VistaClara, an interactive visualization tool [42], the Self Organising Map (SOM) has been used to investigate yeast [81] and human cancers [30] and [92], the latter in combination with the k-means algorithm. Analogously, Principal Component Analysis (PCA) has been used to investigate yeast [46] and to identify tissue-specific expression of human genes [57]. However, both SOM and PCA have significant drawbacks. PCA is a variance-preserving *linear projection*, and this limitation does not lead to a topographic representation [101]. On the other hand, the SOM lacks a sound the-

oretical underpinning (for example, there is no cost function to optimise, and training parameters must be chosen arbitrarily).

Many studies investigate the van't Veer data set in terms of classification rate by building many different supervised classifiers and presenting cooperative performance figures by revealing different rates of classification [90]. This thesis alternatively uses an unsupervised visualisation approach to investigate this study. The fully unsupervised visualisation models allow the data to reveal the true structure of all given samples without any interference of the predefined class labels that depend largely on the samples chosen for training. One could argue that constructing supervised models, such as a classifier, may reveal a better understanding of the data by differentiating data into different groups making interpretation easier. However, this is a misleading argument since the limited number of samples and random correlations in the van't Veer data prevents reliable discrimination as we will discuss later in the thesis. Classifiers trained on these data samples will not be reliable. The results will be sensitive to the chosen classification models and chosen data samples for training. Therefore this thesis will look at an alternative approach using projective data mappings, or, data visualisation. Data visualisation with attached uncertainty will be investigated and explored. The technique therefore will be applied on real data. The extra novelty also explored in this thesis is the incorporation and influence of uncertainty (from models as well as data when available) on the projective models.

1.5 Plan of the thesis

Chapter 2: is a comparative analysis of the established techniques of visualisation which includes both deterministic and probabilistic approaches and suggested ways of improvement toward uncertainty and visualisation.

Chapter 3: demonstrates the use of existing visualisation techniques on the example of the cancer data, the van't Veer data set. This chapter compares and discusses different approaches of visualisation techniques on one data sample. The problem of the uncertainty in the van't Veer data set is being discussed in more details.

Chapter 4: suggests an improvement of the Generative Topographic Mapping

method to incorporate the uncertainty level. Synthetic examples are used for evaluations.

Chapter 5: suggests the extension of NeuroScale models with a probabilistic approach to be able to incorporate uncertainty. This chapter also discusses probabilistic distance measurement and improvement of both heuristic and fully probabilistic approaches toward the standard NeuroScale model.

Chapter 6: uses the van't Veer data set as a case study for the modified GTM and NeuroScale. This chapter suggests an approach to evaluate the uncertainty level of the van't Veer data set. This chapter shows the intrinsic nature of the uncertainty in the data set.

Chapter 7: concludes the thesis with summary and suggests the direction for future research.

Appendix A: shows the lists of predictive gene list as originally suggested in [90] (List A) and the alternative gene list which is suggested in this thesis (List B).

Appendix B: shows the misclassification matrices according to the visualisation results suggested in chapter 3.

Appendix C: explains basic visualisation methods which are Multidimensional Scaling and Principal Component Analysis.

Appendix D: reviews the standard shadow target algorithm which is used for training the standard NeuroScale models.

Chapter 2

Standard Visualisation Techniques

Dimensionality reduction techniques are becoming significant tools for analysing biological data. The traditional techniques are Principal Component Analysis and Multidimensional Scaling.

Recent developments in visualisation focus on retaining the structure of the high-dimensional data. There are many approaches have been discussed. Topographic mappings are mechanisms that map the data in a high dimensional space into a low dimensional space in such a way that preserves the structure of the data.

This "structure of the data" usually means the similarity between two data samples. In other words, the data samples that are similar in a high dimensional space should stay close together in a lower dimensional space and data that are dissimilar in a high dimensional space should remain apart in the lower dimensional space. This chapter compares and contrasts previous significant nonlinear topographic models that have been established. The aim is to establish the state-of-the-art and to identify weaknesses in existing approaches. Methods can be subdivided into deterministic projection methods and probabilistic and generative models. In generative models we assume a latent variable approach in which data samples in the observation or data space are generated from the latent space over some probability distributions. Deterministic approaches provide more direct projections without use of distributions over generator space. We discuss first deterministic approaches.

2.1 NeuroScale

Neuroscale [51, 86, 60] is a deterministic projective mapping based on the statistical methods of Multi Dimensional Scaling, MDS, a topographic mapping that maps the distribution and relative positions of the points in the projection space representing data vectors to reflect the relative dissimilarity between data measurements in the high-dimensional space, and hence generalises the established Sammon map concept [40]. N measurement vectors $\mathbf{x}_i$, $\{i = 1, \ldots, N\}$ in $\mathbb{R}^D$ are transformed using a Radial Basis Function (RBF) [12, 50] network to a corresponding set of feature (visualisation) vectors $\mathbf{y}_i$ in $\mathbb{R}^d$. An RBF comprises a single hidden layer of h neurons which represents a set of basis functions, each of which has a centre located at some point in the input space. NeuroScale uses the shadow targets optimisation algorithm [Appendix D] to help train the network without predefined target vectors for the RBF network. The shadow targets algorithm gives good generalisation performance since it implicitly incorporates an automatic regularisation process. The network performance is insensitive to the complexity of the RBF network and RBF function models [86] provided the network has sufficient initial complexity. The quality of the projection is measured by the *Sammon stress metric* (n.b. we are using a reduced form here, neglecting a denominator often employed):

$$E = \sum_{i}^{N} \sum_{j}^{N} (d_{ij}^{*} - d_{ij})^2, \tag{2.1}$$

where $d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|$ and $d_{ij}^{*} = \|\mathbf{x}_i - \mathbf{x}_j\|$ represent the inter-point distances in projection space and data space respectively. The aim of the training process is to set the parameters of the RBF, W which is the weight matrix of the network: $y_i = RBF(x_i, W)$. A gradient method is used to minimise the stress metric from each w_{kr}, between k hidden unit and output r dimension.

$$\frac{\partial E}{\partial w_{kr}} = \sum_{i}^{N} \frac{\partial E}{\partial w_{kr}}. \tag{2.2}$$

where,

$$\frac{\partial E}{\partial y_i} = -2 \sum_{j \neq i} \left(\frac{d_{ij}^{*} - d_{ij}}{d_{ij}} \right) (y_i - y_j). \tag{2.3}$$

Hence the stress captures the functional relationship between the original data distribution and the projected images. Once the functional mapping has been obtained

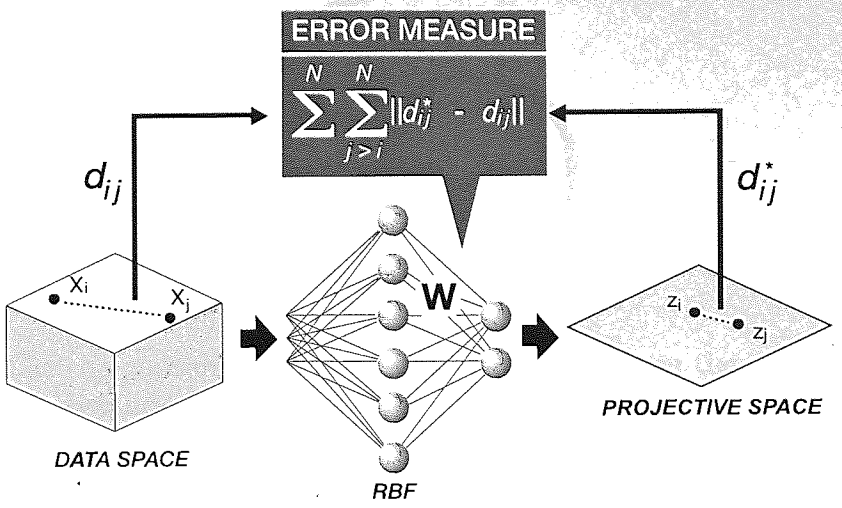


Figure 2.1: The NeuroScale architecture.

using NeuroScale, the model can be reused without reconstruction of the model by using the trained network on the novel data. This is the advantage over the original Sammon mapping. Because the Sammon map only acts as a look-up table, novel data can not be located in the feature space without full reconstruction of a new look-up table.

NeuroScale has parameters, the number and location of the RBF centres that need to be determined. However the outcome is robust to the choice of centres since an implicit smoothing regularisation is used as part of the optimisation process [84]. As normal practice, the number of centres is chosen to be the same as the number of training data points, so that each data point can be used as a centre of the RBF functions. NeuroScale is a deterministic projection approach and lacks a probabilistic interpretation. Therefore if the data is inherently uncertain, NeuroScale simply maps the data samples including the noise. See Figure 2.1 for the architecture of NeuroScale.

2.2 Isometric Mapping

Isometric Mapping (Isomap) [83] builds on classical MDS but improves it by aiming to preserve the geodesic distance of the data. The geodesic distance is defined by the approximation of adding up a sequence of "short hops" between neighbouring points. This can be done by finding the shortest paths in a graph with edges connecting neighbouring data points [83].

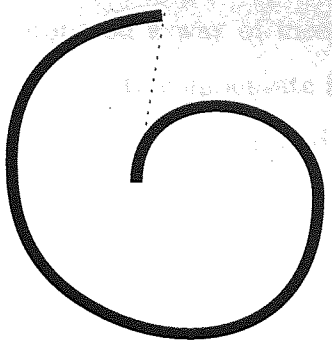


Figure 2.2: The comparison of the two measurements between the distance measured by the geodesic distance and the Euclidean distance.

Figure 2.2 shows that the distance between two points which is measured by Euclidean distance in high dimensional space may not be a good measure of structural similarity. The true measure of this data set or the geodesic distance, is the measure along the surface of the S-curve. Isomap estimates this geodesic manifold distance between data points. It is done by first, defining the graph between two data points, i and j by connecting both points if both points lie in some fixed radius(ϵ) or if i is the K nearest data point of j. The distance of these neighborhoods is defined as $d_x(i,j)$. Next, the geodesic distance between two data points is obtained by computing the shortest path distances in the graph($d_G(i,j)$). This distance is initialised by $d_G(i,j) = d_x(i,j)$ if i,j are linked by edge and $d_G(i,x) = \infty$ otherwise. Then, for $K = 1, 2, \ldots, N$ in turn, replace $d_G(i,j) = \min(d_G(i,j), d_G(i,k) + d_G(k,j))$. The final matrix of graph distance is therefore D_G. Finally, apply MDS (See Appendix C) to preserve the intrinsic distance. The coordinate y_i is chosen to minimise the cost function

$$E = \|\tau(D_G) - \tau(D_Y)\|, \tag{2.4}$$

where D_Y is the Euclidean distances of the projecting space. The τ operator converts distance to inner product.

Isomap is related to the NeuroScale method if the input dissimilarity is changed to reveal the intrinsic distance of the data rather than using just a normal Euclidean distance. The NeuroScale results will be similar to the Isomap but with an extra advantage of interpolation ability which can create a transformation mapping. Another

related method, S-Isomap [28] proposed a way of incorporating class information to the dissimilarity measures which claims to compensate for the noise if the class of each datum is known. However, this method is a supervised technique which will have the same problem as expected for classifiers if data are not representative or dense enough to model each class.

2.3 Locally Linear Embedding

Locally Linear Embedding (LLE) [71] is another local method, similar to Isomap. It focuses on preserving the topographic distance in small neighborhoods by using an eigenvector method [72]. The LLE uses the linearity in the local area and overcomes many limitations that occur in a fully global linear method. We define $\mathbf{x}$ to be a vector of N data points in D dimensional space, sampled from some smooth underlying manifold. Provided there is sufficient data, we expect each data point and its neighbours to lie on or close to a locally linear patch of the manifold. In the simplest formulation of LLE, we need to identify K nearest neighbours per data point, as measured by Euclidean distance from a point of interest. This cost function is defined by:

$$\varepsilon(W) = \sum_{i}^{N} |\mathbf{x}_i - \sum_{j=1}^{K} W_{ij}\mathbf{x}_j|^2. \tag{2.5}$$

W_{ij} is a weight between a point i and its neighbours j. In order to compute the appropriate weight, we minimise the cost function in the equation (2.5) subject to two constraints: first, that each data point $\mathbf{x}_i$ is reconstructed only from its neighbours, set W_{ij}=0 if $\mathbf{x}_j$ does not belong to this set; second, that the rows of the weight matrix sum to one: $\sum_{j=1}^{K} W_{ij} = 1$.

If we look at the equation (2.5) locally, the error contribution ε from each data point $\mathbf{x}_i$ conditioned on $\sum_{j=1}^{K} W_{ij} = 1$ can be written as:

$$\varepsilon_i = |\sum_{j}^{K} W_{ij}(\mathbf{x}_i - \eta_j)|^2 = \sum_{j}^{K}\sum_{k}^{K} W_{ij}W_{ik}C^i_{jk}, \tag{2.6}$$

where C^i_{jk} are elements of a covariance matrix within the neighbourhood of $\mathbf{x}_i$ and $\boldsymbol{\eta_j}$ is the neighbour of the data point $\mathbf{x}_i$.

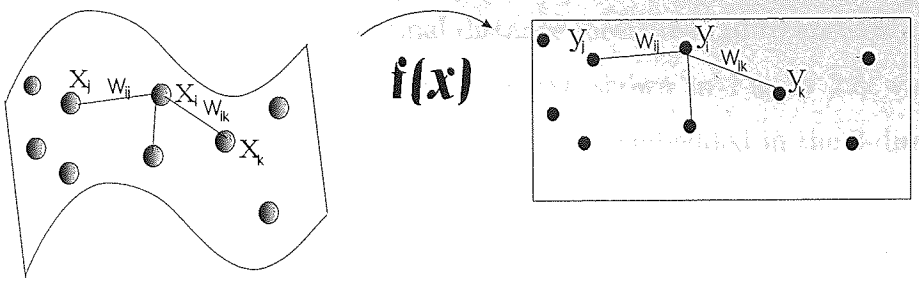


Figure 2.3: The Locally Linear Embedding algorithm.

$$C^i_{jk} = (\mathbf{x}_i - \boldsymbol{\eta}_j)^T.(\mathbf{x}_i - \boldsymbol{\eta}_k). \tag{2.7}$$

This cost function can be minimised by using a Lagrange multiplier to enforce the constraint that $\sum_j W_{ij} = 1$. Therefore, the optimal weights are given by:

$$W_{ij} = \frac{\sum_k C^{i-1}_{jk}}{\sum_{lm} C^{i-1}_{lm}}. \tag{2.8}$$

Each high dimensional point $\mathbf{x}_i$ is mapped to a low dimensional point $\mathbf{y}_i$ in low dimensional space, representing global internal coordinates on the manifold. This is done by choosing d-dimensional coordinates $\mathbf{y}_i$ to minimise the cost function in low dimensional space:

$$\Phi(Y) = \sum_i^N |\mathbf{y}_i - \sum_j^K W_{ij}\mathbf{y}_j|^2, \tag{2.9}$$

where W_{ij} is fixed from (2.8). This cost function can be minimised by solving sparse $N \times N$ eigenvector problem, whose bottom d non-zero eigenvectors provide an ordered set of orthogonal coordinates centered on the origin.

This algorithm has only one free parameter: the number of neighbours per data point, K. The higher the value of K, the more similar to the NeuroScale method this method will be. Practically, it is very hard to find a value of K to suit the given data set. Furthermore, it is hard to find an appropriate value of K which performs well across different choices of data sets. It is typically much smaller than the number of data points. Figure 2.3 summarises the LLE algorithm.

However, the algorithm is easy to implement and it is claimed not to have local minima problems as many other non-linear methods have encountered. Similar to Isomap, which control the number of neighborhood points, these local methods have big advan-

tages on embedding data in which normal distance measures fail to correctly evaluate the geodesic distance. An example is the S-curve shown in Figure 2.5 which LLE and Isomap properly reduce the 2-dimensional surface embedded in the 3-dimensional space, as shown in Figure 2.5(a).

The main drawback of local methods such as LLE is that although they preserve topography in the local area, most of the time they lack the ability to verify the global alignment of different local neighborhoods.

LLE can give more faithful representations when the real distances between points are largely different from Euclidean distance measures. However, when the true distance can be computed straightforwardly or is close to Euclidean distances, using the local approach is not so useful. LLE and Isomap are similar in the way that both of them are topographic which is to use the local distance in the original data space as an important criterion. We now discuss a class of methods motivated from a different perspective, linked to the intent of incorporating probabilistic knowledge.

2.4 Probabilistic PCA

The previous methods are based on the neighbours of the data deterministically and do not take into account the level of noise in the data space. Probabilistic Principal Component Analysis (PPCA) [85] uses a probabilistic approach extending the standard method of traditional PCA (see Appendix C). The probabilistic PCA models the projection of the data to $\mathbf{y}$ from the observed variable $\mathbf{x}$ with a probabilistic generative model,

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\{\frac{1}{2\sigma^2}\|\mathbf{x} - W\mathbf{y} - \boldsymbol{\mu}\|\}, \tag{2.10}$$

where $\boldsymbol{\mu}$ allows the observed data to have nonzero mean with an added Gaussian noise, $\eta \sim N(0, \sigma^2 I)$. The latent variable, conventionally, is assumed sampled from an independent Gaussian distribution with a unit variance, $y \sim N(0, I)$. The objective function is the likelihood:

$$P(\mathbf{x}_i|\mathbf{y}_i, \mathbf{W}, \sigma) = N(\mathbf{x}_i|\mathbf{W}\mathbf{y}_i + \boldsymbol{\mu}, \sigma^2 I). \tag{2.11}$$

Marginalising the latent variable, we obtain

$$P(\mathbf{x}_i|\mathbf{W},\sigma) = N(\mathbf{x}_i|\mu, \mathbf{W}\mathbf{W}^T + \sigma^2 I). \tag{2.12}$$

The objective function is the log likelihood of the function:

$$L = -\frac{N}{2}\{d\ln(2\pi) + \ln|C| + tr(C^{-1}S)\}, \tag{2.13}$$

where $C = \mathbf{W}\mathbf{W}^T + \sigma^2 I$. Maximising the objective function to obtain the optimum solution for W and σ:

$$\mathbf{W}_{ML} = U_d(\Lambda_d - \sigma^2\mathbf{I})^{1/2}\mathbf{R}, \tag{2.14}$$

where the d column vectors of the matrix $\mathbf{U}_d \in \mathbb{R}^{D\times d}$ are the principal eigenvectors of the data covariance matrix, S, with corresponding eigenvalues in the $d \times d$ diagonal matrix Λ_d and $\mathbf{R}$ is a rotation matrix which can be chosen to be $\mathbf{R} = \mathbf{I}$. Similarly,

$$\sigma^2_{ML} = \frac{1}{D-d}\sum_{j=d+1}^{D}\lambda_j, \tag{2.15}$$

which is clearly the variance that is lost during the projection. For dimensionality reduction purposes, the projection of $\mathbf{x}$ can be defined as a posterior mean of $\mathbf{y}$

$$\langle \mathbf{y}_i|\mathbf{x}_i\rangle = \mathbf{M}^{-1}\mathbf{W}^T_{ML}(\mathbf{x}_i - \boldsymbol{\mu}), \tag{2.16}$$

where $M = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}$ is a covariance of size $d \times d$. PPCA can also be extended from a single component Gaussian to a mixture density with a mixing coefficients π_n

$$p(\mathbf{x}_i) = \sum_{n=1}^{M}\pi_n p(\mathbf{x}_i|n), \tag{2.17}$$

where $p(\mathbf{x}_i|n)$ is a single PPCA model. R_{in} is the posterior responsibility of mixture n for the generating data $\mathbf{x}_i$ defined as $R_{in} = \frac{p(\mathbf{x}_i|n)\pi_n}{p(\mathbf{x}_i)}$, with mixing coefficient and mean of each component as:

$$\tilde{\pi}_n = \frac{1}{N}\sum_{i=1}^{N}R_{in}, \tag{2.18}$$

$$\tilde{\mathbf{y}}_n = \frac{\sum_{n=1}^{N}R_{in}\mathbf{x}_i}{\sum_{n=1}^{N}R_{in}}. \tag{2.19}$$

The reduced dimensionality projection data can be obtained from the posterior mean of the highest π_n. This method is one of first methods of the probabilistic approach to visualisation.

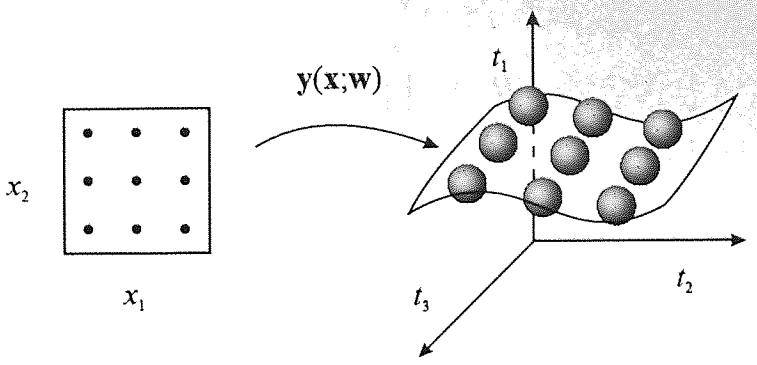


Figure 2.4: The GTM architecture.

2.5 Generative Topographic Mapping

Similar to the previous PPCA, the Generative Topographic Mapping (GTM) [9] is a probabilistic model. However, this method uses a generative model transforming from the latent variable to the data space by using an RBF neural network which makes it straightforward for training and applying to the unseen data set. This algorithm uses a constrained mixture of Gaussians from a basic concept of SOM [43] in that it has representative nodes in the low dimensional space. However, GTM uses a probabilistic density model to form the high dimensional space generated from the low dimensional grid. The GTM architecture is shown in Figure 2.4

GTM overcomes the drawbacks of those in SOM; specifically the lack of an explicit cost function, and a transformation and convergence criteria, between two spaces which are linked by a function $f(\mathbf{t};\mathbf{W})$ which maps $\mathbf{t}$ to $\mathbf{x}$ and forms a corresponding Gaussian with a common standard deviation σ and parameterised by W. The distribution of the data $\mathbf{x}$ conditioned on latent variables $\mathbf{t}$ is given by

$$p(\mathbf{x}|\mathbf{t},\mathbf{W},\sigma) = \frac{1}{(2\pi\sigma^2)^{d/2}}\exp\{-\frac{||f(\mathbf{t};W)-\mathbf{x}||^2}{2\sigma^2}\}. \tag{2.20}$$

The density in data space is obtained by integrating out the latent variables and choosing the prior $p(x)$ to be a sum of delta functions; the marginal probability becomes

$$p(\mathbf{x}|\mathbf{W},\sigma) = \frac{1}{M}\sum_{j=1}^{M} p(\mathbf{x}|\mathbf{t}_j,\mathbf{W},\sigma). \tag{2.21}$$

The model parameters W can be obtained by simply using maximum log likelihood,

given by

$$\mathcal{L}(W) = \ln \prod_{n=1}^{N} p(\mathbf{t}_n|\mathbf{W}). \tag{2.22}$$

Equations (2.21) and (2.22) give

$$\mathcal{L}(\mathbf{W}, \sigma) = \sum_{n=1}^{N} \ln\{\frac{1}{M} \sum_{j=1}^{M} p(t_n|x_j, \mathbf{W}, \sigma)\}. \tag{2.23}$$

By choosing an RBF for the function $f(\mathbf{t}; \mathbf{W})$, we can write

$$f(\mathbf{t}; \mathbf{W}) = \mathbf{W}\boldsymbol{\phi}(\boldsymbol{t}), \tag{2.24}$$

where the elements of $\boldsymbol{\phi}(\boldsymbol{x})$ consist of K fixed basis functions of $\phi_i(\mathbf{x})$, and $\mathbf{W}$ is a $d \times K$ matrix.

GTM uses the (iterative) EM algorithm which in the E-step calculates the posterior probability, or responsibility, of each component j of each data point t_n. We get

$$R_{jn}^{(m)}(\mathbf{W}^{(m)}, \sigma^{(m)}) = \frac{p(\mathbf{x}_n|\mathbf{t}_j, W^{(m)}, \sigma^{(m)})}{\sum_{j'=1}^{M} p(\mathbf{x}_n|\mathbf{t}_{j'}, \mathbf{W}^{(m)}, \sigma^{(m)})}, \tag{2.25}$$

where m indexes the iteration number.

The M-step consists of maximising the expectation with respect to W of the complete-data log likelihood giving:

$$\langle \mathcal{L}_{comp}(W) \rangle = \sum_{n=1}^{N} \sum_{j=1}^{M} R_{jn}^{(m)}(\mathbf{W}^{(m)}, \sigma^{(m)}) \ln\{p(\mathbf{t}_n|\mathbf{x}_j, \mathbf{W}, \sigma)\}. \tag{2.26}$$

Maximising the expectation of the complete-data log likelihood (2.26) with respect to W together with (2.20) and (2.24), gives:

$$\mathbf{\Phi}^T \mathbf{G}^{(m)} \mathbf{\Phi} (\mathbf{W}^{(m+1)})^T = \mathbf{\Phi}^T \mathbf{R}^{(m)} \mathbf{T} \tag{2.27}$$

where $\mathbf{\Phi}$ is the $M \times K$ RBF design matrix with elements $\mathbf{\Phi}_{ji} = \phi_i(\mathbf{x}_j)$, $\mathbf{T}$ is the $N \times d$ data matrix, $\mathbf{R}$ is an $M \times N$ responsibility matrix with elements R_{jn} and $\mathbf{G}$ is an $M \times M$ diagonal matrix with elements $G_{jj} = \sum_{n=1}^{N} R_{jn}(\mathbf{W}, \sigma)$.

Similarly, maximising (2.26) with respect to σ^2 we obtain the following re-estimation formula,

$$(\sigma^{(m+1)})^2 = \frac{1}{Nd} \sum_{n=1}^{N} \sum_{j=1}^{M} R_{jn}^{(m)}(\mathbf{W}^{(m)}) \|\mathbf{W}^{(m+1)} \phi(\mathbf{t}_j) - \mathbf{x}_n\|^2. \tag{2.28}$$

For visualisation purposes, GTM can summarise the whole data set by using the means of each component in the mixture model distribution and gives the projection data $\mathbf{y}$,

$$\mathbf{y} = \langle \mathbf{t} | \mathbf{x_n}, \mathbf{W}, \sigma \rangle = \sum_{j=1}^{M} R_{jn} \mathbf{t}_j. \tag{2.29}$$

Alternatively, the mode of the posterior can be used

$$j_{max} = \mathtt{argmax}_j R_{jn}. \tag{2.30}$$

GTM has the advantage of using a probabilistic approach which is more flexible with noise in the data. Probabilistic approaches are more suitable for dealing with uncertainty in the data than deterministic approaches. However, the number of RBF basis functions and distribution of the latent space sample points are chosen by hand. The complexity of the model is determined by the number of RBF centres. The number of latent points, $\mathbf{t}_n$ helps determine data model. Too few latent points compared to the number of basis functions will result in the Gaussian components becoming relatively independent and effectively no smooth manifold will be found. For visualisation purposes, the number of latent nodes and the latent shape need to be specified in advance. However, there is no correct way of choosing these parameters. As a result, the visualisation can depend on the choices of these parameters.

2.6 Stochastic Neighbour Embedding

Stochastic Neighbour Embedding (SNE) [36] borrows a concept from NeuroScale as it uses a pairwise similarity between points but measures similarity using a probabilistic distance approach to preserve the neighbourhood identities. A Gaussian distribution is centred on each data point in the high dimensional space and a probability density is defined over all the potential neighbours of that point. This approach permits a 1-to-many mapping of high dimensional points to projection space as discussed later.

The high dimensional related probability for each point, i, and each potential neighbour, j, is computed using the asymmetric probability, p_{ij}, that i would pick j as its neighbour,

$$p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ik}^2)}. \tag{2.31}$$

The dissimilarities, d_{ij} can be based on standard Euclidean distances and scaled by a smoothing factor σ_i which is empirically determined, $d_{ij} = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma_i^2}$.

The low dimensional images $\mathbf{y}_i$ of the points are used to define a probabilistic density in the mapping space, q_{ij}, as:

$$q_{ij} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k \neq i} \exp(-\|\mathbf{y}_i - \mathbf{y}_k\|^2)}. \tag{2.32}$$

The aim of this SNE method is to match the above two distributions as closely as possible. The Kullback-Leibler divergence, which is a measure of dissimilarity between two probability distribution is used here as a cost function. This can be achieved by manipulating the coordinates $\mathbf{y_i}$ to minimise the cost:

$$C = \sum_i^N \sum_j^N p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{2.33}$$

The SNE model can be extended to multiple projections of a single object by using a mixture of densities, which produces a probabilistic density in the mapping space:

$$q_{ij} = \sum_b \pi_{i_b} \sum_c \frac{\pi_{j_c} \exp(-\|\mathbf{y}_{i_b} - \mathbf{y}_{j_c}\|^2)}{\sum_k \sum_d \pi_{k_d} \exp(-\|\mathbf{y}_{i_b} - \mathbf{y}_{k_d}\|^2)}. \tag{2.34}$$

The number of clusters in the mixture also needs to be determined empirically. Each data point $\mathbf{x}_i$ can be projected to the various modes of the mixtures $\mathbf{y}_{i_b}$ or $\mathbf{y}_{i_c}$. The benefit of mapping a single data point to multiple locations allows ambiguous objects, like the document count vector for the word bank, to have meaning close to the images of both river and finance without forcing the images of outdoor concepts to be located close to those of corporate concepts. However, for the experiments in this paper, only one projection modes per datum will be used. Because it is quite hard to show in practice.

The main advantage of SNE is its probabilistic approach but the results of the SNE are strongly dependent on the chosen σ. If the chosen σ is too large, the projecting data is likely to collapse to a single point. The suggested σ is $\sigma = \log(K)$, where K is the number of neighbours used to define a local cluster. However the main drawback is the proper way of determining σ to properly fit the data without knowing the underlying projection in advance.

2.7 Parametric Embedding

Another recently proposed method of dimensionality reduction method is Parametric Embedding (PE) [38] which is an extensive of SNE. However this method requires class labels to be specified in advances, since PE uses the conditional probabilities of classes given data. These probabilities can be given as a supervised model. However, the difference to the other supervised classification models is that rather than using the supervised information given as a hard classification, it is given by a probabilistic representation in term of conditional probabilities.

This approach projects both data points and data labels in the embedding space by assuming a spherical Gaussian distribution around each class, c_k. The algorithm will try to match the location of all data points, $\mathbf{y}$ with high the conditional probabilities of any class and will be projected down close to the class location, ϕ, in the low dimensional space.

Parametric embedding approximates conditional probability, $p(c_k|x_n)$, under the assumption of a unit variance spherical Gaussian mixture model in the embedding space

$$p(c_k|\mathbf{y}_n) = \frac{p(c_k)\exp(-\frac{1}{2}\|\mathbf{y}_n - \phi_k\|^2)}{\sum_{l=1}^{K} p(c_l)\exp(-\frac{1}{2}\|\mathbf{y}_n - \phi_l\|^2)}, \tag{2.35}$$

where $\|.\|$ is the Euclidean norm in the projection space.

The aim is to match the conditional probabilities between $\mathbf{x}$ and $\mathbf{y}$ by using the reduced form of Kullback-Leibler divergence which measures the distance between $p(c_k|\mathbf{x}_n)$ and $p(c_k|\mathbf{y}_n)$:

$$E(\mathbf{y}_n, \phi_k) = -\sum_{n=1}^{N}\sum_{k=1}^{K} p(c_k|\mathbf{x}_n)\log p(c_k|\mathbf{y}_n). \tag{2.36}$$

The derivatives of (2.36) are:

$$\frac{\partial E}{\partial \mathbf{y}_n} = \sum_{k=1}^{K} \alpha_{n,k}(\mathbf{r}_n - \phi_k), \quad \frac{\partial E}{\partial \phi_k} = \sum_{k=1}^{K} \alpha_{n,k}(\phi_k - \mathbf{r}_n), \tag{2.37}$$

where $\alpha_{n,k} = p(c_k|\mathbf{x}_n) - p(c_k|\mathbf{y}_n)$.

The visualisation result of y_n depends on the initial coordinates of classes ϕ_k. Both data, $\mathbf{y}$ and ϕ, can be optimised by any non-liner optimisation algorithms. The advan-

tage of this method is that it can be used directly as a classifier. Class labels are also projected down in the low dimensional projection which make very clear distinctions between classes. However, for the general case where classification is not essential, this method may not be useful since class labels need to be specified in advance. Moreover, the appropriate way of estimating the conditional probabilities, $p(c|k)$ is very important in determining the visualisation results.

2.8 Other techniques

The Autoencoder [18] uses a multi-layer feedforward neural network combining an encoder which transforms the high dimensional data to low dimensional data and a decoder which recovers the data from the code by using multi-layer hidden units of a neural network. Many layers are needed to be trained, therefore it is very difficult to find the optimum global solution.

Hinton [35] claims the technique, "Restricted Bolztman Machine" can initialise the weights closer to the optimal solution for this autoencoder method.

ISOTOP [47] uses a modification of a SOM using a neural network to create a functional mapping but it faces the same drawback as SOM as it does not have an explicit cost function.

Stochastic Proximity Embedding [2] uses the geodesic distance by trying to preserve the distances in the low dimensional space not to fall below the distances provided by distance from the original high dimensional space.

There are also techniques which combines global and local techniques, such as, Local Linear Coordination (LLC) [82].

2.8.1 Computational Complexity

This section gives a brief overview of the computational complexity of each visualisation techniques, which is important for the applications. For number of data n and k nearest neighbours, Isomap performs eigen analysis of $n \times n$ matrix and additional neighbourhood search gives the computational time of $(k + \log(n) + n^2)n$. In contrast, SNE computes distances of matrix $n \times n$ for i iterative time, therefore the compu-

tational complexity is $O(in^2)$. LLE has computational complexity from computing eigenvector from sparse matrix($O(pn^2)$) and solving linear system of size $k \times k$, the overall complexity is $(k3 + pn + \log(n))n$ For NeuroScale and GTM, the complexity depend mainly on the size of training data and number and form of basis functions. The main advantage of these two method is that with the large data set, not all data points are necessarily included in the training phase. Once the model is trained both GTM and NeuroScale can retrieve the low dimensional projections of those novel data point with small computational requirement, which is different to all other methods.

2.9 Discussion

Global techniques have the advantage that overall properties and structure are retained while local models use local properties of nearby points which sometimes may not reflect global metric properties. NeuroScale can reflect local properties without any interference to the model by exploiting hand-tuned parameters such as σ in a Gaussian RBF network or by altering the input dissimilarity matrices to reveal the intrinsic distances of the data. Local methods have an advantage over global methods when the intrinsic distance is different from the global metric properties, such as a 2-dimensional embedded S-curve in a 3-dimensional space as shown in Figure 2.5(a). This figure shows the projection using both global and local techniques. Local techniques, such as LLE 2.5(c), Isomap 2.5(d), reveal the true embedding data of the s-curve while the global methods achieve poor reconstructions. Other techniques fail to correctly keep dissimilar points apart, such as some areas at the edges of the curve. However, if the number of neighbours is chosen incorrectly the results will be no different to those in the global techniques. Even though many global techniques are not suitable for this example, most of them are not as sensitive to parameter choice as the local methods. NeuroScale generally produces consistent mappings with different parameter settings given the same distance measure.

Using this example for comparison with some global techniques, PCA and NeuroScale gave similar results in 2.5(e) in which the edges of the curve combine with the middle part of the S-curve and are seen as nearby points and hence projected down

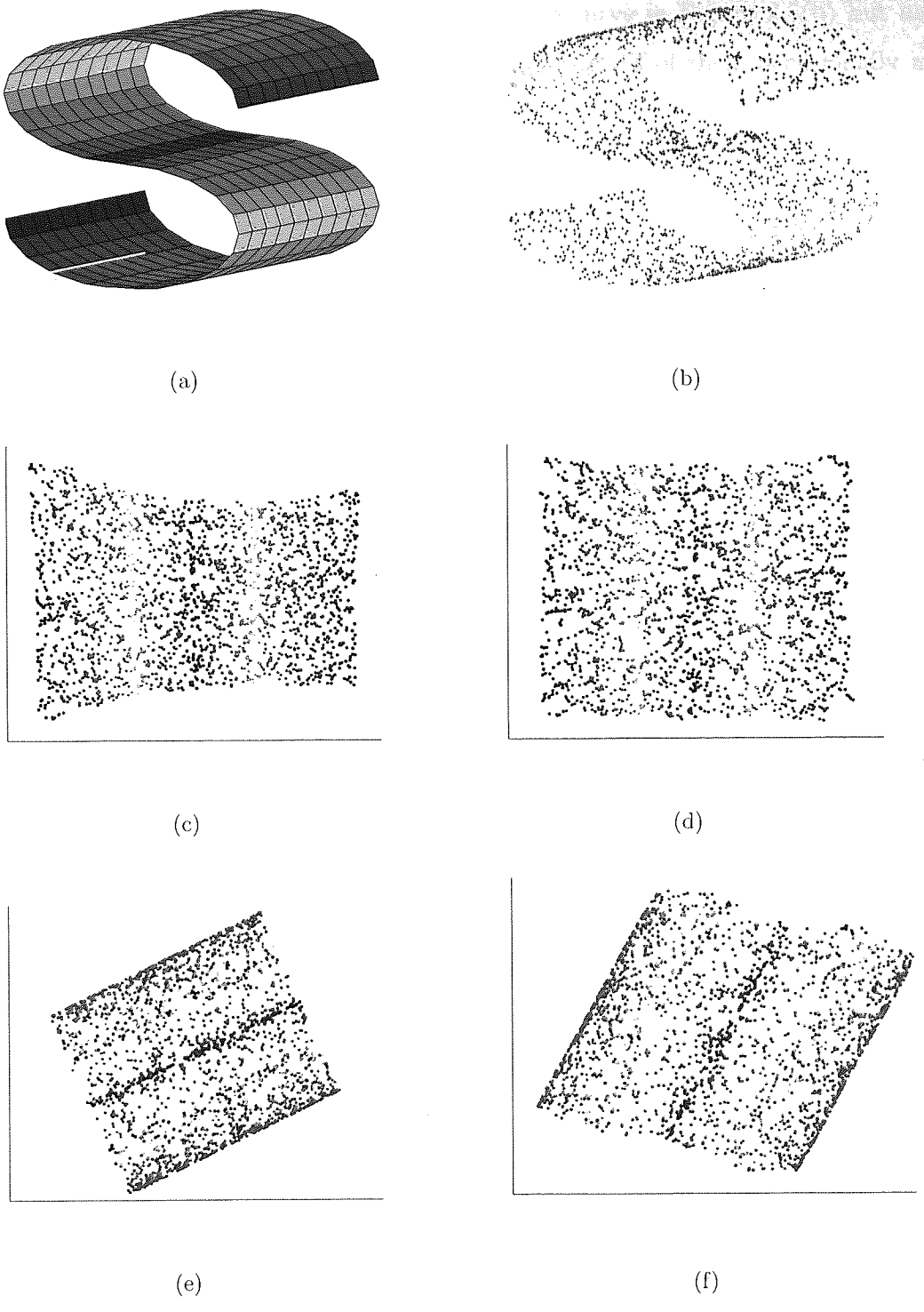


Figure 2.5: The three dimensional data (b) sampled from two-dimensional manifold (a). The LLE algorithm discovers the neighbourhood-preserving mappings shown in (c) where $K = 12$; the colour coding reveals how the data is embedded in two-dimensions. (d) shows the resulting mapping using Isomap. PCA (e) and NeuroScale using Euclidean similarity (f) incorrectly map the intrinsically faraway points close together.

close together.

Figure 2.5(e) shows the results from the same S-curve in Figure 2.5(b) but using PCA (Figure 2.5(e)) and NeuroScale (Figure 2.5(f)). Both of them occasionally map faraway points to nearby points in this example.

Another example is shown in figure 2.6. This example has data in 2 clusters with one point between clusters in 3 dimensional space shown in figure 2.6(a). Different clusters are colour coded with different colours. Figure 2.6(b) and 2.6(c) show successful data projections using PCA and NeuroScale respectively. Figure 2.6(d) shows the LLE projection using $K = 15$ where the structure of one of the clusters is distorted. It can be seen that the local methods do not well preserve the original structure in this type of example where data from different clusters can be viewed as nearby neighbors. Global methods do not have this problem.

In addition, the approaches have been categorised into deterministic and probabilistic approaches. SNE, PE and GTM use probabilistic intuition by assuming Gaussian distributions centred around each data point. The stochastic approach is easier for mapping a single data point to multiple locations in a low dimensional space. However, the known uncertainty knowledge cannot be used with these probabilistic models. Modifications of these models are required.

Furthermore, the main drawback of existing deterministic techniques is that they do not support uncertainty information if it is available. Uncertainty is dealt with by alternative approaches such as regularisation. These techniques assume all data points have the same uncertainty information which is not always true as discussed previously in Chapter 1. Techniques using probabilistic approaches are potentially useful in modelling the data with uncertainty. In addition, the main advantage of using explicit mapping functions such as neural networks for dimensionality reduction is that they provide an explicit implementation of applying the trained network to new unseen data points. [6] discusses the possibility of the extension of using LLE, Isomap and MDS with a small modification of the algorithms to be able to project unseen data. NeuroScale which is currently a deterministic approach and SNE and PE which are probabilistic approaches have the same idea of preserving the topology of the data. However, NeuroScale has the advantage of using neural networks to implement

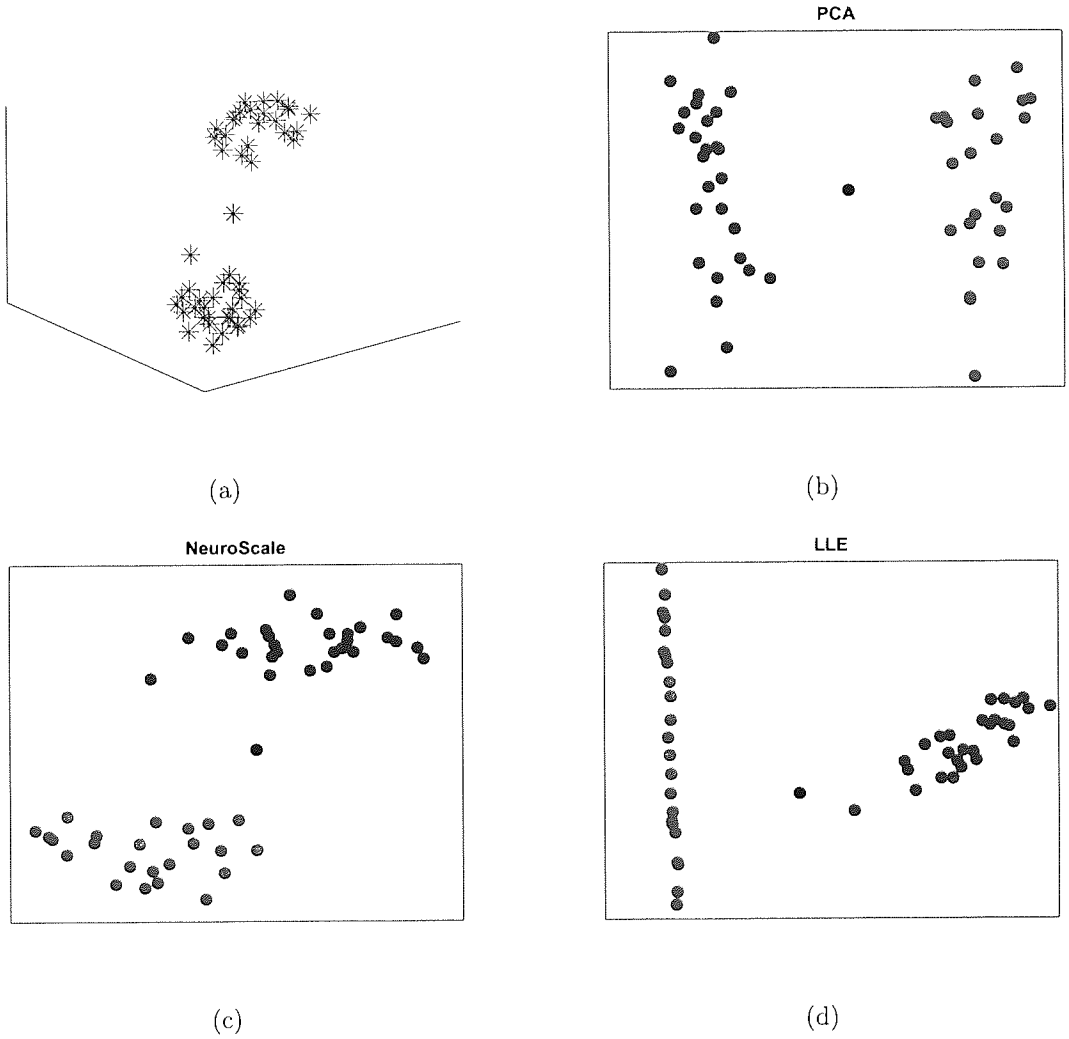


Figure 2.6: The three dimensional data (a). The LLE algorithm discovers the neighbourhood-preserving mappings shown in (d) where $K = 15$; using PCA (b) and NeuroScale (c) can well preserve the structure of this data set.

the projection while the latter two just use simple look-up optimisation techniques which are not as useful as building mapping functions between the data space and the projection space. The main disadvantage of NeuroScale - its inability to explicitly deal with uncertainty - will be examined later in the thesis when we extend the method into a probabilistic NeuroScale, capable of incorporating uncertainty and local information.

Next chapter will examine some of these established methods and applied to the controversial microarray data set in order to see the hidden uncertainty in the data set.

Chapter 3

Prognosis Gene List and Patient Visualisation

3.1 Introduction

This chapter discusses the traditional approach to data analysis which neglects imprecision in the data. We illustrate standard methodologies on an important real-world study, the van't Veer data set, discussed in the first chapter. An important consequence that this chapter will illustrate is that generic finite feature subset selection in such high-dimensional noisy data can have a strong correlation with a desired outcome. The implication of this is that the selection of predictive features in such situations is not reliable. This is a further motivation to include uncertainty in such analysis tools.

3.1.1 The van't Veer data problem

In the area of genetic diseases, which includes the van't Veer study, there has been controversy surrounding the non-uniqueness of 'predictive gene lists' (PGL) to be used as prognostic indicators based on small selected subsets of genes from very large numbers of potential candidates as available in DNA microarray experiments [7, 45, 24]. Many studies have focused on constructing discriminative semi-parametric models and as such are also subject to the issue of random correlations of sparse model selection in high dimensional spaces.

However, it still remains controversial whether the use of gene expression profiles

really outperforms traditional methods based on non-genomic biodata. Although many studies claimed success in using gene expression profiles, there was little commonality across preferred gene subsets [24]. In recent work [70, 22], it was argued that the 70-gene subset of the van't Veer study [90] does not significantly outperform existing clinical criteria based on non-genomic biomarkers. In this latter work, the gene expression prognosis ability was compared to the Nottingham Prognostic Indicator (NPI) and traditional non-genomic biomarkers. This lack of superiority of the genomic approach is not surprising from a systems biology perspective, where we would regard cancer as the result of complex interactions between genetic, temporal, biological and environmental influences.

The original gene selection process used the correlation between the response and outcome to rank potential genes. The problem is whether the quality gene expression data values are accurate enough for extracting appropriate numbers of genes. Although feature selection and feature extraction are often unsupervised methods, in the literature in this domain it is more usual for feature selection for a supervised approach to be used based using a specific choice of a nonparametric model linking the gene expressions to the outcomes. Therefore, the question arises as to whether a specific PGL can be obtained based on clinical datasets, given these concerns over reliability of pattern processing techniques.

This chapter explores visualisation projections of high dimensional data using several nonlinear visualisation models to introspect the van't Veer breast cancer study to investigate whether PGL determined by the van't Veer group can be used as a successful biomarker. As a comparison with the original PGL selected by the van't Veer study, an alternative PGL based on cross-patient consistency is selected. We examine and compare the performance of our alternative PGL to the claimed prognostic PGL from the van't Veer study. Additionally, in this chapter we will construct supervised classifiers to make a comparative investigation of the prognosis indicator of each patient using the resulting projections from visualisation techniques and to investigate whether *a-posteriori* two prognosis groups are separable on the evidence of the gene lists.

3.1.2 The previous study of non uniqueness gene list

In [24], it was shown that the top 70 most correlated genes in the van't Veer study can vary significantly depending on the specific training set of patients used. Different randomly selected 70 gene PGL's were selected and shown to have similar prediction ability. They suggested that there is no unique set of genes that can be assumed to be the best or the only set of genes for prognosis accuracy of breast cancer. A follow-up study [25] also suggested a similar conclusion, that we cannot create a definitive classifier from a small subset of genes based on the small patient datasets available. Generally, large patient sample sizes are needed to produce viable and robust prediction outcomes of cancer prognosis.

An alternative PGL

To illustrate the lack of prognostic uniqueness of capability of the van't Veer gene list, which we denote List A, we will compare results using a different gene list, denoted List B, selected on the basis of cross-patient consistency rather than maximising classification accuracy on a specific classification model [75].

Let $\mathbf{x}^i$ denote the gene expression vector for patient i of the van't Veer PGL. $\mathbf{x}_G$, where $G \in \{1, 2, \ldots, 44\}$ represents a set of expression values across all good prognosis patients, and $\mathbf{x}_P$, where $P \in \{45, 46, \ldots, 78\}$ represents the set of all poor prognosis patients. The expression values have been normalised before the selection.

The variance of individual gene expression values within each patient group is estimated by

$$\sigma_L^2 = \langle (\mathbf{x}_i - \bar{\mathbf{x}}_L)^2 \rangle_{i \in L},$$

where $L = \{G, P\}$ and the average is taken across all patients. Assume R_j^L is the rank order of gene j by variance for each patient group. The unique top T ranked genes from each group are extracted,

$$L_G = \{j | R_j^G \leqslant T\},$$

$$L_P = \{j | R_j^P \leqslant T\}.$$

The number of genes, T, is chosen so that List B has a total number of genes equal to 70, the same as List A. Specifically, in this case the 35 lowest non-overlapped

variance genes from each patient group were extracted.

$$\texttt{List B} = \{L_G \cup L_P\} - \{L_G \cap L_P\}.$$

This selection criterion emphasises *consistency* of gene expression across patients within patient group, rather than explicitly seeking discrimination. Examining the details of the two 70-gene subsets, we observe that there are only *six* genes in common between the van't Veer study and our alternative gene list. List B is trying to search for the gene expressions which are similar in the same patient group to extract genes that act differently in different conditions and may reveal the true underlying cause of the cancan patients. It can be expected that patients with the same prognosis outcome is likely to have a similar molecular profile [87].

If List A has superior prognostic value as mentioned in the literature, its projective visualisation and discrimination properties should be better than those of List B, since List A was chosen explicitly to maximise discrimination using a supervised training process while the maximisation of classification accuracy is not the main criteria of List B. The list of gene in List A and List B can be found in Appendix A.

Both gene lists will be used to compare the separability between pre-specified patient groups by using standard but state-of-the-art visualisation techniques.

3.2 Experiments on the van't Veer data

Since microarray data distributions have a non-linear structure in high dimensional spaces, four different approaches of non-linear data visualisation methods were used for comparison; the NeuroScale model, Generative Topographic Mapping, Local Linear Embeddings (LLE) and Stochastic Neighbour Embeddings (SNE) as have been discussed in the previous chapter.

Both List A and List B were used for constructing 2-dimensional projections.

3.2.1 Classifier Comparison

For comparison with the literature, we will also construct classifiers based on the two gene lists.

In order to avoid problems of high dimensionality in small data samples that might have occurred in the original study which constructed classifiers using leave-one-out cross-validation. Alternatively, we constructed classifiers on the visualisation space which sensibly reduced the dimensionality structure of the data. The classifiers are constructed on the two dimensional input of the visualisation space using separate RBF nonlinear classifiers [77, 78]. The classifiers use 2 coordinate input values and produce 2 output values indicating good and poor prognosis likelihood respectively and are trained using the original 78 patients. Specifically, the desired target value is $\mathbf{T} = \{T_1, T_2\}$ where $\mathbf{T} \in \{[1,0],[0,1]\}$ represents good and poor prognosis patients respectively.

The outputs of the RBF network are then transformed using the softmax function, giving a vector prognosis indicator for each patient, $P = \frac{\exp(\mathbf{y})}{\sum_j \exp(y_j)}$. One of the two outputs which represents the good prognosis class is used as an indicator and contours of the indicator values are superimposed on the projection map space to show the likelihood of the good prognosis indicators.

The use of classifiers will show the 'confidence' of being in one class versus the other. Furthermore, the classifier should provide evidence for the separability performance of each gene lists. The contour lines showing the results of the classifier will be superimposed on the visualisation results. Patients with predicted prognosis values in the range $0.3 \rightarrow 0.7$ are considered as ambiguously classified. This range is chosen to maximise the classification results for both List A and List B.

3.2.2 NeuroScale Projection

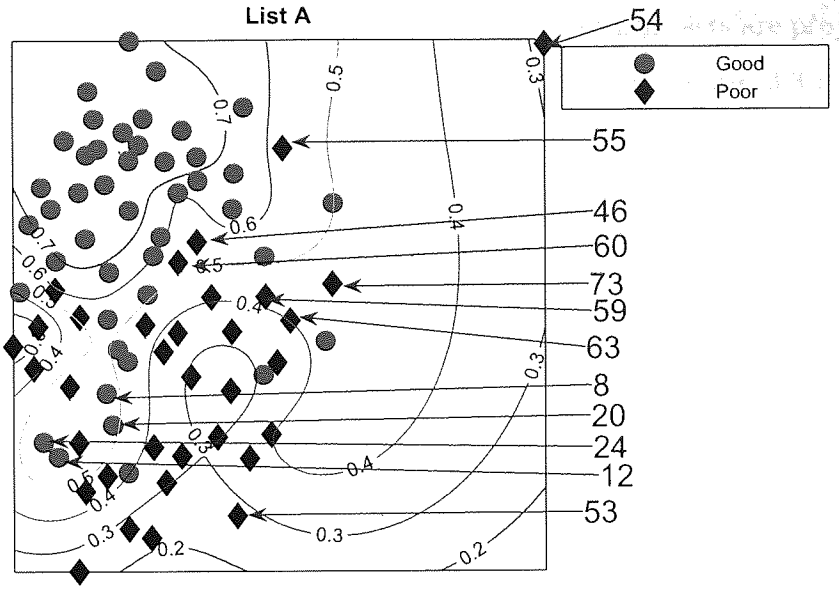
First, the NeuroScale model was applied to the 70 dimensional input vectors corresponding to the individual patient PGL to give the 2 dimensional projection. The number of centres is 77, which is as many as data points excluding one for the bias and the activation function was the 'thin plate spline', $(r^2 \log(r))$. The advantage of this method, as stated in the previous chapter, is the model once trained can be reused without reconstructing the projection over the extended data by just passing the new data points, $\mathbf{x}_{new}$ through the transformation function. $\mathbf{y}_{new} = f(\mathbf{x}_{new}, \mathbf{W})$. Figure 3.1(a) is the result from a 2-dimensional NeuroScale projection using List A and Figure 3.1(b)

is the result using List B. The group of poor-prognosis and good-prognosis patients are labelled differently, with black diamonds and grey circles respectively although this class information was not used in the NeuroScale projection model. The results both show some separation between the two groups of patients with a few patients wrongly mapped into the opposite class. This projection appears to support the previous result of van't Veer et. al. in that List A appears to have some discriminatory capability, although it is evident from these figures that any discrimination is on a graded and overlapping scale rather than providing completely separable distributions.

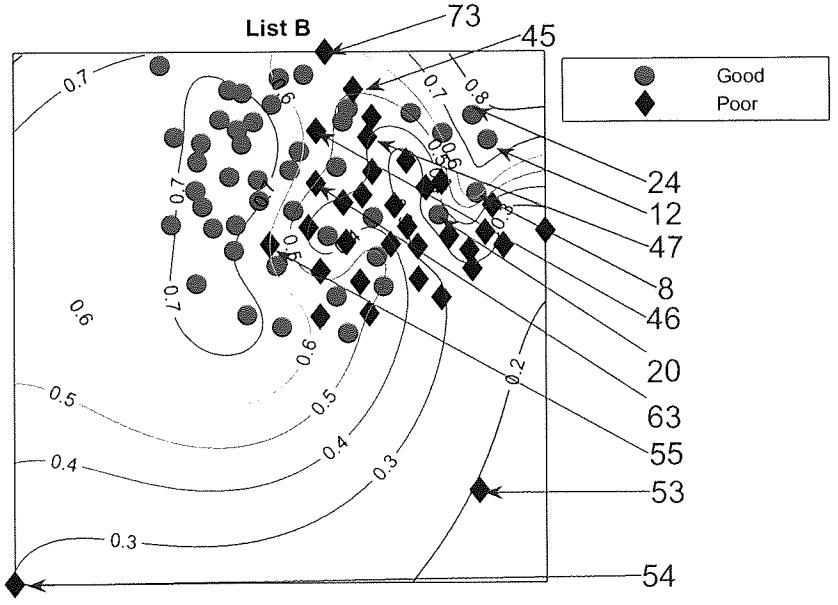
The prognosis indicator values as given on the contour lines, vary on the level of overlap marked between the two prognosis groups. The areas where there is large overlap between the two patient groups reflects ambiguity of any likely class membership. Therefore, we regard these patients as low confidence samples as far as determining class information and we should regard them as 'unclassifiable'. Ignoring those data and regarding those intrinsic unclassifiable data as one of the correctly classifiable results may give the wrong impression and given on overly optimistic impression of the prediction to that data set [69].

Both gene lists show equivalent performance in separating the two patient groups. A detailed analysis will be given in the next section.

As in Figure 3.1(a), there are some patients that are projected to the wrong class, however if only high confidence patients are used for consideration the classification result increases to 100% accuracy. The classification matrices can be found in the Appendix B. There are only 30 patients who fall in the high confidence regions, less than one half of all patients. Similarly the results using List B , give the same classification rate as using list A but with 4 fewer high confidence patients. The results of the projection maps and the classifications support the lack of uniqueness of a single PGL since there is no clear separability or global distribution between the results of List A or List B.



(a) The NeuroScale map of gene List A.



(b) The NeuroScale map of gene List B.

Figure 3.1: NeuroScale projections of Lists A and B. Note the approximate separation of the centroid between poor (diamonds) and good (circles) prognosis groups. Specific individual patients are highlighted with arrows. Contour lines show the likelihood values of belonging to the good prognosis class as determined by the classification models.

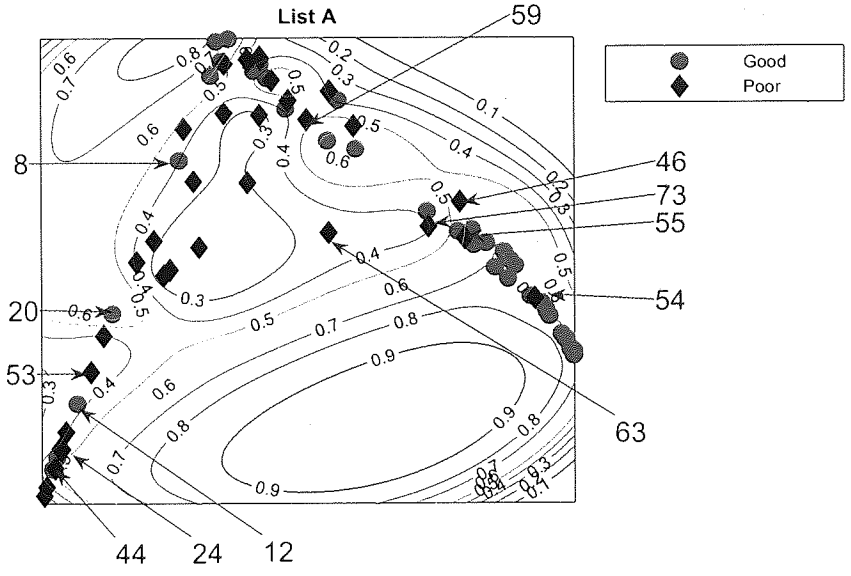
3.2.3 Locally Linear Embedding

The Locally Linear Embedding results using two different gene sets are projected down and shown in Figures 3.2(a) and 3.2(b) using $K = 5$ and in Figures 3.3(a) and 3.3(b) using $K = 20$ together with the classification contour lines of the good prognosis indicator. K is the number of neighbours used to construct the mapping which has to be chosen empirically. With $K = 5$, within a good prognosis cluster of both gene lists, there are four obvious poor prognosis patients. Three of them are common across both list projections. These four patients remain in the wrong place even after the number of neighbours increases. Between 13 and 16 poor prognosis patients are likely to be misclassified as good prognosis patients in the 'boundary layer'.

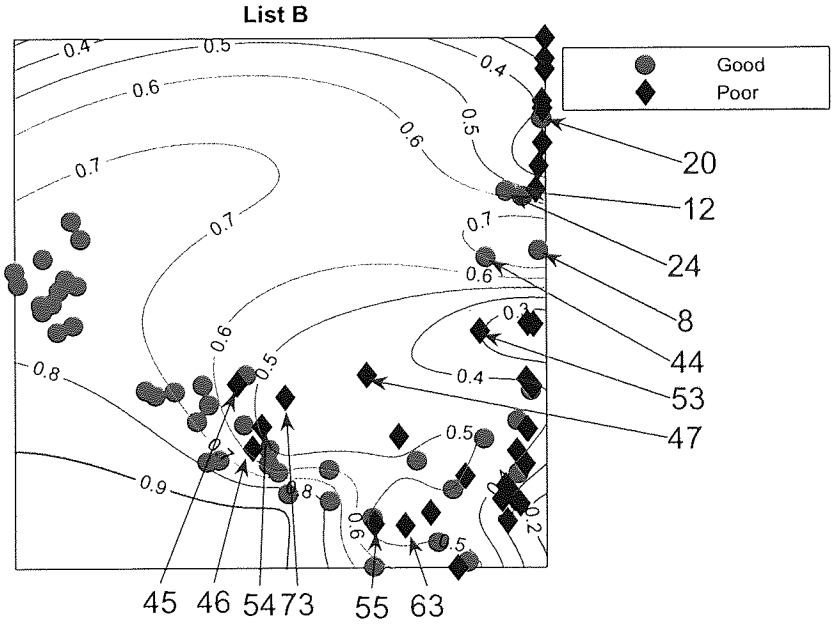
The best representation seems to be $K = 20$ with List A giving slightly better separation with fewer patients misclassified, with 7 good prognosis and 4 poor prognosis patients likely to be misclassified, from inspection of the figures without the classification results. However, some regions can be classified better when the classifier is trained on the particular data set. For example, a few good prognosis patients in Figure 3.3(a) are on the right of the projection while most of the good prognosis patients are supposed to be on the left side. Those few patients create the region where patients are likely to be assessed as good prognosis even though this could be the result of these few outliers.

Both List projections have a separability of the modes of the two groups even though some patients appear in the wrong relative positions for their prognosis groups. Nevertheless, the difficulty for LLE is choosing the appropriate value for K. The result shows better separation of the training data with $K = 20$. For Figure 3.3(a), poor prognosis patients $P45$, $P55$, $P54$ are isolated from the other patients. However, having these three patients correctly classified could result in poor generalisation across new data. Other than this, the LLE projections reflect some similarities to the NeuroScale projections.

For $K = 5$, visually from Figure 3.2, List B gives a more distinct projection than List A with more clusters of good prognosis patients separated without overlap of many poor prognosis patients. When only high confidence patients are retained, no patients are projected to the wrong class using either gene list although the number of high

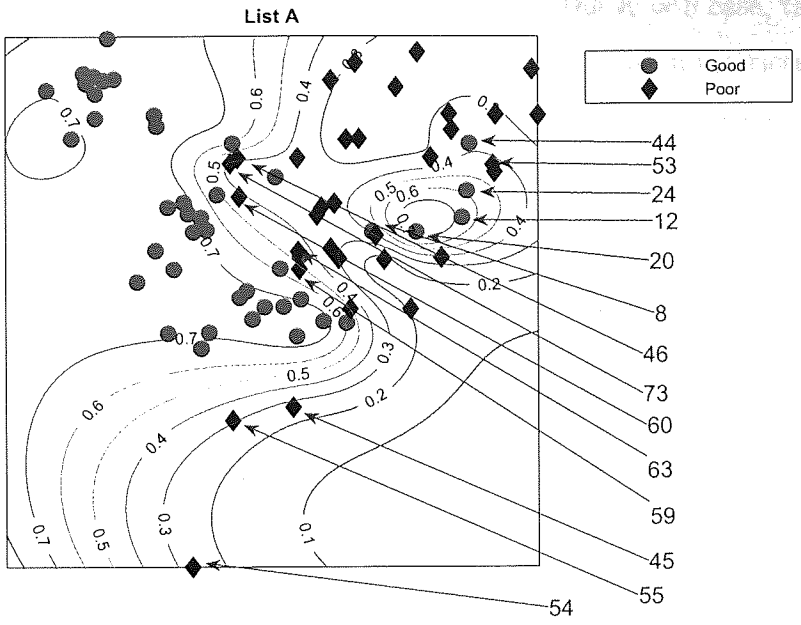


(a) List A.

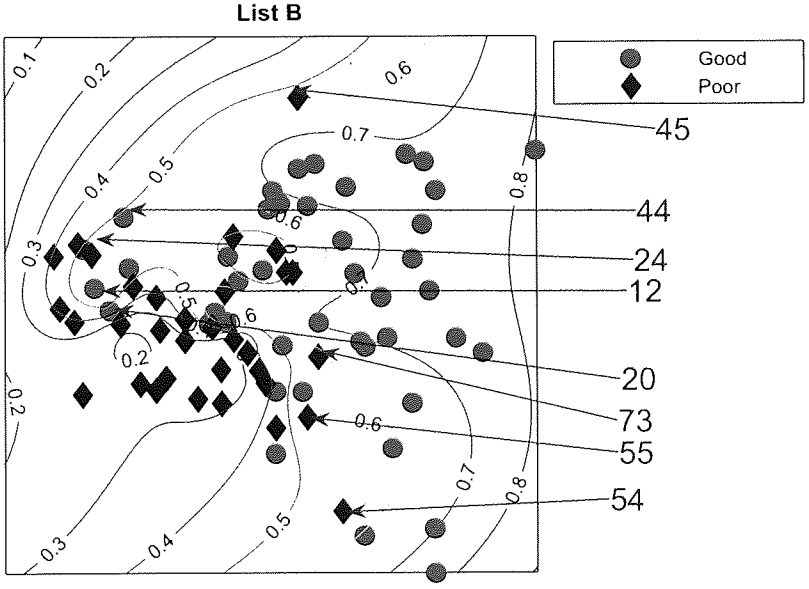


(b) List B.

Figure 3.2: The LLE results with $K = 5$. (a) LLE projection of patients using PGL A. (b) the projections of the same patients but using PGL B. Also superimposed are the contour lines from the classification model.



(a) List A.



(b) List B.

Figure 3.3: The LLE results with $K = 20$. (a) LLE projection of patients using PGL A. (b) the projections of the same patients but using PGL B. Also superimposed are the contour lines from the classification model. With this K, the separation between classes are better to with the $K = 5$ shown in 3.2. P54 is a significant outlier for List A but it is not with List B.

confidence patients using List B is more than using List A by 8 patients.

For $K = 20$ which is shown in Fig 3.3, contrary to the $K = 5$ case, the result using List A with $K = 20$ gives better performance. The classification matrices can be found in Appendix B. The classification rate is quite high with 93.58% accuracy with larger numbers of high confidence patients compared to the other methods, but this could result from the overfitting of this particular model. As can be seen in Figure 5, the gap of the contours between 0.4 to 0.7 is quite narrow. Choosing the exact boundary that determines the prognosis signature of each patient is therefore critical. As a result, if patient values contain uncertain information or noisy data, the resulting classification outcome of such patients is likely to be effectively random. Therefore the data of such uncertain patients should not be taken into account in representing performance results. We investigate the generalisation of these results in the next section.

3.2.4 Generative Topographic Mapping

The Generative Topographic Mapping model uses the Radial Basis Function as in the NeuroScale models. This model needs a specification of the latent shape. The model was trained using the 70 dimensional input vectors corresponding to the individual patient PGL to give the 2 dimensional projection. The number of centres is 77, as many as the data points excluding one for bias and the activation function was the 'thin plate spline', $(r^2 \log(r))$. This method also has the same advantage as NeuroScale as the trained model can be reused easily. The visualisation results used the mean of the mixture model distribution giving the projection data $\mathbf{y}$,

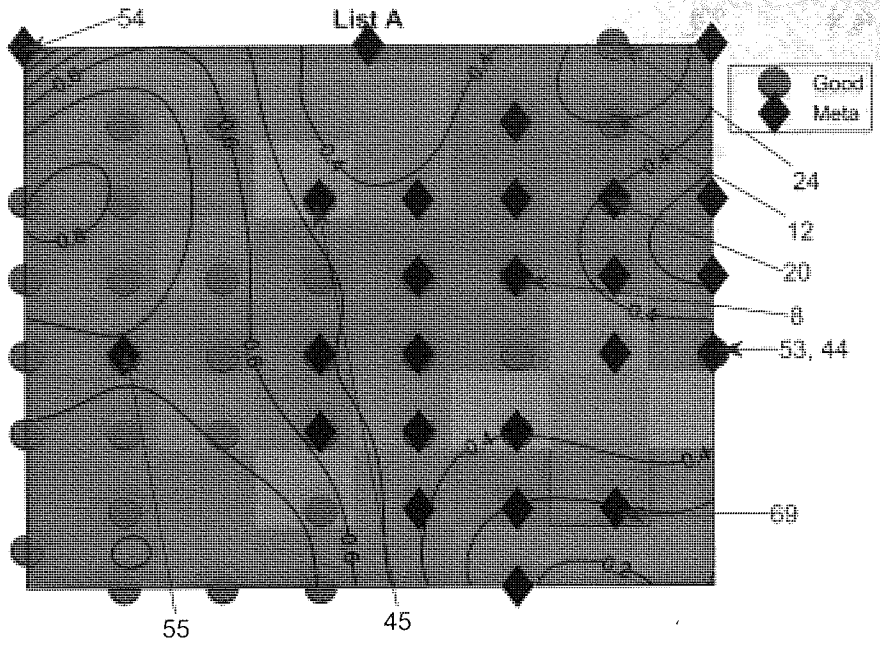
$$\mathbf{y} = \langle \mathbf{t} | \mathbf{x_n}, \mathbf{W}, \sigma \rangle = \sum_{j=1}^{M} R_{jn} \mathbf{t}_j. \tag{3.1}$$

For the new data projection, $R_{j,n}$ is therefore recalculated according to the stored $\mathbf{W}$.

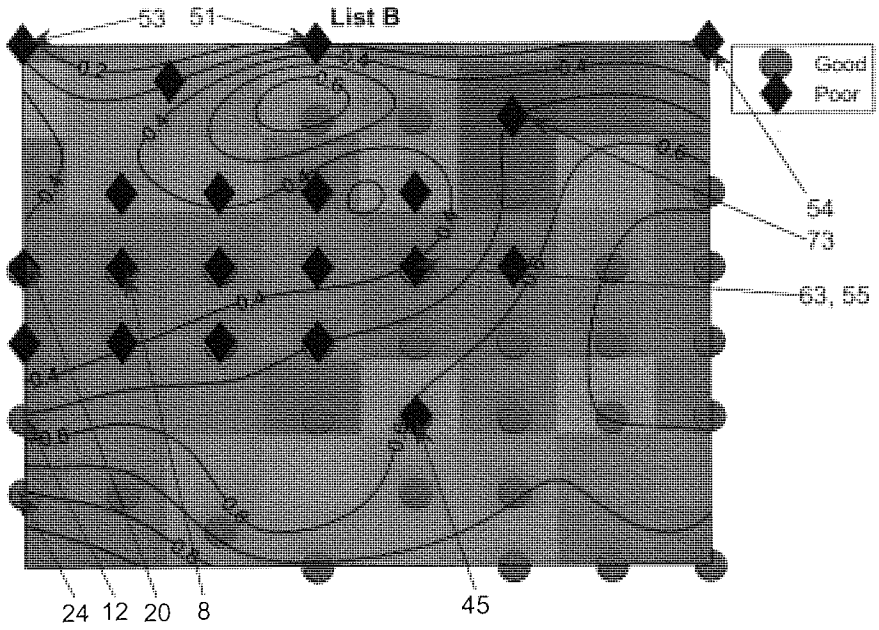
$$\mathbf{y}_{\mathtt{new}} = \langle \mathbf{t} | \mathbf{x}_{\mathtt{new}}, \mathbf{W}, \sigma \rangle = \sum_{j=1}^{M} R_{jn} \mathbf{t}_j. \tag{3.2}$$

Figure 3.4(a) shows the GTM visualisation results using List A, with latent shape 8 which gives a number of latent points that is close to the number of data points. Similarly, the model is also applied to List B. The result of List B is shown in Figure 3.4(b). The contour lines of Figure 3.4(a) using List A, have narrow gaps between prognosis indicators, especially in the middle. Magnification factors are also shown in the background to indicate the strength of the corresponding area to fit the data set. The magnification factor is most area are quite similar, except for the area closed to $P54$, an outlier, where it shows high magnification factor. For comparison, Figure 3.5 shows the GTM results of both List A and List B using the latent shape of 56. The results of List A is similar to the results of using latent shape of 64 but with List B, there are more overlaps of data points with the smaller latent shape. Therefore, we will focus more on the latent shape of 64. From Figure 3.4, fewer patients, in List A, give high confidence compared to Figure 3.4(b). Furthermore, there are many patients projected on top of each other. List B shows better performance, using these 78 patients. No poor prognosis patients are projected to the wrong class. In addition, List B shows correct classification results with high confidence of P44 and P53. However, List A

projects these two patients on top of each other and regards them as poor prognosis with good confidence, even though it is incorrect for P44.

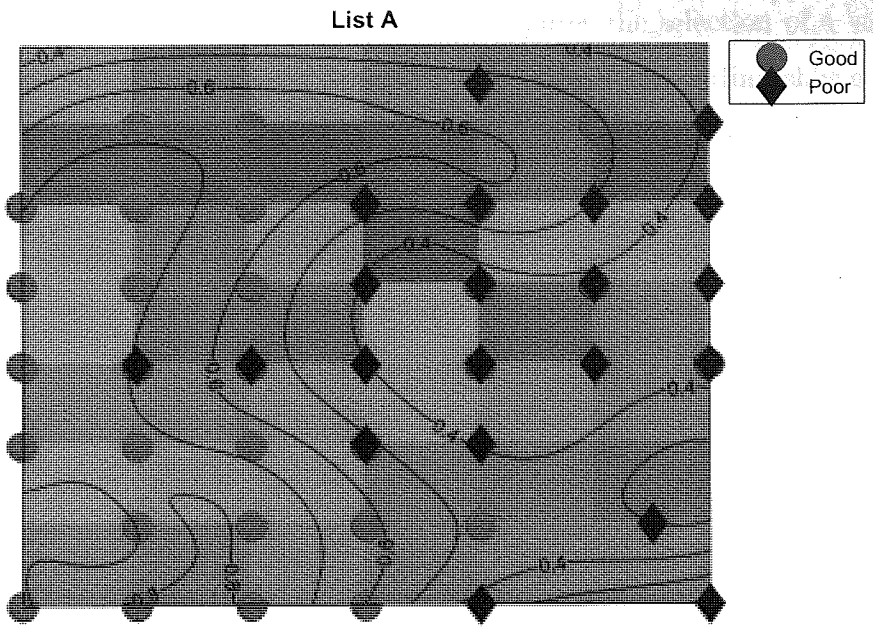


(a) List A.

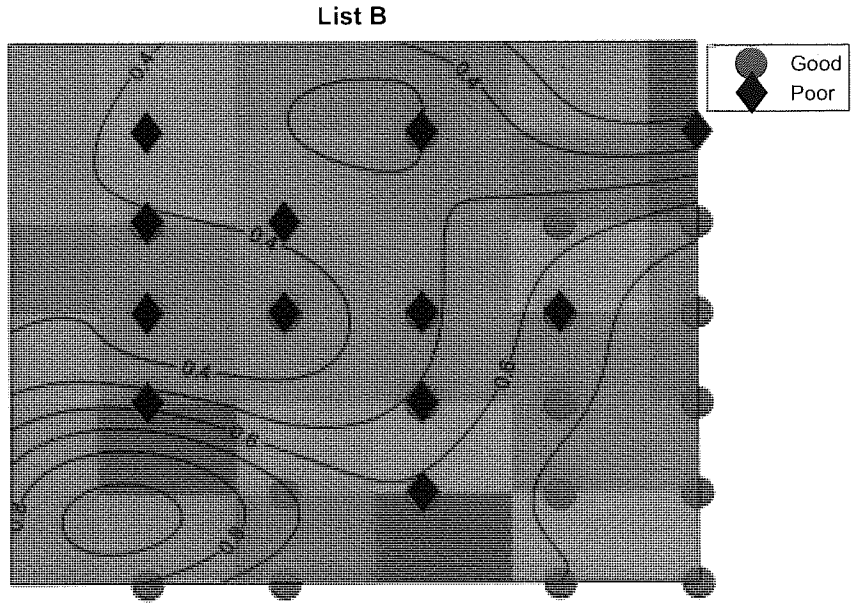


(b) List B.

Figure 3.4: The GTM results with 77 RBF centres and the latent shape is 8 . (a) the GTM projection of patients using PGL A. (b) the projections of the same patients but using PGL B. Also superimposed are the contour lines from the classification model. List A creates small gaps between contour lines and many patients collapse to a single point. Magnification factor is shown in colour. Large magnification factor is more in pink while smaller magnification factor is plotted in blue.



(a) List A.



(b) List B.

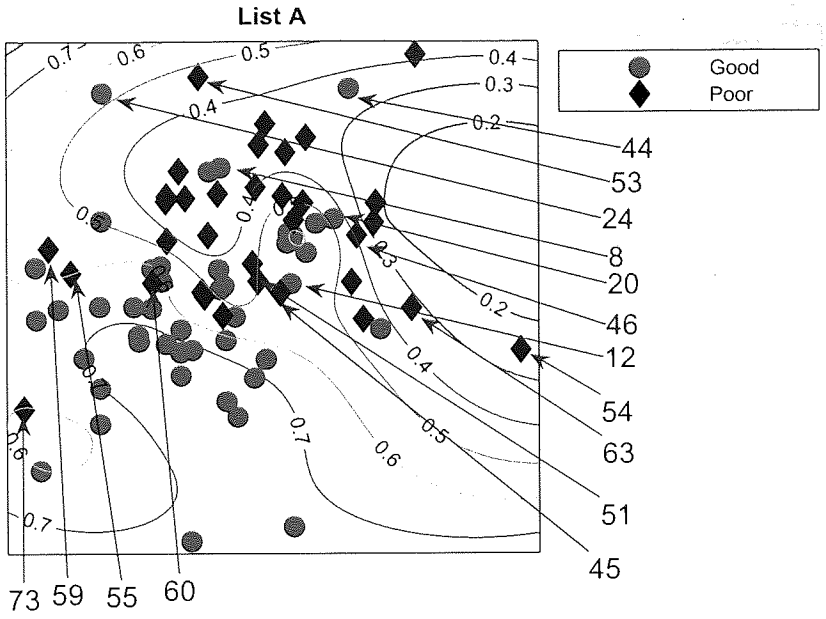
Figure 3.5: The GTM results with 77 RBF centres and the latent shape is 56. (a) the GTM projection of patients using PGL A. (b) the projections of the same patients but using PGL B. This figure is for comparison with the previous GTM projection.

3.2.5 Stochastic Neighbour Embedding

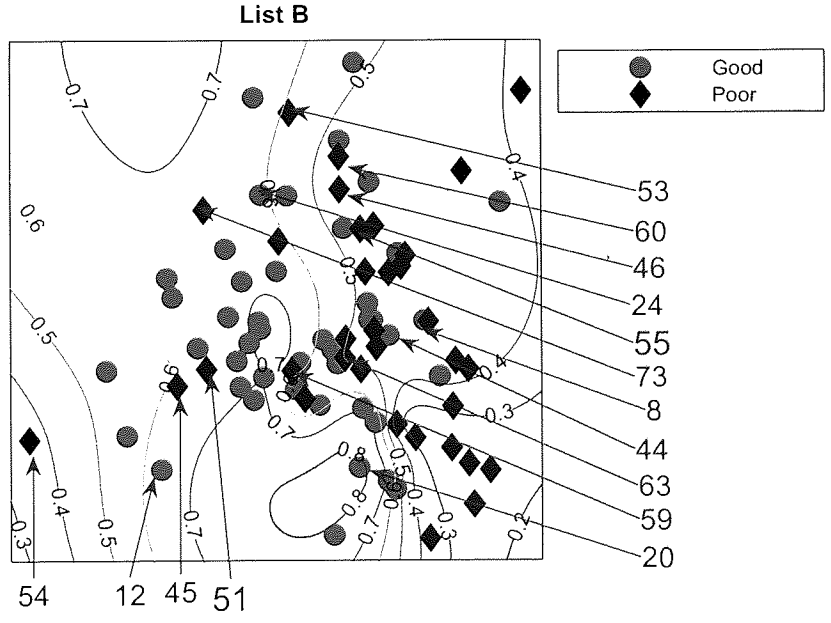
The Stochastic Neighbour Embedding method requires the selection of a 'smoothing' factor σ. The projection maps of Stochastic Neighbour Embedding shown in Figure 3.6 reflect different results for $\sigma = \log(5)$ shown in Figures 3.6(a) and 3.6(b), and $\sigma = \log(20)$ shown in Figures 3.7(a) and 3.7(b). From these figures it can be seen that the relative distributions of the patient projections are quite different for differing choices of the value of σ and it is is quite hard to determine the optimal value of σ. With $\sigma = \log(20)$, patients from both gene groups are mostly overlapped. The separation is not as good as in the previous two models.

Figures 3.6(a) and 3.6(b) show the classification contour lines superimposed on the SNE projection maps using the two different gene lists with $\sigma = \log(5)$, and figures 3.7(a) and 3.7(b) with $\sigma = \log(20)$.

List B gives better overall performance but only when high confidence patients are being measured. No patients are misclassified with almost the same number of high confidence patients using both gene lists. Again, this supports the proposition that equivalent performance can be obtained on dissimilar gene lists. For $\sigma = \log(20)$, both gene lists gave perfect classification rates when restricted to high confidence patients although list A has more high confidence patients (18 patients), compared to 14 patients in list B. Nevertheless, the number of retained high confidence patients in this method is very low.

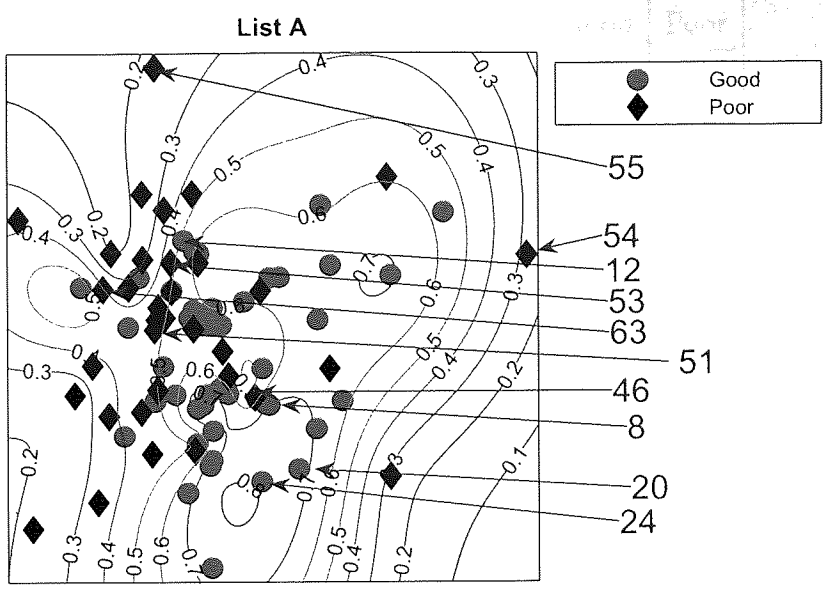


(a) List A.

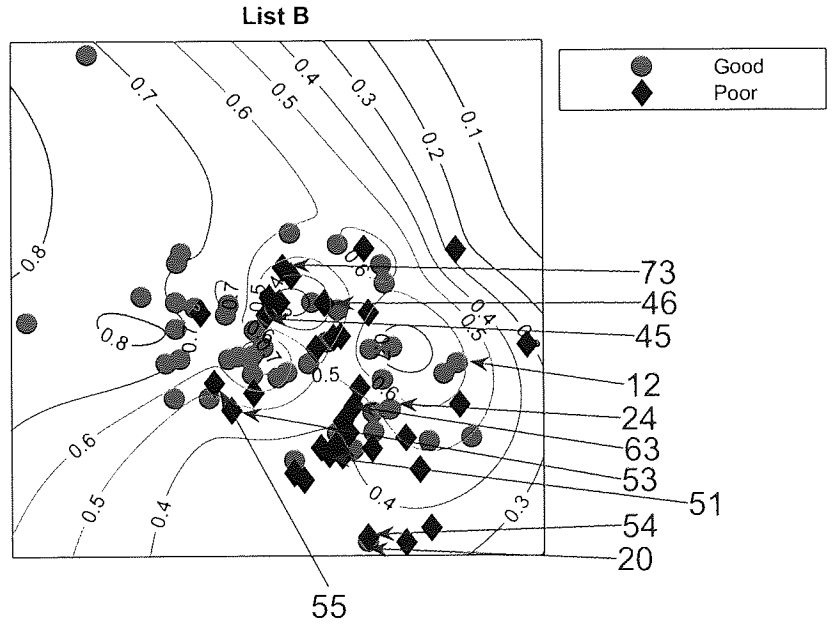


(b) List B.

Figure 3.6: The SNE results with $\sigma = log(5)$. (a) the SNE projection of patients using PGL A. (b) the projections of the same patients but using PGL B. Also superimposed are the contour lines from the classification model. The separation between the two classes is quite poor compared with LLE and NeuroScale.



(a) List A.



(b) List B.

Figure 3.7: The SNE results with $\sigma = log(20)$. (a) the SNE projection of patients using PGL A. (b) the projections of the same patients but using PGL B. Also superimposed are the contour lines from the classification model. The result of SNE shows the sensitivity of σ. In addition P54 which contains missing information is classified with high confidence in this method.

Model used	List A		List B	
	Good	Poor	Good	Poor
NeuroScale	7	6	6	7
LLE($K = 5$)	7	9	3	7
LLE($K = 20$)	1	4	7	5
GTM	4	11	6	10
SNE($\log(\sigma) = 5$)	7	9	9	6
SNE($\log(\sigma) = 20$)	10	6	9	11

Table 3.1: The table shows misclassified patients from all different models from the visualisation results.

3.3 Discussion

3.3.1 Comparison across models

The four methods gave different visualisation outcomes. Table 3.3.1 summarises the misclassification results from all models and parameters.

From the visualisation results, LLE with $K = 5$ represents the data as a quasi 1-dimensional representation while the other three models present 2-dimensional distributions. However on inspection, they reveal some similarities. For both gene sets, poor prognosis patients whose gene feature vectors are significantly placed in the wrong cluster are consistently misplaced across models. For LLE, there are 4 poor prognosis patients (in both gene sets) who are projected to the wrong cluster. Recall that the feature sets used are almost non-overlapping. These patients may be used to compare between models. For List A, instead of $P73$ using LLE, $P60$ has a low confidence and is more likely to be misclassified. Only NeuroScale places $P54$ far from the remaining patients. We note that $P54$ is exceptional in that the patient's gene list has several missing values (and for this reason was eliminated from analysis in the paper by Ein-Dor [24]). The NeuroScale model correctly identifies $P54$ as an outlier patient requiring further investigation. Note that a classification model built from this projection would place $P54$ into a good or poor prognosis class despite the missing information.

With $K = 20$ in LLE, three patients, $P54$, $P55$, $P45$, are separated from the remaining patients, instead of clustering amongst the other good-prognosis patients as

indicated by the other projection models. With this number of neighbours in LLE, the projection is giving better patient group separability despite having outliers.

The GTM models do not obviously show the outliers. Even though $P54$ is an outlier and projected far from the remaining patients, the projection is not so obvious as in LLE and NeuroScale because the visualisation created by GTM has been controlled by the shape of latent points. In addition, many patients are projected on top of each other. However, this model gave very small misclassification of good prognosis patients, none for List B.

For the SNE projective visualisation, $P54$ has a surprisingly high confidence of being correctly classified and does not reflect the problems of missing information. Instead of patient $P54$, $P59$ is misclassified by the SNE projection but with low confidence. On the other hand the likely misclassified good prognosis patients are common using both LLE and NeuroScale but with some slight difference to the SNE for which $P12$ does not significantly project to the wrong cluster.

In addition, for List B, all of LLE, shown in Fig 3.3(b), GTM, shown in Fig 3.4(b), and NeuroScale, shown in 3.1(b), give similarly consistent results for projections of good and poor prognosis patients into the incorrect groups as shown in both visualisation and classification results. The difference using SNE is that SNE gives a better representation of $P46$ but gives an incorrect projection to $P51$ instead. Similarly, the significantly misclassified good prognosis patients are the same using both LLE and NeuroScale but it is very difficult to discriminate using Stochastic Neighbour Embedding.

Both LLE and SNE show sensitivity of projections to empirical choices of selectable parameters, but projections can be found with some consistency of patient distribution across all four nonlinear topographic projection methods. However, NeuroScale and GTM have an advantage over the other two methods because of their principled basis in a machine learning parameterised mapping that can be reused in a generalisation experiment without the need to retrain any models. We will validate this feature in the next section.

3.3.2 Comparison of PGLs across patient groups

Both gene lists A and B produce similar projections, except that $P55$ in List A is mostly projected to the wrong group by all four models. However, with List B it is not as unambiguous since this patient is projected into the interface between the two prognosis groups. $P45$ appears as one of the wrongly projected poor prognosis patients instead of $P55$. Except for these two patients with different results, both gene lists create similar projections, despite the fact that both gene sets have very few genes in common.

This supports the opposing view, that different gene lists can be created from small sample patient groups which randomly correlate with arbitrary outcome. The classification results, which can be seen in the Appendix B, confirm the similarity using both gene lists.

From the observations, most patients have similar representations in the projected mappings. Some patients are better represented by one PGL or the other. Nevertheless both gene lists produce an overlapping region in which patients in this area cannot be separated into either good or poor prognosis groups. The clinical prognosis label of these patients should be *unclassifiable* instead of being assigned into any one prognosis group. This new patient type can be crucial in the medical domain where the advice to the clinician should be that no prediction can be made on the available information and extra information is needed in addition to the gene expression profile.

Additionally, there are some possibilities for identifying the disease pathogenesis from the visualisation results by looking at high certain patients which can be separated from the remaining patients and searching for the similarity in the gene expression among those patients. Nevertheless, to retrieve reliable underlying structure of the data requires sufficient numbers of data samples.

3.4 The validation set

A follow-up study [89] from the van't Veer group was performed which marked the progression of another set of patients to verify the original study. This follow-up data set, which we refer to as the van de Vijver data, contains 295 patients with 106 poor-

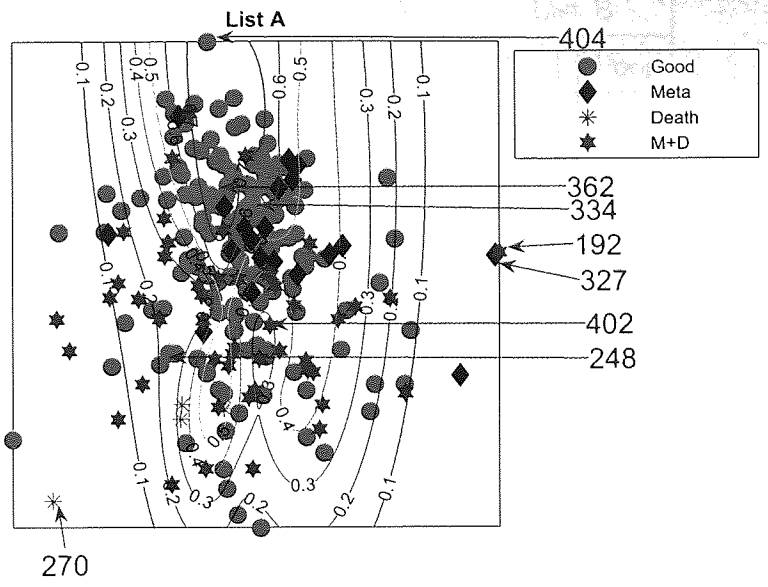
prognosis and 189 good-prognosis patients. The poor-prognosis patients are categorised into 3 sub-categories: patients with metastasis who did not die as a direct result of the metastasis; patients who died without developing metastasis; and the patients who developed metastasis and eventually died. Within these 295 patients, 61 of them are present in the original study. Those patients were removed in this chapter to ensure we can examine generalisation on a separate set of 234 different patients, 159 of whom are categorised as good-prognosis patients.

In the earlier studies of van't Veer et. al., validation of the original 70 PGL was made by obtaining good performance on an additional patient data set. In this section we perform the same comparison, making use of the functional mapping ability of NeuroScale applied to the extended new data.

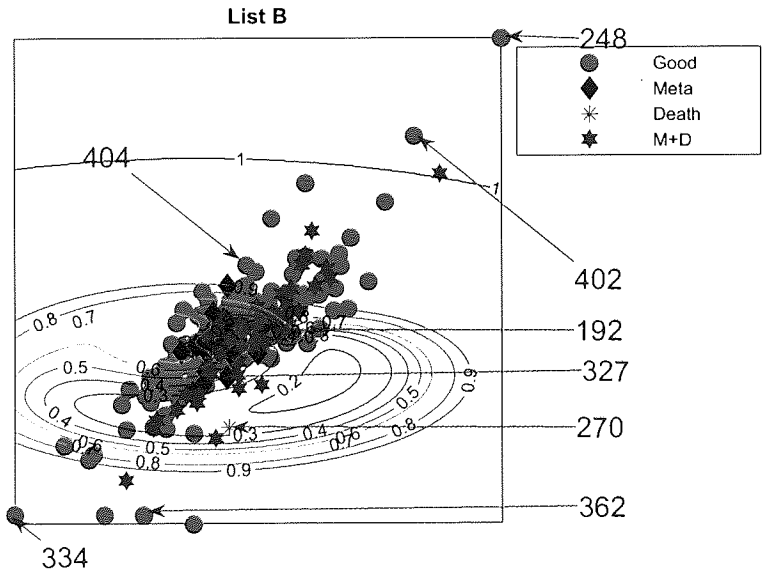
NeuroScale and GTM have an advantage over some other topographic models in that new data can be projected through a prior learned projection mapping. Once the functional mapping has been obtained using NeuroScale or GTM, the model can be reused without reconstructing the projection using novel data. For additional comparison, this section will also use a retrained LLE model (trained on the new full data set) with $K = 20$ which performed well with the previous patient set, as a comparison to NeuroScale and GTM.

For validation, the new patient set of van de Vijver will be projected using the same networks for both gene lists. In their study [89], they verified the viability of their 70 nominated genes as prognosis indicators of breast cancer by using the previous 78 patients as a training set and testing on this new patient set. We will investigate this claim further by applying trained topographic visualisation to this new patient set using both PGL lists A and B, and observe the consistency of the distribution between the two groups of patients. The classification results of the validation set are shown in table 3.4.

Figures 3.8(a) and 3.8(b) show the projections of those remaining 234 patients, labelled into 4 different groups, which are (1) good-prognosis patients (circles), (2) metastasis patients (asterisks), (3) death (stars) and (4) both metastasis and death (diamonds). Both PGL projective visualisations seem to give similar projections on the new data. For List A from Figure 3.1(a), good prognosis patients are likely to be



(a) List A.



(b) List B.

Figure 3.8: The NeuroScale visualisation projection of the new 234 patients trained using the original 78 patients based on List A and List B. Circles represent healthy patients who did not develop any further sign of relapse, asterisks for patients who developed metastasis but did not die, diamonds for the patients who died without developing metastasising cancer and stars for patients who developed metastasis and then died consequently. The contour lines from the previous 78 patient projections are superimposed on the visualisation results of this validation set of both GTM and NeuroScale.The overall performance using the new patient set is dramatically reduced. Both figures show a different projection of $P192$ and $P327$ where they are at the edge with List A but they are in the central region using List B.

Model used	List A		List B	
	Good	Poor	Good	Poor
NeuroScale	19	77	43	59
GTM	15	80	11	111

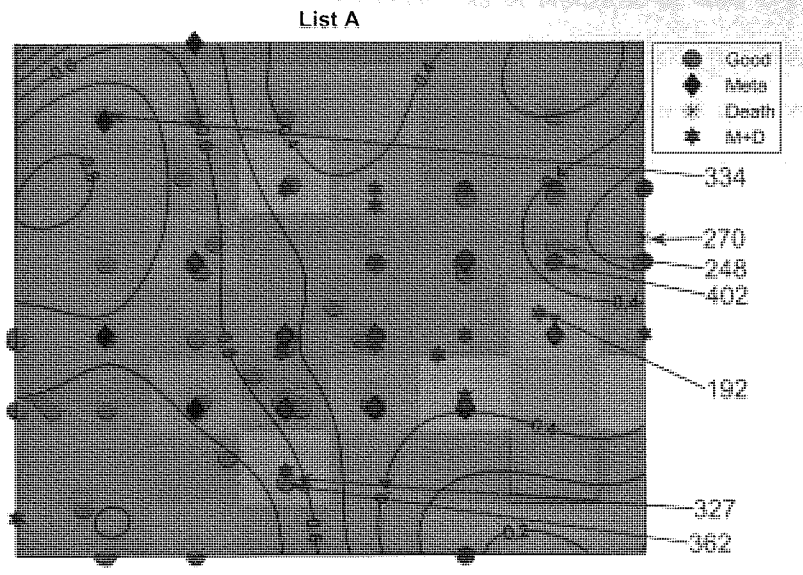
Table 3.2: The table shows misclassified patients of the validation set using the RBF network trained on the original 78 patient set.

projected towards the top of the visualisation map while the poor prognosis patients are at the bottom. The projection of new data for List A in Figure 3.8(a) shows some density of good-prognosis patients on the top of the visualisation map, similar to Figure 3.1(a). However, many good-prognosis patients are distributed across the visualisation map. For List B, the patient gene vectors are quite dense in Figure 3.8(b); however a number of good-prognosis patients tend to be projected on the top left of the plot, similar to Figure 3.1(b).

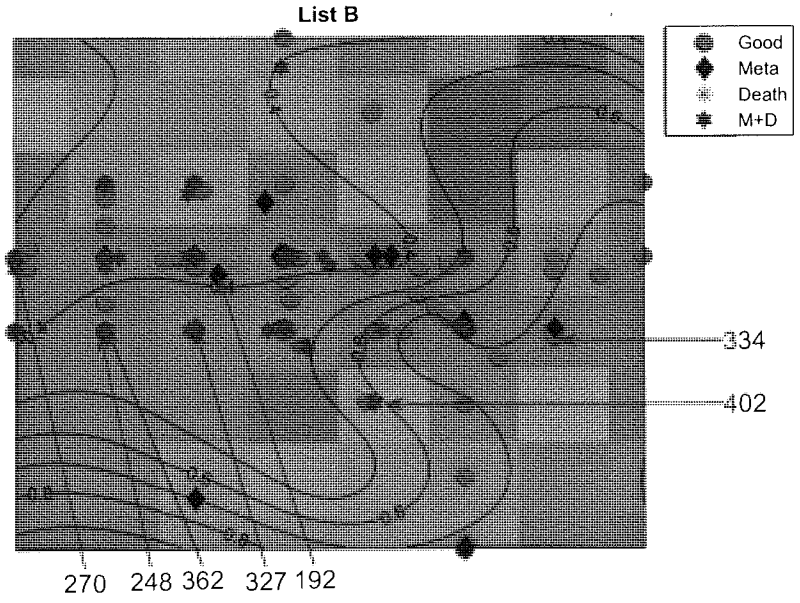
Similarly, the GTM model has the same advantage of being capable of extension to the new data set. Figure 3.9(b) and 3.9(a) shows the visualisation results using GTM with the 234 patients. Only 44 patients are high confidence using List A and all of them are projected to be good prognosis. No poor prognosis patients have high confidence. Even though in the 44 high confidence patients, only 1 patient is in the wrong class (which is a high classification rate), the model does not give advantages over other models. By contrast, List B gives a poor result but it improves when only high confidence patients are considered. Nevertheless, the size of high confidence regions are quite small compared to the NeuroScale model. More patients have high confidence using NeuroScale. However, similar to the NeuroScale, more good prognosis patients are likely to be separated from the poor prognosis patients. However, the separation is not as good as NeuroScale.

For comparison, LLE was *retrained* on the full set of 234 patients and also projected down. The results are shown in Figures 3.10(a) and 3.10(b) using $K = 20$. These new projections of good or poor prognosis patient groupings have little resemblance to the previous results obtained on the original training set.

There is no separation between the two prognosis groups using the LLE model, which shows the poor consistency between the chosen K for the previous patient set



(a) List A.



(b) List B.

Figure 3.9: The GTM results with the latent shape 8 of the new 234 patients trained using the original 78 patients based on List A and List B. Circles represent healthy patients who did not develop any further sign of relapse, asterisks for patients who developed metastasis but did not die, diamonds for the patients who died without developing metastasising cancer and stars for patients who developed metastasis and then died consequently. (a) denotes the GTM projection of patients using PGL A. (b) denotes the projections of the same patients but using PGL B. Also superimposed are the contour lines from the classification model.

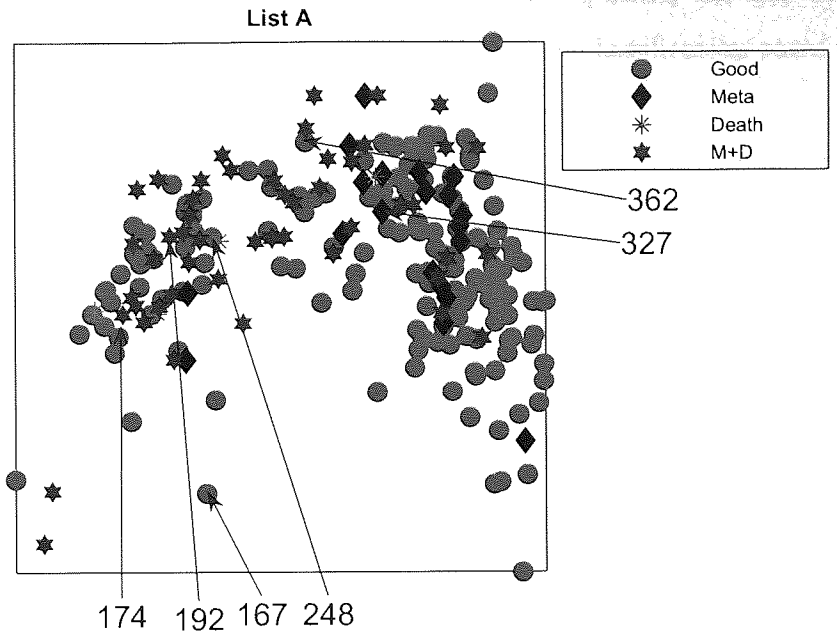
and the new patient set. The pre-trained networks of NeuroScale and GTM, provides a more separable projection map. Nevertheless, all visualisation models of the new patient set create large sections of overlap between the two patient groups.

The two different gene lists give disparate views. Many poor prognosis patients are clustered in the middle while the good prognosis patients are likely to be more widely distributed. This is especially true for List B.

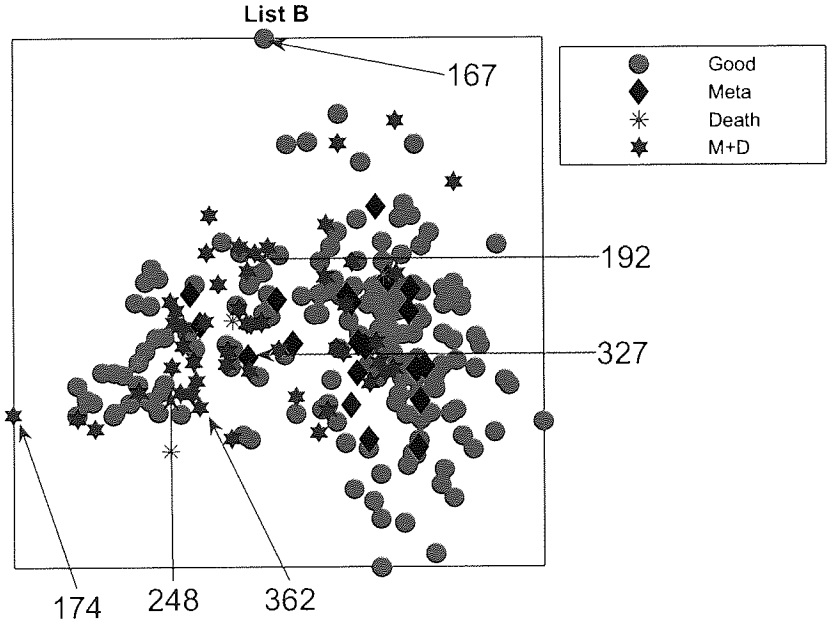
Some patients give different projections: $P192$ and $P327$ are at the right edge of the visualisation projection using List A in Figure 3.8(a) which would therefore give very high confidence of being poor prognosis patients while List B projects them down into the central regions which gives lower confidence of being poor prognosis patients. Alternatively many good-prognosis patients are projected to the central regions using List A but are scattered around the edges in Figure 3.8(b) using list B: for example $P362$, $P248$ give very high confidence of being good prognosis patients while list A does not give such high confidence of $P362$ and in addition, misclassifies $P248$. However, the overall separation between the two patient groups can be seen to be significantly worse for this set of generalisation patients when compared to the original patient set used in the training phase.

These results indicate that the selected 70 genes are not representative of robust Predictive Gene Lists for prognosis on this problem domain for these patients. The 70 genes extracted from the original data set perform well only on particular patients due to the random correlation effect occurring because of the large dimensionality of the original $25,000$ genes and the small patient sample size. The two different lists of 70 genes perform equivalently in the projection mapping showing the non-uniqueness of the selected gene list. Only parts of the map may be used to identify the good-prognosis patient cluster with any degree of confidence. However, the broad overlap demonstrated in this chapter indicates that most patients should be considered as *unclassifiable* based on these subset PGLs.

In the van de Vijver study, the generalisation performance from the van't Veer data set is mentioned. However, the publication only implemented a Kaplan-Meier analysis which shows the good surviving rate outcome of patients predicted as good prognosis patients. However, they did not show the random outcome resulting from the patients



(a) List A.



(b) List B.

Figure 3.10: The LLE projections of the new 234 patients using List A. The LLE Visualisation projection with $K = 20$ of the new 234 patient. Because LLE is trained on the new 234 patients, the contour lines can not be reused from the previous projection. No contour lines are shown here. Circles represent healthy patients who did not develop any further sign of relapse, asterisks for patients who developed metastasis but did not die, diamonds for the patients who died without developing metastasising cancer and stars for patients who developed metastasis and then died consequently. (a) is using PGL A. (b) is using PGL B.

predicted as poor prognosis signatures. However, this patient list also contains most of the original 78 patients which obviously makes the classification rates higher than expected for a genuine generalisation performance test.

Furthermore, a definite threshold that divides the good and poor prognosis signatures was specified. Therefore, patients who are intrinsically unclassifiable are also included in the classification results which is not advisable.

3.5 Conclusion

Similar to the Ein-Dor [24, 25] study, our visualisation results using different approaches show a consistency of the projections from the two different gene lists which are almost orthogonal to each other. They give comparative separability between the two patient groups of the van't Veer data sample. However, the separation between two prognosis groups dropped dramatically when the visualisation method is applied to a new set of patients. We found that the gene list which yielded good biomarkers of the breast cancer patients in the van't Veer study does not provide as good separability as they previously suggested. The gene list gave reasonable separability of patient types in the preliminary experiments but does not separate patient groups of the later study. In addition, the overlap between patient groups is large and can lead to misleading prognoses, which indicates that using a small number of patient samples to identify gene markers generically yields unreliable results.

These patients have intrinsic *uncertainty* attached to their gene expression profiles and should not be used in prospective studies without further investigation. An overoptimistic intepretation of the results is likely to be made if uncertainty issues in the data are ignored. This is one example of the fact that in real-world data analysis, intrinsic uncertainty is commonly ignored. It is very important to take into account all the possible uncertainty and incorporate it into the data modelling process before making any important analysis and decision, especially in the biomedical field which can risk the loss of lives. One suggestion which has been given in this chapter is to incorporate the prognosis indicator together with the results rather than giving the outcome as precisely good or poor. Since we have argued that uncertainty should be

accounted for within the modelling process itself, in the next chapter we introduce an extended visualisation model which incorporates uncertainty by being probabilistic.

Chapter 4

Modified GTM

4.1 Introduction

This chapter will propose a method for modifying the GTM model in order to be capable of incorporating uncertainty information. The Generative Topographic Mapping (GTM) [9] visualisation model, as discussed in chapter 2 was introduced as a non-linear transformation from latent space to data space. It can usefully apply to data visualisation applications, by defining a projection from higher dimensional data space onto a two-dimensional visualisation space.

4.2 The Modified GTM Model

The GTM can be modified to incorporate uncertainty information. According to the standard GTM, the distribution of the high-dimensional data t conditioned on latent variables x is given by

$$p(t|x, W, \sigma) = \frac{1}{(2\pi\sigma^2)^{l/2}} \{\exp\left\{-\frac{||y(x;W) - t||^2}{2\sigma^2}\right\}\}, \tag{4.1}$$

where l is the dimensionality, $y(x;W)$ is a function which maps x to t and is parameterised by the matrix W which controls complexity. In the standard GTM, σ^2 in (4.1) refers to a noise variance arising from the inaccuracy of the mapping from the latent space to the data space. This variance is the same for all the data points. However, in our microarray application, every gene expression measurement has a different confi-

dence value attached to it. In other words, data points in higher dimensional space have different variances. Therefore, the number of variances (from the confidence values) should be equal to the number of data points.

Since in our modified GTM model, variances depend on data points t, we shall write the new log likelihood, as:

$$\mathcal{L}(W, \sigma_n) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{M} \sum_{j=1}^{M} p(t_n|x_j, W, \sigma_n) \right\}. \tag{4.2}$$

By choosing an RBF for the function $y(x; W)$, we shall write

$$y(x; W) = W\phi(x), \tag{4.3}$$

where the elements of $\phi(x)$ consist of K fixed basis functions of $\phi_i(x)$, and W is a l x K matrix.

In the EM algorithm, the E-step remains the same as the standard GTM (as in Chapter 2).

$$\begin{aligned} R_{jn}^{(m)}(W^{(m)}) &= P^{(m)}(j|t_n, W^{(m)}) \\ &= \frac{p(t_n|x_j, W^{(m)})}{\sum_{j'=1}^{M} p(t_n|x_{j'}, W^{(m)})}. \end{aligned} \tag{4.4}$$

However, the M-step is different. From the complete-data log likelihood:

$$\langle \mathcal{L}_{comp}(W) \rangle = \sum_{n=1}^{N} \sum_{j=1}^{M} R_{jn}^{(m)}(W^{(m)}) \ln\{p(t_n|x_j, W)\}, \tag{4.5}$$

maximising the expectation of the complete-data log likelihood (4.5) with respect to W gives:

$$\sum_{n=1}^{N} \sum_{j=1}^{M} R_{jn}^{(m)}(W^{(m)}) \frac{\|W^{(m+1)}\phi(x_j) - t_n\|}{\sigma_n^2} \phi^T(x_j) = 0 \tag{4.6}$$

Solving the above equation, we get

$$\Phi^T G^{(m)} \Phi (W^{(m+1)})^T = \Phi^T R^{(m)} T, \tag{4.7}$$

where Φ is the $M \times K$ RBF design matrix with elements $\Phi_{ji} = \phi_i(x_j)$,

T is the $N \times l$ data matrix,

R is an $M \times N$ responsibility matrix with element $\frac{R_{jn}}{\sigma_n^2}$,

G is an $M \times M$ diagonal matrix with elements $G_{jj} = \sum_{n=1}^{N} \frac{R_{jn}(W)}{\sigma_n^2}$.

The result in (4.7) remains almost the same as the standard GTM. The difference is the definition of R and G matrices which have elements R_{jn} and $G_{jj} = \sum_{n=1}^{N} R_{jn}(W)$ respectively. The responsibility matrix in this equation is divided by the variance and it is interesting that the data with larger variance (low confidence) will be less responsible for the low-dimensional projection. Since in our new model σ_n^2 is dependent on n, it can not be eliminated from the equation as if it was a constant as in the standard GTM model.

In the standard GTM approach, we also need to estimate σ, but in our new model σ_n are given in the form of confidence values. The simplest assumption that we can make is that variances are the inverse of the confidences, so that $\sigma_n^2 = \frac{1}{C_n}$ where C_n is a confidence value of each data point. This can make variance become too large since the variance ranges from 1 to ∞. As a result, the conditional probability in (4.1) could be dominated by the variance only and not the position in the data space which is determined by $||y(x_i; W) - t||^2$. A solution to this problem is to introduce a constant, K, to control the scaling of the variance:

$$\sigma_n^2 = K\sigma_n^{*2}, \tag{4.8}$$

where σ_n^{*2} is directly obtained from the inverse of a confidence, $\frac{1}{C_n}$.

Using (4.8), equation (4.1) becomes

$$p(t|x, W, \sigma^*) = \frac{1}{(2\pi K\sigma^{*2})^{l/2}} \exp\left\{-\frac{||y(x_i; W) - t||^2}{2K\sigma^{*2}}\right\}. \tag{4.9}$$

The value of K can be estimated by using the maximum likelihood approach. Maximising (4.5) but using (4.9) with respect to K gives the following equation to solve:

$$\sum_{n=1}^{N}\sum_{j=1}^{M} R_{jn}^{(m)}(W^m)\left(-\frac{l}{2K} + \frac{1}{2}\frac{\|W\phi(x_j) - t_n\|^2}{K^2\sigma_n^{*2}}\right) = 0. \tag{4.10}$$

Therefore, the re-estimation of K becomes

$$K^{(m+1)} = \frac{1}{Nl}\sum_{n=1}^{N}\sum_{j=1}^{M} R_{jn}^{(m)}(W^{(m)})\frac{\|W^{(m+1)}\phi(x_j) - t_n\|^2}{(\sigma_n^{*(m)})^2}. \tag{4.11}$$

The estimation of K is similar to the estimation of σ in a standard GTM, which is

$$(\sigma^{(m+1)})^2 = \frac{1}{Nl}\sum_{n=1}^{N}\sum_{j=1}^{M} R_{jn}^{(m)}(W^{(m)})\|W^{(m+1)}\phi(x_j) - t_n\|^2. \tag{4.12}$$

If σ_n^{*2} is the same for all the data points, equation (4.11) will be exactly the same as (4.12).

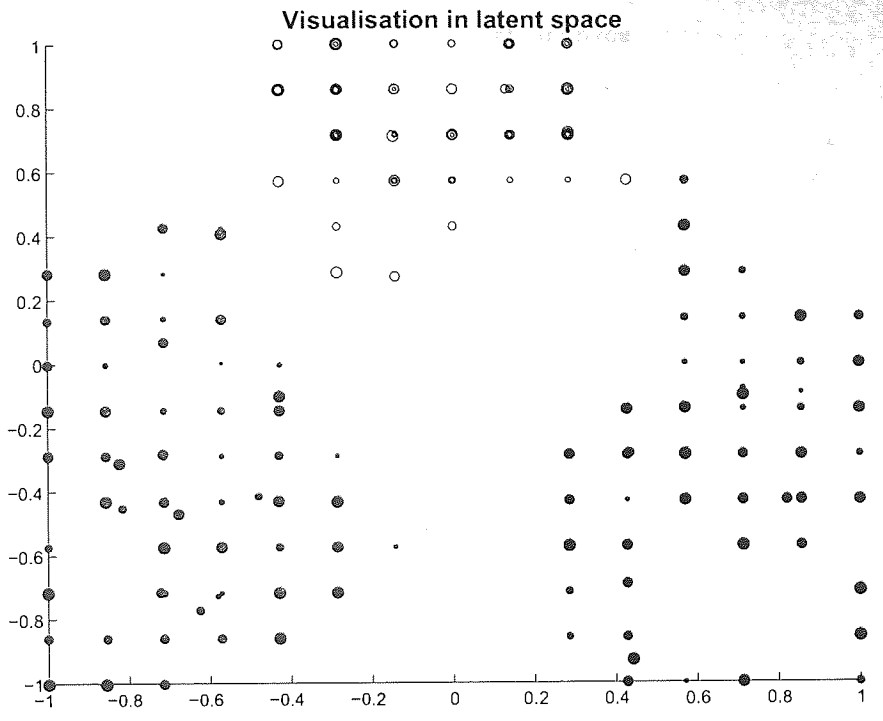
4.3 Illustrations of modified GTM

We demonstrate the abilities of the modified GTM by examining a synthetic control problem of three Gaussian mixture components in a four-dimensional space. The centres of three mixture models are aligned in a triangle. The number of points in each cluster is 100. The estimation of constant variance obtained from the original EM algorithm is σ^2= 0.0557.

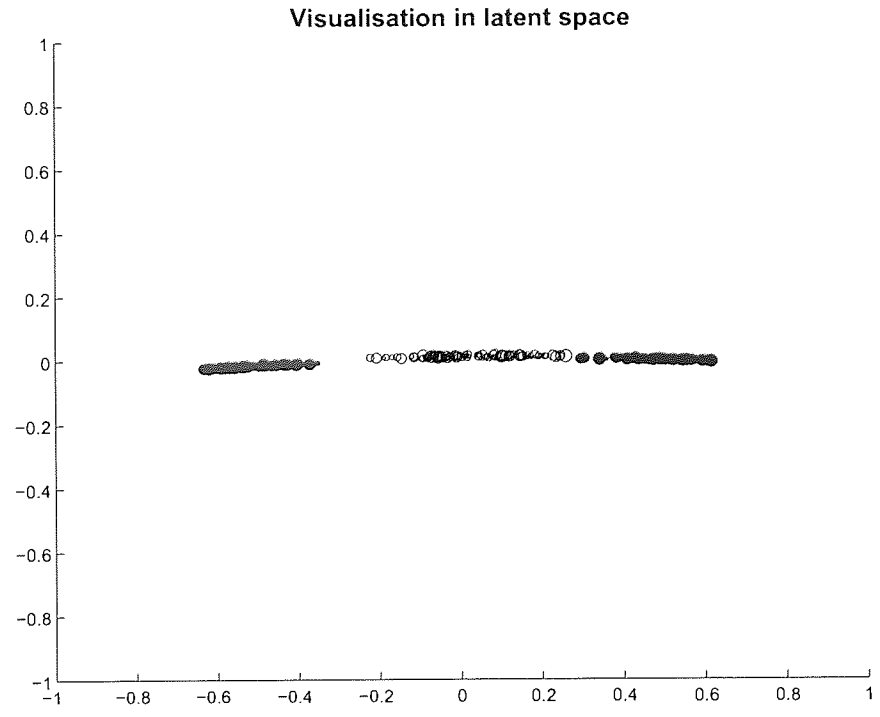
Fixed Variance

The following results were achieved using a fixed variance on all the data points with the modified GTM algorithm. Figure 4.1(a) illustrates the visualisation of data using a variance a lot smaller than the estimation, σ^2= 0.0001. The result has been affected by a numerical problem that the result do not represent the data reasonably well. Figure 4.1(b) is a result using a variance that is too large, i.e. σ^2= 1. The data is clearly separated in three clusters but they are formed in a straight line, not a triangle as it was originally created.

Figure 4.2(a) uses σ^2=0.05 which is similar to the result using the standard GTM in Figure 4.2(b).

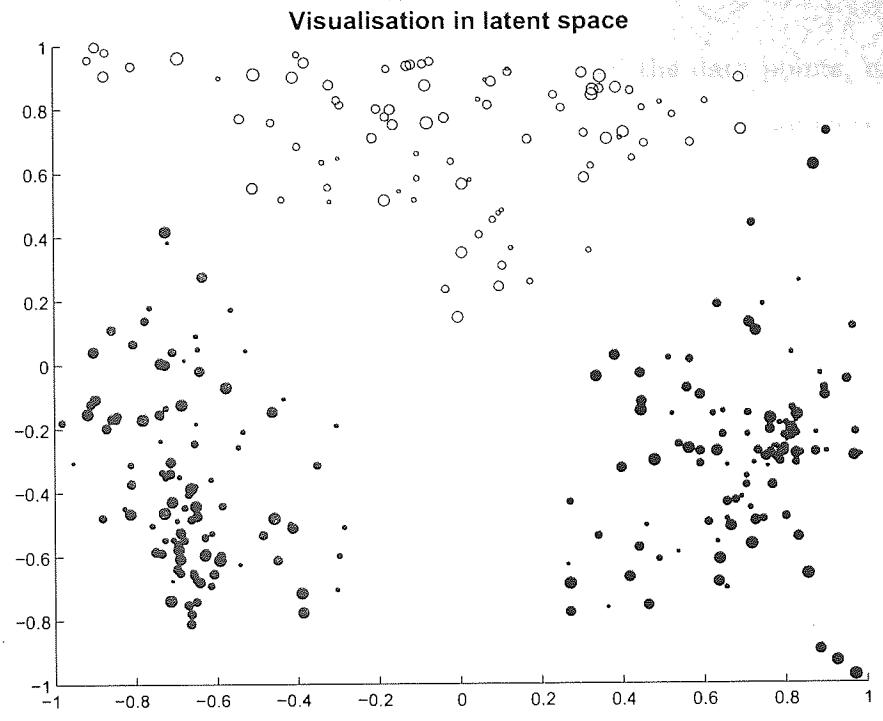


(a)

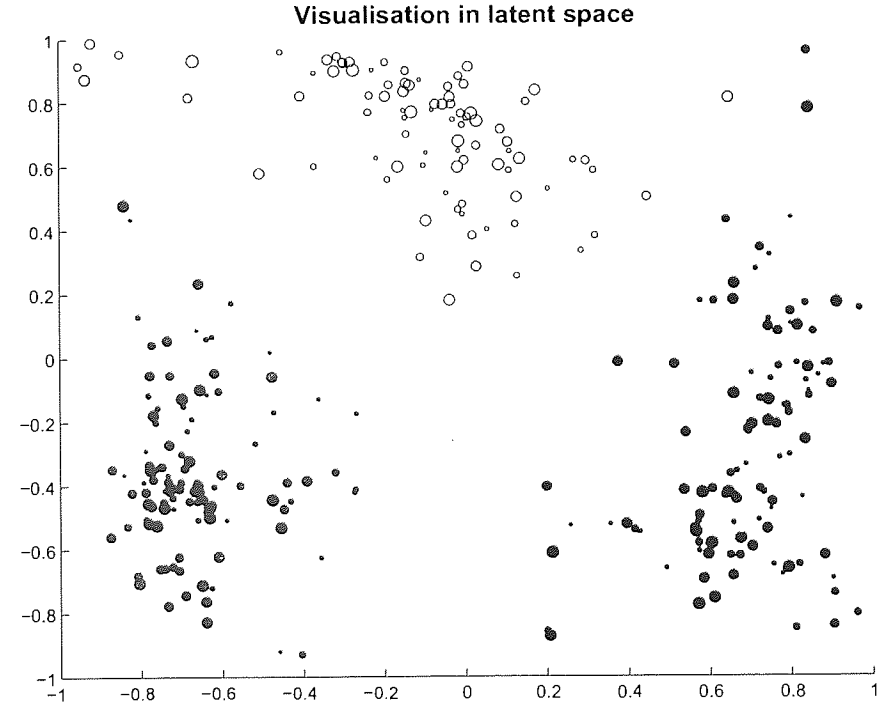


(b)

Figure 4.1: The resulting visualisation using modified GTM. (a) with σ^2=0.001 and (b) with σ^2=1. The data created from different Gaussian centres are colour coded differently. Different sizes of points refer to different levels of variance. Large points correspond to large variance.



(a)

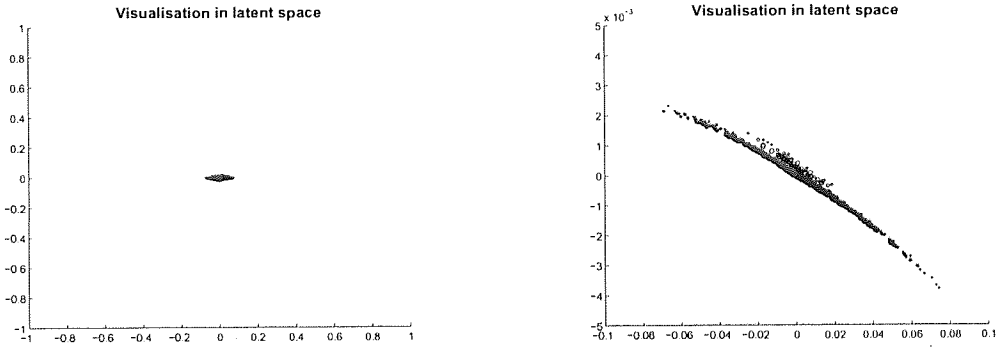


(b)

Figure 4.2: The 2-dimensional visualisation. (a) uses modified GTM with σ^2=0.05. (b) uses the standard GTM. The data created from different Gaussian centres are colour coded differently.

4.3.1 Data-dependent Variance

In the previous section, the variance was uniform for all the data points, making the result almost the same as the standard GTM function. To make confidence matter in the algorithm, the variance is different for each data point converting from confidence values by, $\sigma_n^{*2} = \frac{1}{C_n}$. The variance is calculated from (4.8). Figure 4.3 shows the result when not using a K value, $K = 1$. In this figure, all the data points are mixed together at the centre of the plot. This shows a very poor representation of the data that are created from three different components. In the following experiment shown in Figure 4.4, different fixed K values are used on this data. The bigger circles identify lower confidence value and the smaller ones are the higher confidence value (ie. large circles for large uncertainty).



(a) (b)

Figure 4.3: The resulting visualisation using modified GTM where $\sigma_n^2 = \frac{1}{C_n}$. (a) is with normal scaling. (b) is a zoomed version. The poor result is due to bad scaling of σ_n^2.

In Figure 4.4(a), K=0.5, data become more separated than not adapting K, but the separation is still not clear. The data points still stay at the centre of the picture. After decreasing the value of K to 0.05 and 0.01 in Figure 4.4(b) and Figure 4.4(c) respectively, the data become more dispersed. The results show that when K=0.01, they have a very clear separation. Compare the result with Figure 4.4(d) where K is optimised by maximum likelihood from equation (4.11). The estimated K value from the maximum likelihood is K=0.0167. Fig 4.4(c) and 4.4(d) give very similar visualisations.

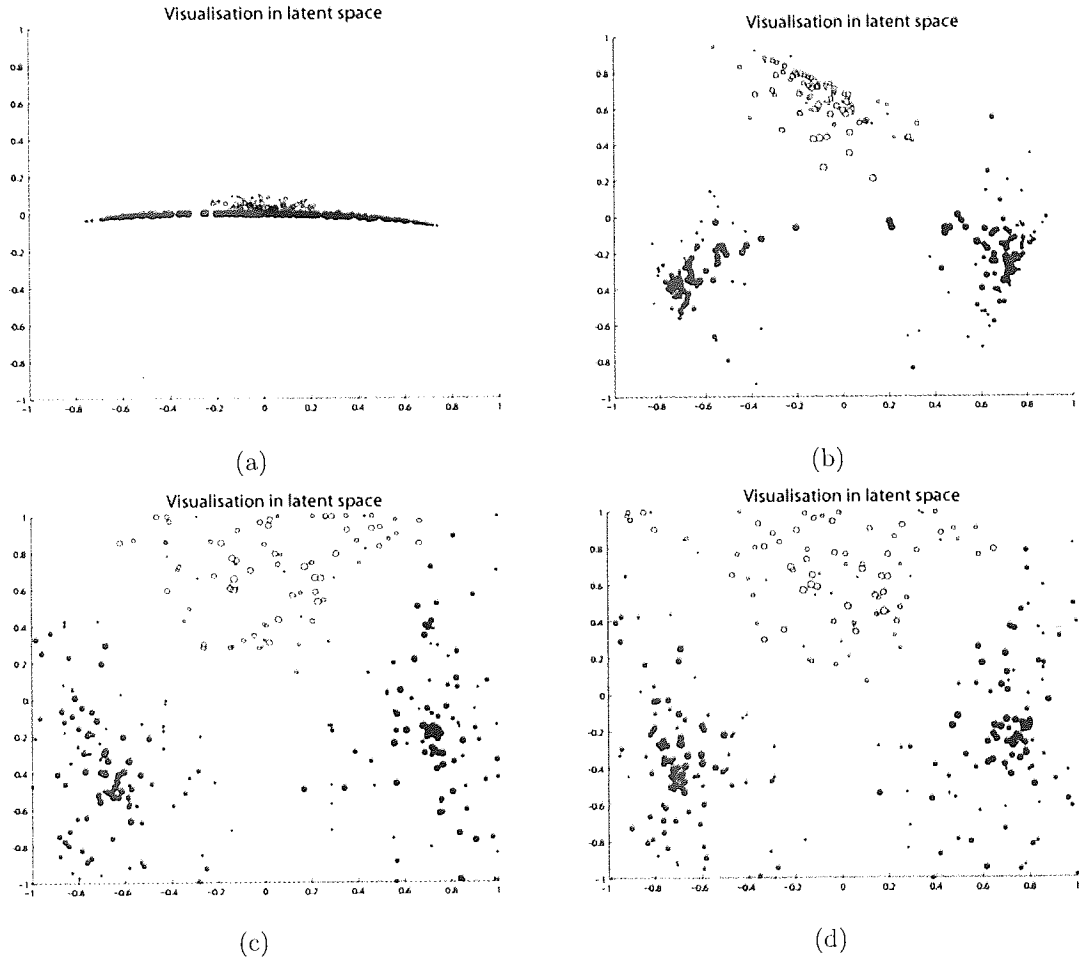


Figure 4.4: The resulting visualisations using modified GTM with different K values. (a) is using K=0.5, (b) is using K=0.05, (c) is using K=0.01 and (d) estimates K to find the appropriate K to suit $\sigma^2 = K\sigma^{*2}$, K=0.0167. The size of the circles represent the size of variance. Big circle size corresponds to big variance. Different colours of circles refer to data generated from different Gaussian centres.

The previous experiments show that the modified GTM algorithm works well when using an estimated K value to find the appropriate ranges of the variance. However, the experiments do not verify that the modified GTM improves the projection of the data when taking into account the uncertainty. The following experiments will be used to verify this by applying noise on the same data set as the previous one but some points are selected to add greater noise to make them outliers. The uncertainty values in this problem are measured by using the distance of the point to the original plane. The farthest point has the confidence value 0 and the points that lie on the plane have confidence value 1. Then two GTM maps are applied. Figure 4.5(a) shows the resulting visualisation using standard GTM to reduce dimensions when there is small noise. Figure 4.5(b) is the result using the standard GTM with higher noise, σ^2_{noise}=5 with 10% of the data points selected as outliers. Figure 4.5(c) is the resulting visualisation from the modified GTM. The result shows that the standard GTM with small noise creates high distortion but the modified GTM creates a better visualisation, similar to the smaller noise in 4.5(a). This figure also shows points divided into four different groups representing the projection of the four distinct cluster.

However, the GTM model contains the size of the latent space as a parameter. Changing this shape can alter the result. Figure 4.6 shows the result using the same data as in Figure 4.5 but reducing the latent space from 8x8 to 4x4. Both results from the standard GTM and the modified GTM are changed. Although the new result creates a worse representation compared to the previous one, the modified GTM gives a lot better representation than the standard GTM. The modified GTM can better preserve the original clusters of the points.

The number of RBF centres also affects the result on both the standard and modified GTM models. Figure 4.7 uses 64 RBF centres and the number of latent points equals 8x8. The results are better than the previous two figures. However the large number of RBF centres tend to create overfitting. However, regularisation can help to ovoid overfitting. Furthermore, using number of centres closes to number of data point is difficult in the real-world data with large data set.

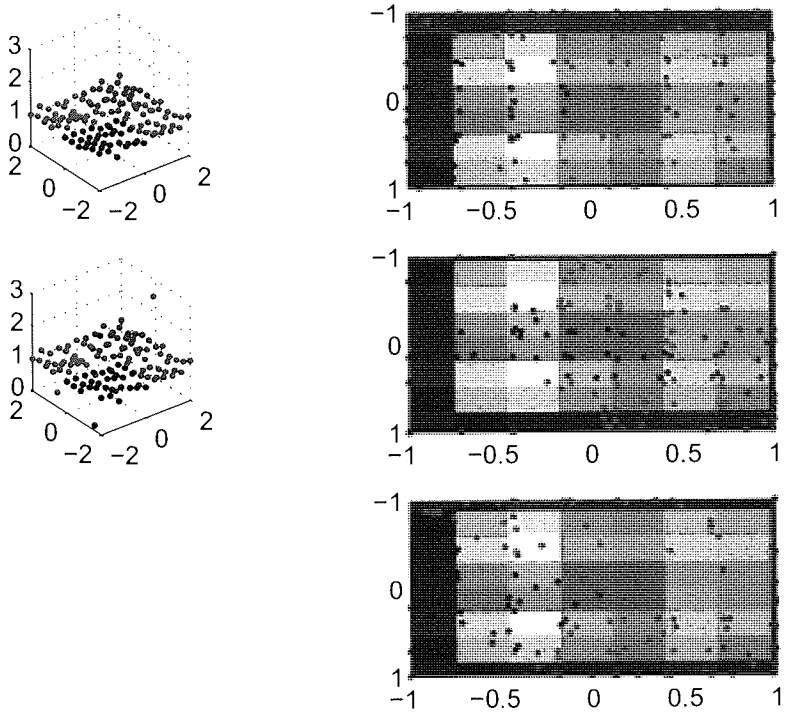


Figure 4.5: The resulting visualisation of the synthetic data. The number of latent points is in an 8x8 rectangular grid and the number of RBF centres is 16. Data points divided into four different groups representing the projection of the four distinct cluster. (a) shows the resulting visualisation using standard GTM to reduce dimensions when there is small noise. (b) is the result using the standard GTM with higher noise, σ^2_{noise}=5 with 10% of the data points selected as outliers. (c) is the resulting visualisation from the modified GTM. This shows no overlap between clusters when the data are projected down to the lower dimension. The magnification factor is shown in the background. The lighter color represent higher magnification factor.

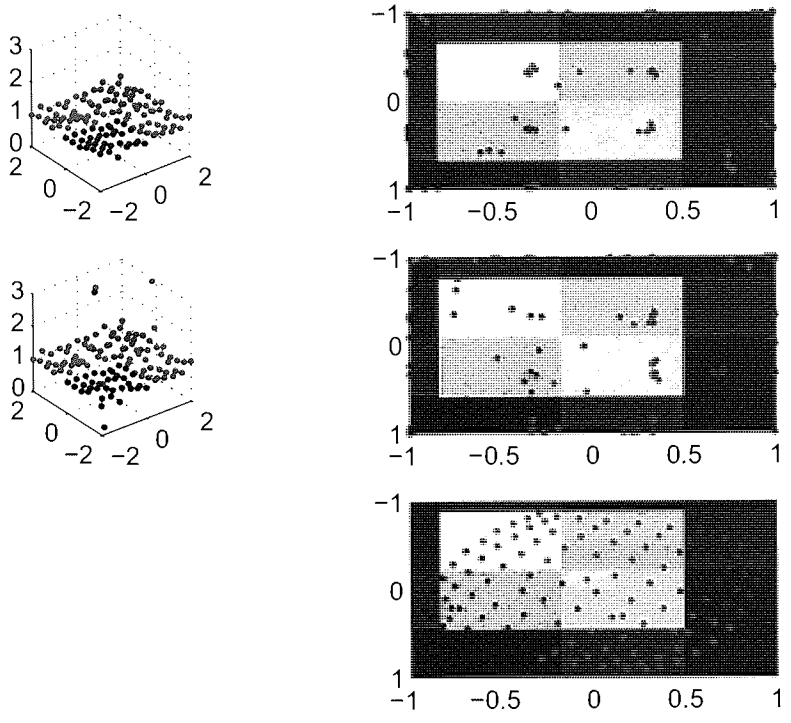


Figure 4.6: The resulting visualisation of the synthetic data. The number of latent points is in an 4x4 rectangular grid and the number of RBF centres is 16. Data points divided into four different groups representing the projection of the four distinct cluster. (a) shows the resulting visualisation using standard GTM to reduce dimensions when there is small noise. (b) is the result using the standard GTM with higher noise, σ^2_{noise}=5 with 10% of the data points selected as outliers. (c) is the resulting visualisation from the modified GTM. Too few latent points results in the collapse of the projection space into fewer data point. Nevertheless, the modified GTM gives a better representation compared to the standard model.

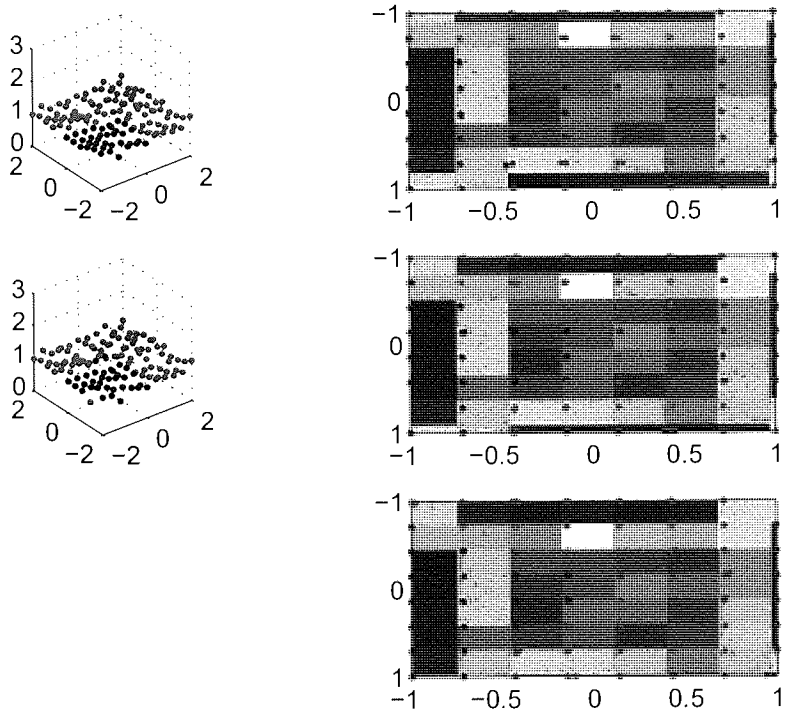


Figure 4.7: The resulting visualisation of the synthetic data. The number of latent points is 8x8 and number of RBF centres is 64. (a) shows the resulting visualisation using standard GTM to reduce dimensions when there is small noise. (b) is the result using the standard GTM with higher noise, σ^2_{noise}=5 with 10% of the data points selected as outliers. (c) is the resulting visualisation from the modified GTM. The number of RBF centres is large and tends to overfit the model. Four different colours represents four different clusters.

4.3.2 Comparisons of the synthetic data results

The synthetic examples in the previous section only show the visual comparison of the standard and modified GTM models. This section will show quantitative comparison by using classification results of the four classes shown in different colours. RBF classifiers from the visualisation results are used to create 4-class outputs. The data are trained on randomly selected 70 out of 121 data points. The classification rates and number of misclassified data points of Figure 4.5, Figure 4.6 and Figure 4.7 are compared in Table 4.1.

	Standard GTM		Modified GTM	
Figure	Accuracy(%)	Points missed	Accuracy(%)	Points missed
4.5	57.15%	46	90.75%	6
4.6	44.07%	63	91.52%	4
4.7	74.79%	30	78.95%	21

Table 4.1: Classification results of the synthetic examples comparing visualisations between standard and modified GTM.

Table 4.2 compares the error values per data point which are the negative of the log likelihoods divided by the number of points with p-value=0.01. The results show that the modified GTM gives better results than the standard GTM. In this Table, the latent space size is 8x8 and number of RBF centres is 16.

		Standard GTM		Modified GTM	
Outliers		Error		Error	
Percent	Variance	mean	Variance	mean	Variance
10	1	2.703	0.027	2.10	0.035
	2	2.998	0.027	2.221	0.037
	3	3.118	0.060	2.352	0.057
	4	3.094	0.025	2.331	0.061
	5	3.236	0.040	2.450	0.031
	6	3.303	0.037	2.527	0.078
	7	3.327	0.044	2.551	0.11
	8	3.401	0.040	2.478	0.046
	9	3.397	0.042	2.791	0.016
	10	3.345	0.042	2.849	0.060
20	1	3.056	0.012	2.290	0.017
	2	3.346	0.045	2.655	0.051
	3	3.461	0.022	2.682	0.008
	4	3.50	0.071	2.774	0.036
	5	3.568	0.032	2.921	0.028
	6	3.530	0.041	3.042	0.031
	7	3.591	0.049	3.065	0.013
	8	3.494	0.008	2.955	0.054
	9	3.682	0.023	3.145	0.078
	10	3.637	0.025	3.077	0.035

Table 4.2: A table showing the means and variances of the Error (the negative of log likelihood) for different percents and variance of outliers.

4.3.3 The real data sample with attached uncertainty

The example dataset we are using is temporal microarray data provided by Prof. Colin Smith of University of Surrrey as part of the BBSRC project "MARIE" on exploiting Genomics: *S. coelicolor* is a complex mycelial Gram-positive bacterium which undergoes developmental changes leading ultimately to sporulation and production of antibiotics and other secondary metabolites[16]. The data consists of 7,825 genes which are collected from ten different times which are 16, 18, 20, 21, 22, 23, 24, 25, 39 and 67 hours after the inoculation of the growth medium. Each Cy3-labelled cDNA sample was using to compare against Cy5-labelled *S. coelicolor* genomic DNA (gDNA) as the common reference.

Each gene also has a confidence level added which is derived from the Bluefuse software. The preprocessing of the microarray data consisted of: i) correcting the

data for spatial effects, ii) taking the log-ratio of the signal and the reference measurements;applying across-condition normalisation; iii) applying a variance filtering and a low-value filtering to remove genes that are not significantly expressed; and finally iv) normalising each pattern of gene expression by subtracting the values at the first time step. The last procedure has been applied in order to investigate the gene expression patterns factoring out the absolute expression level.

Applying this modified GTM to the real data set of microarray expression values, gives the following results. In Figure 4.8, where K=0.01, the data congregate at the centre of the plot. While in Figure 4.9, where the K value is obtained by log likelihood the visualisation is more scattered. For comparison, Figure 4.10 illustrates the result from standard GTM. In these figures we are also displaying the magnification factors behind the data points.

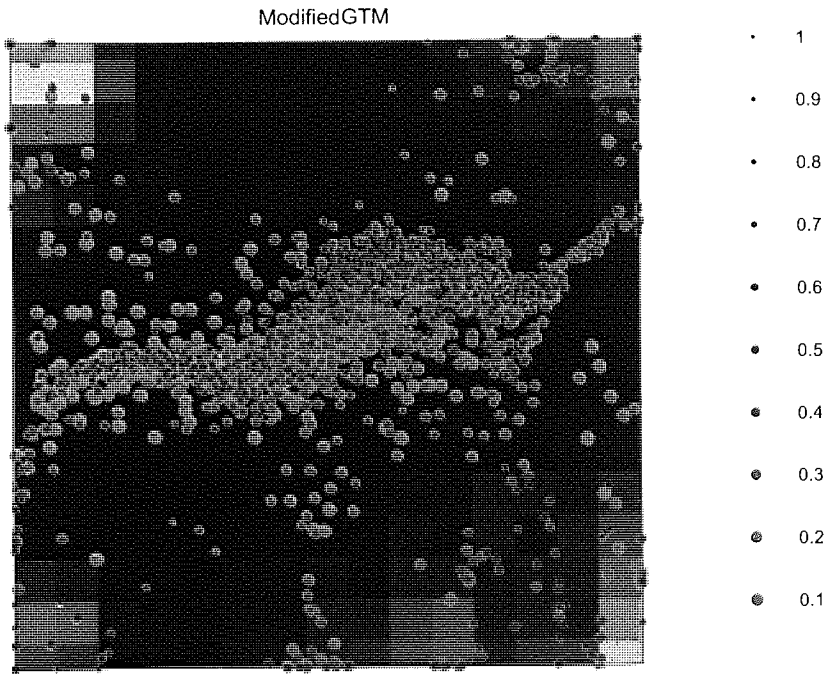


Figure 4.8: The resulting visualisation using modified GTM with fixed K=0.1.

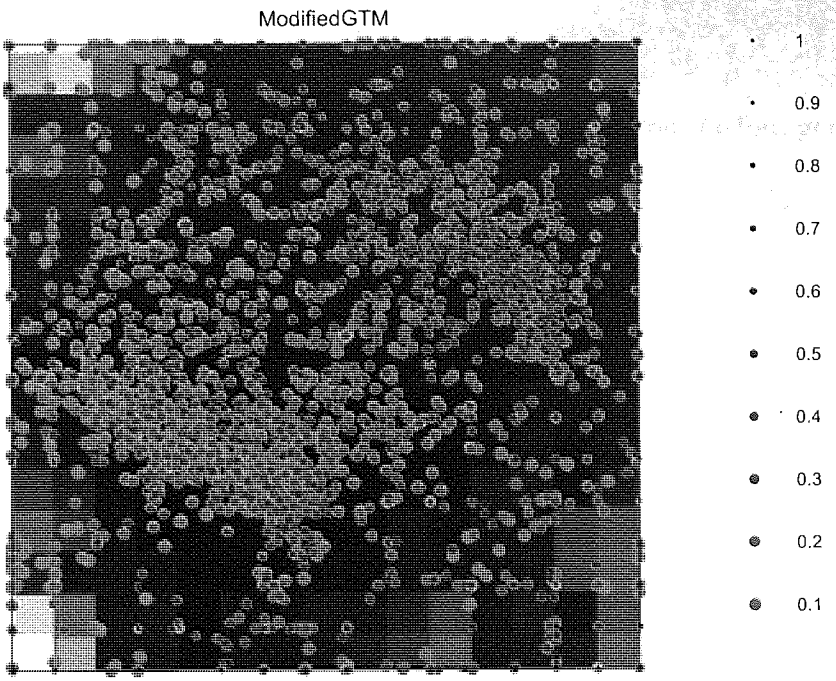


Figure 4.9: The microarray visualisation of the microarray data set using modified GTM and estimating K to find the appropriate K to suit $\sigma^2 = K\sigma^{*2}$. The value of K is estimated by using the maximum log likelihood.

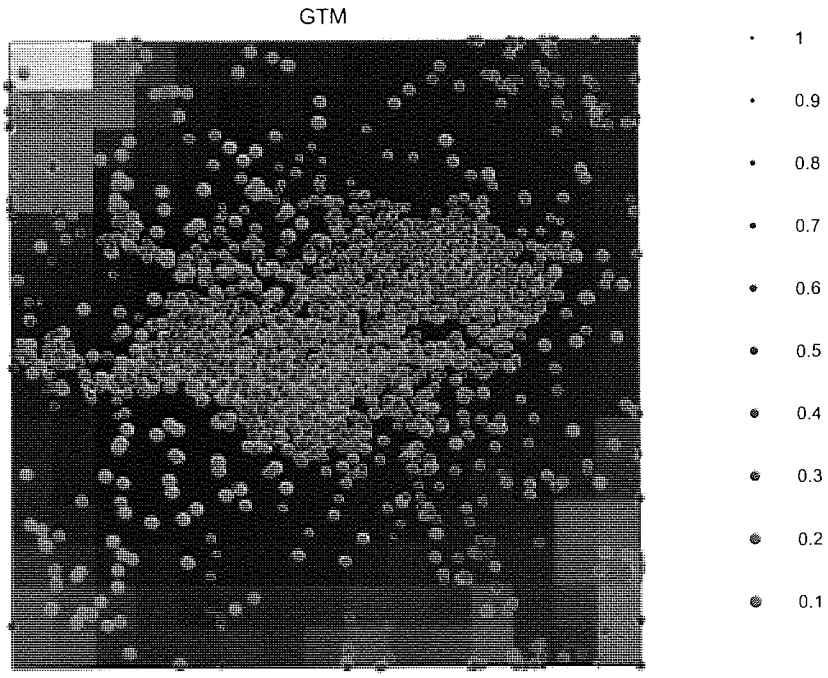


Figure 4.10: The resulting visualisation using the standard GTM .

4.3.4 The Comparison between the Standard GTM and the Modified GTM

To make a comparison between the two models, we will focus on the four genes grouped together in the standard model. Those genes are highlighted in figure 4.11. Figure 4.12 gives a closer look of those genes we are focusing on, 'SCO0271', 'SCO5208', 'SCO4897' and 'SCO4260'. Figure 4.13 shows the gene expression profiles of those four genes. Although they have similar behaviours, they should not be grouped together since some genes have a lot lower confidence than the others. The low level of confidence of genes can alter the gene expression collected from the measurement process. The similar genes expressions with higher confidences should rather be grouped together.

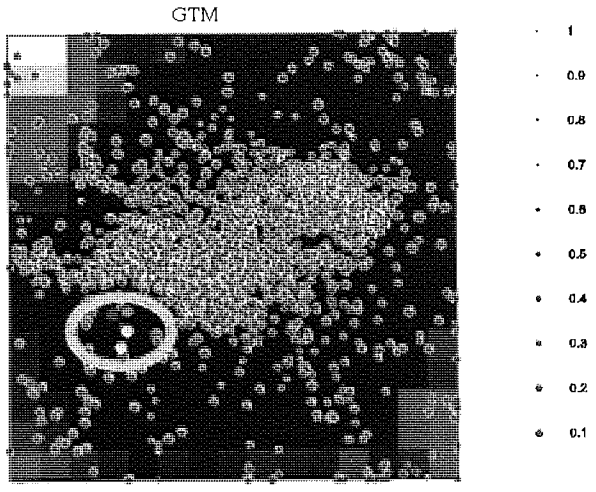


Figure 4.11: The resulting visualisation of the standard GTM model, focusing on a grey circle region that we will use to investigate.

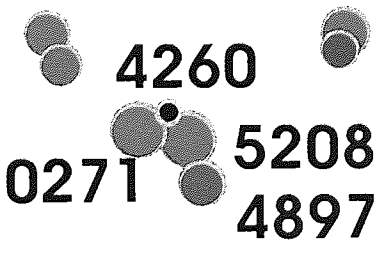


Figure 4.12: Genes 'SCO0271', 'SCO5208', 'SCO4897' and 'SCO4260' are labelled in the standard GTM visualisation.

In modified GTM, those four genes become separated. Their locations in the new visualisation space are highlighted in grey colours shown in figure 4.14. The next four

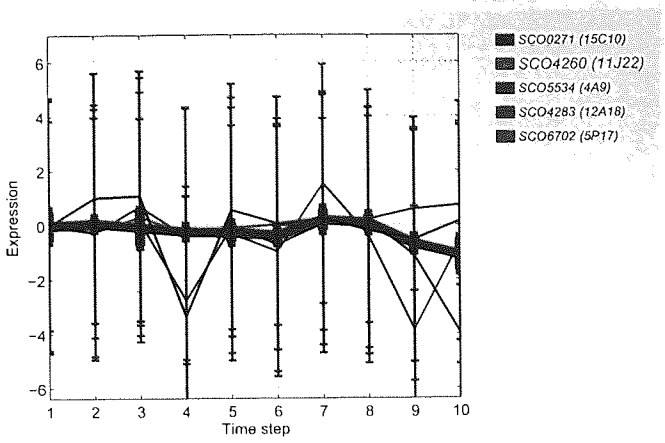
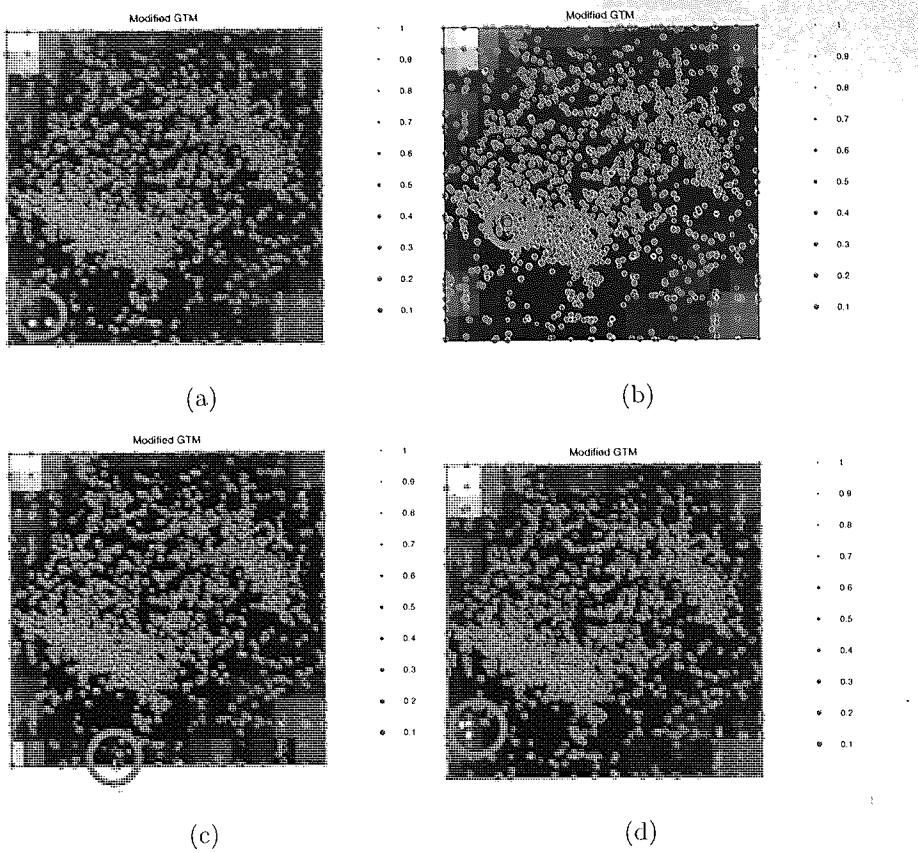


Figure 4.13: The expressions of the selected genes

figures after that are the gene expressions of those genes with their new neighbours. Although the four genes are spread away from one another, they still occur in the bottom left corner of the picture as it was in the standard GTM.

The highest confidence genes amongst the four is genes 'SCO4260'. The gene expression of it and its new neighbour in the modified GTM is shown in Figure 4.17. This gene and its new neighbour have similar expressions and all of them have high levels of confidence, which is a good representation that we expected. However, they are not highly expressed genes, their locations are at the lower edge of the plot. In standard GTM, only highly expressed genes will come close to the edge and genes with low levels of expression will be close to the centre. This has changed in this modified GTM model. The standard GTM model is better at representing the location of the genes which corresponds to the levels of expression. This modified GTM model poorly separated the low and high levels of gene expressions compared to the standard GTM.

On the contrary, the modified GTM better grouped the genes with similar expressions together compared to the standard model as shown in Figure 4.15, 4.16, 4.17 and 4.18 which show the gene expression profiles of previous neighbours of gene 'SCO4260'. Their new neighbours have similar values of confidence. These figures show that the modified GTM also tends to group genes with similar values of confidence together. However, the similar gene expression does not guarantee that they will be grouped together as in the standard GTM because the modified GTM takes into account the



(a) (b)

(c) (d)

Figure 4.14: The resulting visualisations of the modified GTM of those genes that previously grouped together in the standard model. The new locations are highlighted in a grey colour. Figure 4.14(a) is the location of 'SCO04897'. Figure 4.14(b) is the location of 'SCO5208'. Figure 4.14(c) is the location of 'SCO4260'. Figure 4.14(d) is the location of 'SCO0271'.

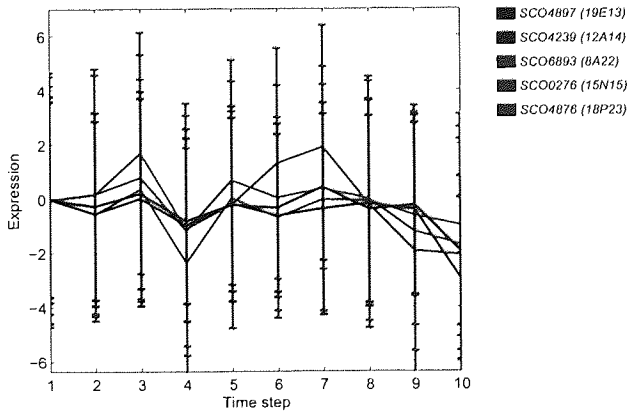


Figure 4.15: The gene expression profiles of gene 'SCO4897' and its neighbours in the modified GTM.

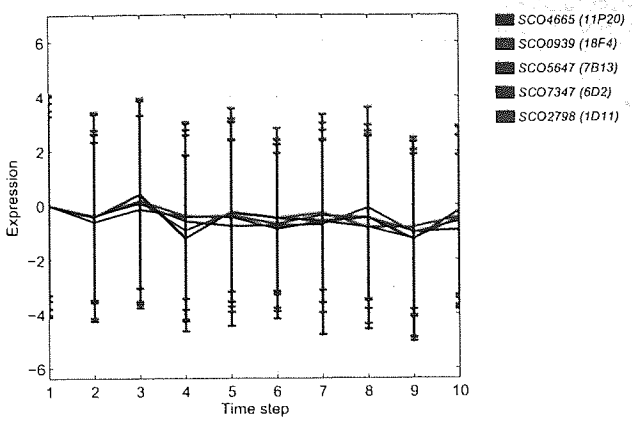


Figure 4.16: The gene expression profiles of gene 'SCO5208' and its neighbours in the modified GTM.

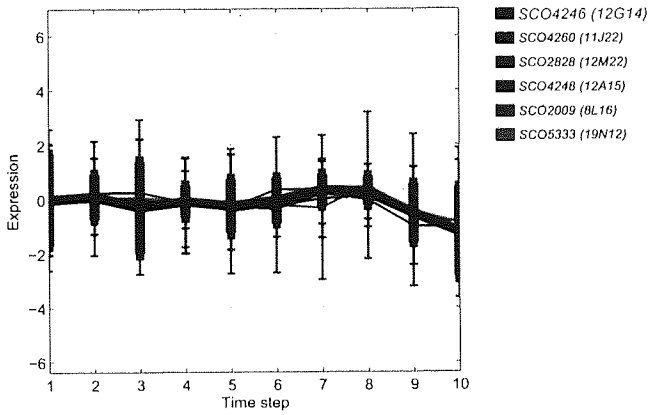


Figure 4.17: The gene expression profiles of gene 'SCO4260' and its neighbours in the modified GTM.

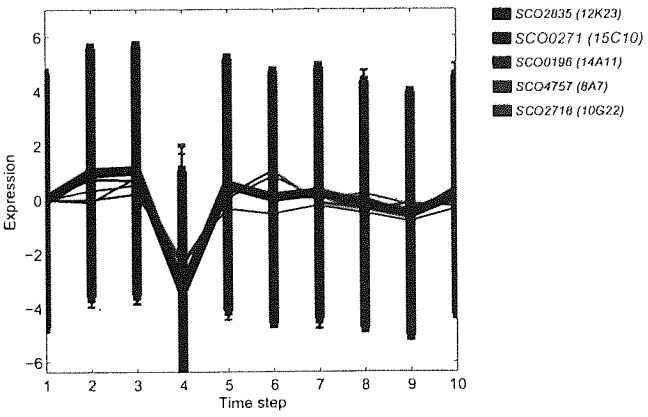


Figure 4.18: The gene expression profiles of gene 'SCO0271' and its neighbours in the modified GTM.

confidence level.

4.4 Conclusion

This chapter has introduced an approach to incorporate uncertainty into the GTM model. From the synthetic data, it was shown that uncertainty influences the final visualisation projections and give quite consistent results, as shown by variances on different runs.

In addition to this modified GTM approach, the next chapter will discuss modifications of the projective model, the NeuroScale mode, to incorporate uncertainty. Furthermore, the real application and comparisons between different types of modified visualisation will be applied and discussed in Chapter 6.

Chapter 5

Probabilistic Approaches to NeuroScale

We previously discussed the requirement for methods capable of incorporating uncertainty. In the previous chapter we have discussed the possibility of incorporating uncertainty measures by using the GTM technique. Now in this chapter, an alternative approach is proposed by using one of the most effective projective visualisation methods - NeuroScale. However, the NeuroScale projection is a deterministic approach and is not designed for probabilities. In this chapter we introduce new methods to enhance NeuroScale with probabilistic techniques.

5.1 Heuristic Approaches

In order to add uncertainty to the NeuroScale model, modifications from the original method are required. This section will suggest the more heuristic approach that could be used for incorporating uncertainty which can be simply modifying the input vector to modify the STRESS measure or error function. The available uncertainty information can therefore be attached easily to create a modified heuristic NeuroScale. We will then proceed to describe a more fundamental approach.

5.1.1 Additional input noise

The simplest method to attach uncertainty to the basic NeuroScale model is by attaching noise variance as an extra dimension to the input pattern vector [15]. This method provides a simple implementation and can be used successfully in some applications with low dimensionality. It is important that data with the same values but different uncertainty levels will be projected down to different locations in a visualisation space. If the original data is $\mathbf{x}$, the modified input, $\hat{\mathbf{x}}$, is defined as

$$\hat{\mathbf{x}} = [\mathbf{x} \quad \sigma]. \tag{5.1}$$

The new distance measure between two data points, defined as d'_{ij}, will become

$$d'^2_{ij} = (d^2_{ij} + (\sigma_i - \sigma_j)^2), \tag{5.2}$$

where d_{ij} is the original Euclidean distance between two data points.

However, the main drawback of this method is that the new dimension will only be one extra dimension. So when the dimensionality is high, this new dimension will not contribute significantly to the projection, depending on scaling used.

5.1.2 Modified cost function

In this second approach the confidence values in data values are used to modify the original STRESS function:

$$E = \sum_{i}^{N} \sum_{j}^{N} (D^*_{ij} - D_{ij})^2, \tag{5.3}$$

by including a confidence term, C_i which can be defined as the inverse of the variance $C_i = \frac{1}{\sigma^2}$.

The modified cost function is

$$E = \sum_{i}^{N} \sum_{j}^{N} K_{ij}(d^*_{ij} - d_{ij})^2, \tag{5.4}$$

where K_{ij}=min (C_i, C_j); C_i is a confidence value on data point i. More generally, K_{ij} is a function between points i and j that reduces the influence of these points on disturbing the map depending on our confidence in these data.

Equation (5.4) can be interpreted that points with low inter-point confidence are less important in determining the mapping parameters.

The inter-point confidence, K_{ij} is the minimum confidence values between the two points. Hence, the new network gained by using this new cost function in training will be less likely to be influenced by those data which have low confidence.

In addition, the extra dimensionality can also be attached as suggested in the previous section. The main advantage is especially for the novel data point which has a different uncertainty measure to the training data samples. However, the confidence measure will be included both in cost function and the distance measure. Nevertheless, attaching the confidence to the cost function does not have a proper probabilistic explanation. In addition, the two points are treated as exact points rather than two probability distributions.

5.2 The Probabilistic Approach

In a more complete probabilistic approach, rather than viewing each given piece of information as a single precise location of a data point, each piece of information is viewed as a sample from a probability distribution. Therefore, the distance measure needs to be modified to consider distances between probabilistic density functions instead of two data points as in the standard NeuroScale.

In this section, the introduction of distance measures of the probability function will be made and the modification of the NeuroScale cost function and training algorithm using the probabilistic approach will be developed.

5.2.1 Background

In terms of uncertainty measurement, *entropy* is a common expression for representing the uncertainty of information. Information and uncertainty are related since from a given event if all the uncertainty is removed, the remaining is information [41]. The measurement of uncertainty as developed by Shannon is referred as the Shannon entropy [74]. However, a more abstract generalised form of entropy is suggested by Renyi [68].

The Renyi Entropy

Entropy gives knowledge on how well data contains reliable information. It measures the knowledge of uncertainty in the information. The less predictable a piece of information the larger the information content is. The more generalised entropy is the Renyi's entropy which is defined as:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \int_{-\infty}^{\infty} f_x^\alpha(x) dx. \tag{5.5}$$

In this Renyi's family, there are many different types of entropy that are commonly used by choosing the appropriate α. For $\lim_{\alpha \to 1} H_\alpha(X)$, it becomes the Shannon entropy, which is one of the most famous entropy measures. For $\alpha = 2$, it is called the quadratic entropy.

The entropy can be efficiently estimated through kernel approximation by using the Parzen window where the estimated pdf $f_x(z)$ of a random vector $\mathbf{x}$ is:

$$\hat{f}_x(\mathbf{z}, \{\mathbf{x}\}) = \frac{1}{N} \sum_{i=1}^{N} G\left(\mathbf{z} - \mathbf{x}_i, \sigma^2 I\right), \tag{5.6}$$

where $\mathbf{x_i}$ are observations of the random vector and the kernel, $G()$ is often , but not exclusively a symmetric Gaussian:

$$k(\mathbf{z}) = G(\mathbf{z}, \sigma^2 \mathbf{I}) = \frac{1}{(2\pi)^{M/2}\sigma^M} \exp(\frac{\mathbf{z}^T\mathbf{z}}{2\sigma^2}). \tag{5.7}$$

Therefore a simple estimation of Renyi entropy can be made from Parzen estimation. The quadratic Renyi entropy is:

$$H_{R_2} = \log \int f(z)^2 dz, \tag{5.8}$$

which can be written as the logarithm of a potential field in the data space,

$$H_{R_\sigma} = -\log(V(x)) \tag{5.9}$$

$$= -\log\left[\frac{1}{N^2} \sum_i \sum_j G(x_i - x_j, 2\sigma^2 I)\right]. \tag{5.10}$$

We are here following Principe [31]. The information potential, $V(y)$ is named after the physical interaction in mechanics in which each particle, y, generates a potential which is a non-negative term and inversely proportional to the distance.

Similar to its physical interpretation, the derivative of the information potential is the information force.

$$\begin{aligned} F_i &= \frac{\partial V(y_i)}{\partial y_i}, \\ &= -\frac{1}{N^2\sigma^2}\sum_{j=1}^{N} G(y_i - y_j, 2\sigma^2\mathbf{I})(y_i - y_j), \\ &= \frac{1}{N^2\sigma^2}\sum_{j=1}^{N} V_{ij}d_{ij}. \end{aligned} \tag{5.11}$$

Divergence

The relative entropy or the divergence between two densities is equivalent to a measure of the distance between two distributions. The Renyi divergence is :

$$H_{R_\alpha}(f,g) = \frac{1}{1-\alpha}\log\int_{-\infty}^{\infty}\frac{f(x)^\alpha}{g(x)^{\alpha-1}}dx. \tag{5.12}$$

In the previous section, we mention $\alpha = 2$ however, it is more common to set $\alpha = 1$. The analogy to $\alpha = 2$ will be mentioned later in the section. For $\alpha = 1$, it is called the Kullback-Leibler divergence (KL divergence):

$$\lim_{\alpha\to 1} H_{R_\alpha}(f,g) = D_{KL}(f,g), \tag{5.13}$$

which is defined as the dissimilarity between two distributions $f(x)$ and $g(x)$, defined as:

$$D_{\mathrm{KL}\{f(x)\|g(x)\}} = \int_{-\infty}^{\infty} f(x)\log\frac{f(x)}{g(x)}dx. \tag{5.14}$$

It is clear from (5.14) that the divergence is not symmetric,

$$D_{\mathrm{KL}\{f(x)\|g(x)\}} \neq D_{\mathrm{KL}\{g(x)\|f(x)\}}.$$

To symmetrise this divergence in order to make it an appropriate distance measure an average of the two is performed:

$$D_{\{f(x)\|g(x)\}} = \frac{D_{\mathrm{KL}\{f(x)\|g(x)\}} + D_{\mathrm{KL}\{g(x)\|f(x)\}}}{2}. \tag{5.15}$$

The summation of all the distance measures using KL divergence between two data samples with identical variances is equivalent to finding the information potential $V(y)$.

Therefore minimising the KL divergence is similar to minimising the information force in the space.

If both $f(x)$ and $g(x)$ are assumed to be Gaussian distributions with mean μ_1, μ_2 and covariance Σ_1, Σ_2 respectively. The divergence is

$$D_{\mathrm{KL}\{f(x)\|g(x)\}} = \frac{1}{2}(Tr(\Sigma_i^{-1}\Sigma_j) + \log|\Sigma_j/\Sigma_i| + \Sigma_j^{-1}(\mu_i - \mu_j)^T(\mu_i - \mu_j) - 2l), \quad (5.16)$$

where l is the dimensionality of the data. The symmetrised distance between distributions indexed by i,j can be expressed as:

$$D_{ij} = \frac{1}{2}(Tr(\Sigma_i^{-1}\Sigma_j) + Tr(\Sigma_i\Sigma_j^{-1}) + (\Sigma_i^{-1} + \Sigma_j^{-1})(\mu_i - \mu_j)^T(\mu_i - \mu_j) - 2l). \quad (5.17)$$

For Gaussian mixture distributions with spherical variance $\Sigma_i = \sigma_i^2\mathbf{I}$:

$$D_{ij} = \frac{1}{4}\left(l\left(\frac{\sigma_i^2}{\sigma_j^2} + \frac{\sigma_j^2}{\sigma_i^2} - 2\right) + \left(\frac{1}{\sigma_j^2} + \frac{1}{\sigma_i^2}\right)(\mu_i - \mu_j)^2\right). \quad (5.18)$$

This distance measure can be used for the NeuroScale model as a probabilistic distance measure.

5.2.2 Uncertainty weighted dissimilarity

In standard NeuroScale, data points are assumed to have precise locations without a proper way of attaching uncertainty information. The Euclidean distance does not take into account the level of uncertainty and treats every data point equally. As a result, the low certainty data points can distort the whole dissimilarity matrix completely.

In order to express this extra uncertainty information path, a probability distribution could be used for representing the data points with means and variances indicating different levels of uncertainty.

As mentioned earlier, KL divergence is one of the convenient ways of estimating the dissimilarity measures between pdfs. The dissimilarity between two data points, (5.16), does not only depend on the distance between the means of the distribution but also the size of the uncertainty. Low certainty data points, high variance, tend to give small dissimilarity measures since they give a large denominator compared to the low variance data points. Nevertheless, the distance also depends on the Euclidean distance of means of the distributions weighted by the uncertainty.

If two probabilities have the same variances, from (5.18) the distance measure will become

$$D_{ij} = \frac{(\mu_i - \mu_j)^2}{2\sigma^2}, \tag{5.19}$$

which is proportional to the Euclidean distance with a common variance, σ^2 as a dividing factor. This form is similar to the heuristic NeuroScale in 5.4. However, if the variance is large the distance will be reduced. Eventually, the distance of all data points will be similar and may not give sensible results. It is clear that the scaling between two projections will be different but the visualising projection will be the same.

The advantage of this approach is that it can directly transform data dissimilarity using means and variances into a single representation of distance measures. It can be used with the NeuroScale easily without any modification, if the projection space remains unchanged.

The modification of the input dissimilarities is straightforward and does not require any modification in the model networks' structure for NeuroScale. The cost function remains the same providing the output measure is still being measured by Euclidean distance. The Euclidean distance in the projection space is used for ease of interpretation especially for visual perception. Uncertain data can simply be added as an additional input of the network as useful information especially for the projection of the novel data. However, the projection dissimilarities in the low dimensional space are measured by using Euclidean distance. As a result, the low variance data are positioned further away from each other than the high variance ones, since D_{ij} is higher. Moreover, if there is a big variation in variance, the result can be dominated by the variance rather than the Euclidean distance. It can create, for example, a straight line projection with low variance data at the edge and the high variance data clustered in the middle.

5.2.3 The probabilistic approach

In the previous approach, the projection of a distribution was represented by a single point. However, the projection of the distribution should also be a distribution rather than a precise single point, if it is to reflect uncertainty fully.

Consequently, instead of comparing two spaces with different distance measures

which will give different impressions from the original structure of the data, the KL divergence measure can be applied to both data and projection spaces. However, this requires a modification of the optimisation process since the gradient will be different. To achieve a topographic representation, the standard NeuroScale cost function is modified to

$$E = \sum_{i}^{N} \sum_{j}^{N} (D_{ij}^{*} - D_{ij})^2, \tag{5.20}$$

The distribution in both spaces are based on the assumption of Gaussian distributions whose distances are calculated by (5.17). The variance of points in the low dimensional space is fixed to be the same as the points in the original space. As mentioned in the previous section, the distance, D_{ij}, which involves high variance data points will make the distance in the visualisation space smaller. As a direct result the distance of the mean between two probability distributions, indexed i and j, will make a small contribution to the stress measure. Therefore, the high precision data will give higher impact on the stress measure.

Replacing (5.20), with the KL divergence derived from the Gaussian distribution from (5.17) assuming diagonal Gaussians, we get the equation for modified Stress in terms of Euclidean distance d as:

$$E = \frac{1}{2} \sum_{i}^{N} \sum_{j}^{N} \left((q-p) \left(\frac{\sigma_i^2}{\sigma_j^2} + \frac{\sigma_j^2}{\sigma_i^2} - 2 \right) + (d_{ij}^2 - d_{ij}^{*2}) \left(\frac{1}{\sigma_i^2} + \frac{1}{\sigma_j^2} \right) \right)^2, \tag{5.21}$$

where p, q are the dimensionalities of high dimensional data space and low dimensional visualisation space respectively. It can be seen that the latter part of (5.21) has an analogy to the cost function created by the heuristic approach in (5.4). However, the difference is that this approach uses the squared distance rather than the absolute distance because of the derivation with the KL divergence. Moreover, the distances of data points in this equation are weighted by the sum of the inverse of the variances which resembles the minimum of the inverse in the heuristic approach. Nevertheless, the weight of (5.21) is obviously stronger than the weight resulting in (5.4) because of its square. Therefore, for the probabilistic NeuroScale the distance of the low confidence data will have less effect on the overall STRESS measure. As a result, the location of low confidence data are not likely to influence the overall distribution of points in the

visualisation space.

The Gradient

The optimisation of the locations of y can be achieved by a gradient-based method for which we need the derivative of (5.20), which is given by

$$\frac{\partial E}{\partial y_i} = 2\sum_j (D_{ij} - D^*_{ij})(y_i - y_j)\left(\frac{1}{\sigma_i^2} + \frac{1}{\sigma_j^2}\right). \tag{5.22}$$

The model is initialised by PCA and the locations of y_i can be simply achieved by using optimisation algorithms, such as scaled conjugate gradient or to imitate the mechanism of the shadow target algorithm.

5.2.4 Modification of the Shadow Target Algorithm

In NeuroScale the shadow target of a data point is required for the network training. However, for the probabilistic NeuroScale the shadow target of the mean of the data point is required. In addition to the expression of the error function in (5.21), the error for the distribution can be achieved by minimising the error function of the negative log likelihood of the distribution, $\phi(x)$, which we restrict the distribution to a Gaussian kernel function with spherical covariances,

$$\phi(t_i|x_i) = \frac{1}{(2\pi\sigma_i^2)^{\frac{p}{2}}} \exp\left\{-\frac{\|t_i - y(x_i)\|^2}{2\sigma_i^2}\right\}, \tag{5.23}$$

where y is the mapping function from high dimensional space to the projection space and t_i represents an explicit target value.

Therefore the likelihood function is given by

$$E = \sum_{n=1}^{N} -\ln\{\phi(t_n|x_n)\}. \tag{5.24}$$

Differentiating E over y_i gives

$$\frac{\partial E}{\partial y_i} = \frac{y_i - t_i}{\sigma_i^2}. \tag{5.25}$$

The idea is the same as standard NeuroScale in that the target value will need to be estimated using the shadow target algorithm in which the hypothetical target, $\hat{t}$,

can be estimated using (5.25) and (5.22) as:

$$\hat{t}_i = y_i - \sigma_i^2 \frac{\partial E}{\partial y_i}, \tag{5.26}$$

$$= y_i - 2\sigma_i^2 \left(\sum_j (D_{ij} - D_{ij}^*)(y_i - y_j) \left(\frac{1}{\sigma_i^2} + \frac{1}{\sigma_j^2} \right) \right). \tag{5.27}$$

For the RBF network as a mapping function, the output of the network is obtained by

$$\mathbf{Y} = \mathbf{\Phi W}, \tag{5.28}$$

where $\mathbf{W}$ represents the matrix of the weights in the network and $\mathbf{\Phi}$ is the matrix of hidden unit activations including bias terms and $\mathbf{Y}$ is the output of the network . The weight updating process is the same as in standard NeuroScale, which uses the shadow target to update the weight. For a fixed set of targets, the problem can be solved directly by

$$\mathbf{W} = \mathbf{\Phi}^\dagger \hat{\mathbf{T}}. \tag{5.29}$$

The algorithm is the same as updating the target of the NeuroScale algorithm in which an additional parameter η will be introduced by

$$\hat{t}_i = y_i - \eta \sigma_i^2 \frac{\partial E}{\partial y_i}, \tag{5.30}$$

η is a step length to be restricted to $(0, 1)$ and is just a parameter governing convergence of the gradient descent. So the algorithm for the modified probabilistic NeuroScale can be summarised as:

1. Initialise the network weights, $\mathbf{W}$ to small random values.
2. Initialise η to some small positive value.
3. Initialise $\mathbf{\Phi}$ by intialising appropriate network centres and variances as appropriate.
4. Calculate the pseudo-inverse of the activation function $\mathbf{\Phi}^\dagger$.
5. Use (5.30) to compute estimated targets $\hat{t}_i$.
6. Calculate E from (5.21) and compare with the previous value

(a) If E has increased which means η_y can be too large, set $\eta_y = \eta_y \times k_{\text{down}}$, where k_{down} is a small predefined value to reduce the η_y value. Restore previous values of W.

(b) If E has decreased which means η can be too small, set $\eta = \eta \times k_{\text{up}}$. Generally $k_{\text{down}} < k_{\text{up}}$.

7. Solve for the weights $\mathbf{W} = \mathbf{\Phi}^{\dagger}\hat{t}_i$.

8. If (5.21) has not converged, return to Step 5.

5.3 Illustrations of Modified NeuroScale

5.3.1 Synthetic Example

A synthetic example will be used to illustrate the model performance here [76]. The example consists of samples from three separated Gaussians generated in Re^4 randomly with the same priors and uniformly distributed unit variances. The three clusters are centred differently, with additional noise added to those data with $\sigma^2 = 0.7$. The experiments are applied with different visualisation approaches. Data samples from three different Gaussians will be represented using different labels and the different levels of added noise will be represented by different symbol sizes, in the visualisation, simply for illustration purposes.

Figure 5.1 shows the projection of the standard NeuroScale with standard Euclidean input. Figure 5.2(a) illustrates the projection when the input dimensionality has been extended with the uncertainty information and Figure 5.2(b) uses standard NeuroScale with KL distance as a dissimilarity measure together with added noise information.

This example shows a small variation of the results between the standard NeuroScale, Figure 5.1, and the extended dimensional NeuroScale in Figure 5.2(a). The three clusters whose data points are represented by three different symbols, are well separated with this uniform variance projection. Figure 5.2(b) shows the results of the data points where the KL divergence is taken as an input dissimilarity measure and the visualisation is created by using the standard NeuroScale. This projection created almost a straight line of the data points which are dominated mainly by the

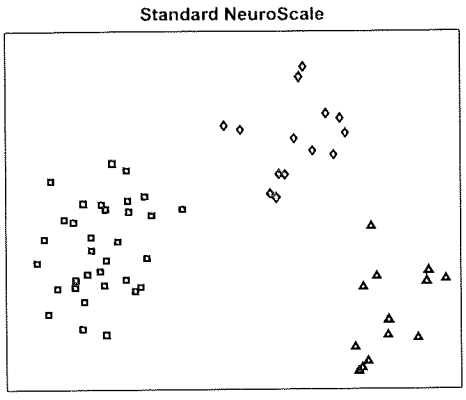
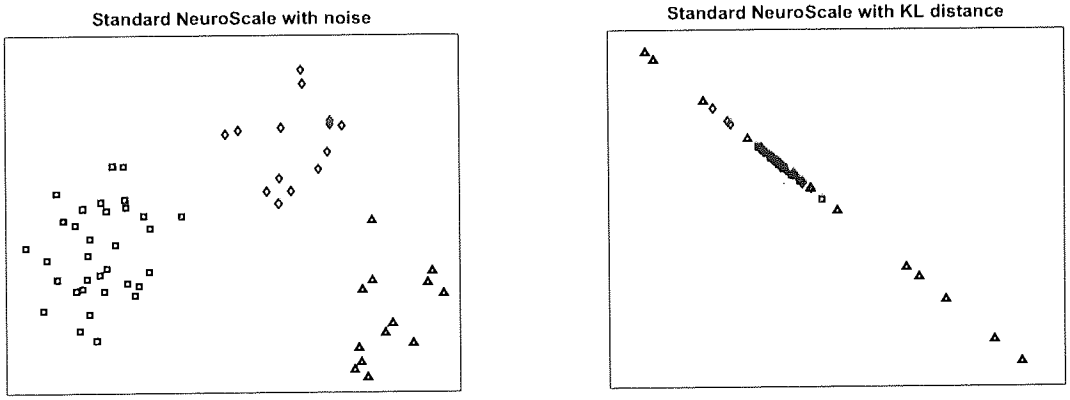


Figure 5.1: The standard NeuroScale projection of the four dimensional data generated from three different Gaussian centres using the standard Euclidean input. Each data input is also disrupted by noise. The data points generated from different centres are represented by different symbols.



(a) (b)

Figure 5.2: (a) The NeuroScale projection with attaching uncertainty as a new dimension. (b) The new NeuroScale projection with uncertainty as an extra input and measuring input distance using a symmetric KL distance.

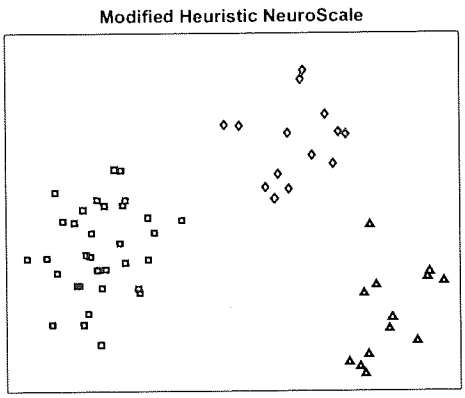
variance of the data points even though there are small differences between variances. They showed no separation between clusters. Data points from different clusters are projected on top of one another.

Additionally, the projecting data as represented by the standard NeuroScale can only be projected to a single precise location for each datum rather than the probability representation of the data. The projection of the data and the dissimilarity measures used should be the same as in the original space if the topographic property is to be exploited, which is the probability representation using KL divergence as a measure. However, to do this, a model modification is required. The heuristic approach of the probabilistic NeuroScale as described in section 5.1 is used for this example and the result is shown in Figure 5.3(a). The modified STRESS function as in (5.20) can be used to optimise the location of the means of the projection space by using scaled conjugate gradient methods [58]. In order to gain the advantage of the novel data projection, the optimised location of the data can be interpolated using RBF networks, called a posterior mapping. This example assumes a spherical Gaussian noise model and attaches the noise variance σ^2 as an extra input. Figure 5.3(b) shows the result of the posterior mapping with noise information as an additional input. Figure 5.3(c) is the result of the fully probabilistic NeuroScale with the modified shadow target.

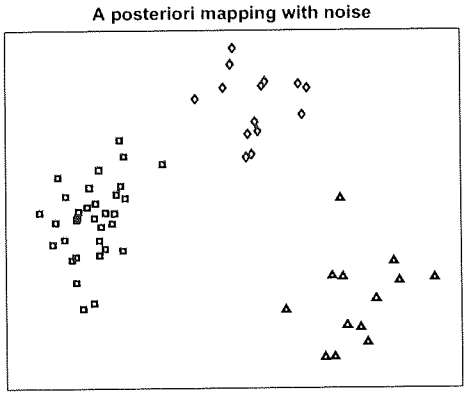
Except for the Figure 5.2(b) from modifying the dissimilarity matrix to use KL distance with the standard NeuroScale, all the projections show similar characteristics as they can separate the three clusters from each other.

5.3.2 Novel data projection

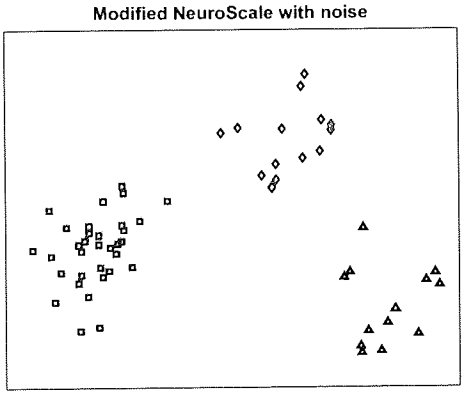
The advantage of using the NeuroScale approach with an embedded neural network is its ease of applying the trained networks to new data without the necessity of retraining. This section shows the ability of applying the standard NeuroScale and modified NeuroScale to new data. Figure 5.4(a) shows the novel data projection of the standard NeuroScale model and Figure 5.4(b) for the standard NeuroScale with the additional dimension. The modified NeuroScale using heuristic, posterior, and modified shadow target are shown in Figure 5.4(c), 5.4(d) and 5.4(e) respectively. All of them provide similar projections with good separation on all three clusters. However, Figure 5.4(d)



(a)



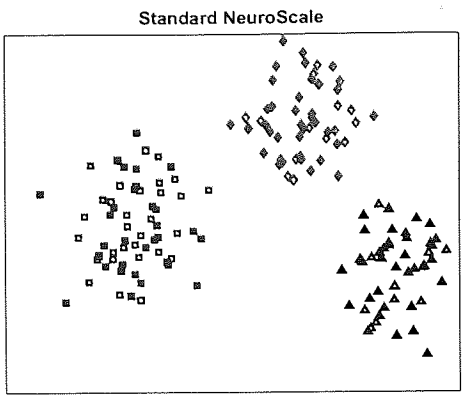
(b)



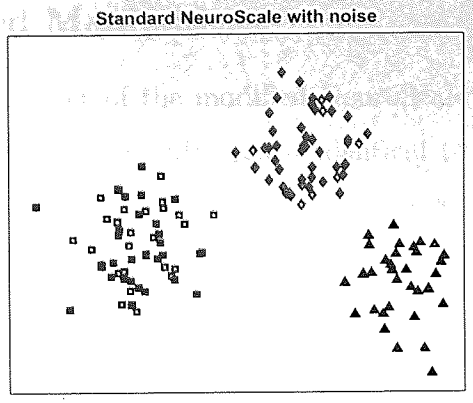
(c)

Figure 5.3: A synthetic example of three Gaussian clusters disturbed by noise. (a) The heuristic modified NeuroScale projection as in (5.4). (b) The probabilistic NeuroScale with uncertainty as an extra input and measuring input distance by symmetric KL divergence using the simple scaled conjugate gradient using cost function in (5.21). (c) using symmetric KL divergence with the RBF networks.

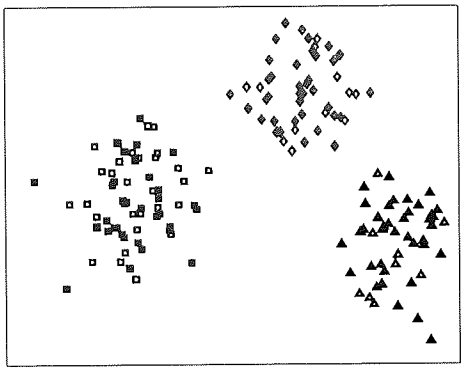
does not provide as a good projection of the test data as the other approaches. Even though it shows quite a clear separation in three clusters, there are some points from different clusters (red, blue and black) close together. The results showed that the modified algorithm can achieve a equivalent topographic visualisation representation in the case considered here of uniformly distributed noise.



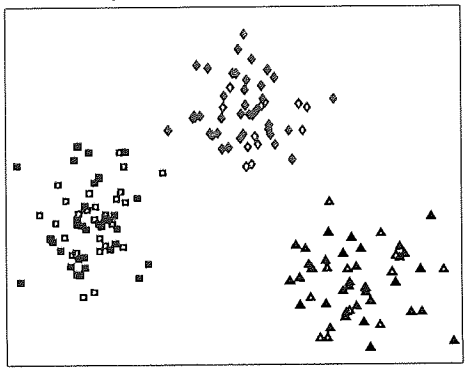
(a)



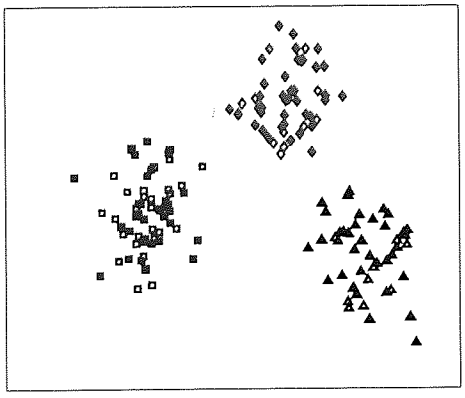
(b)



(c)



(d)



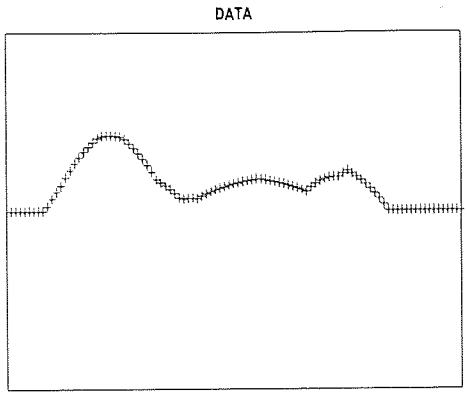
(e)

Figure 5.4: The 2-dimensional projections of novel data obtained by additional new sampling from the same pdfs used to generate the training data projected on (a) the standard NeuroScale (b) the standard NeuroScale with additional noise dimension (c) using the heuristic modified cost function (5.4) (d) using the modified cost function (5.21) optimised by scaled conjugate gradient (e) using the modified shadow target.

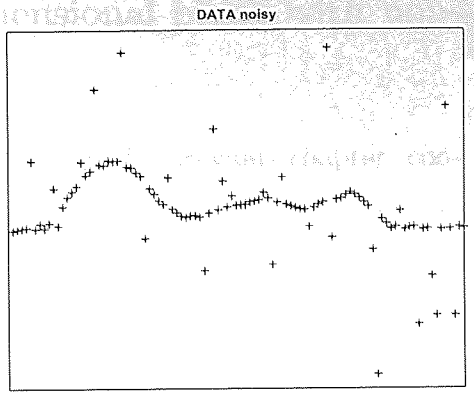
5.3.3 Synthetic Example of a Curved Manifold

The previous example does not show much improvement of the modified NeuroScale models. It only shows that the modified NeuroScale can provide almost identical to the NeuroScale given the data which has uniformly small noise. However, in the next example we will compare the performance of different NeuroScale approaches with non-uniformly distributed noise. This example is a simple 2-dimensional example with smooth curvature of 100 data points, as shown in Figure 5.5(a). Small Gaussian noise is applied to this data sample and about 10 data points are selected to be outliers, disturbed by larger noise, which is shown in Figure 5.5(b).

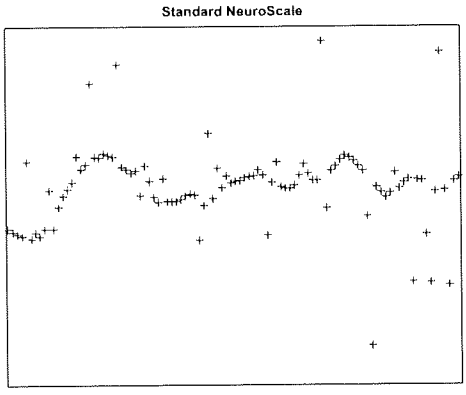
This synthetic example is a simple example to reconstruct noisy 2-dimensional mapping of the data to show effectiveness of removing noise using different NeuroScale projections. Different NeuroScale models were used to reconstruct the original data(Figure 5.5(a)). The results of standard NeuroScale, modified NeuroScale with and without noise are shown in Figure 5.5(c), 5.5(d), 5.5(e) respectively. The results show that the modified NeuroScale results gave better reconstruction of the data. It is clearly shown that the outliers distorted the smoothness of the remaining data points while trying to retrieve the noisy location of the outlier data. By contrast, the modified NeuroScale can retrieve the smoother curve of the non-outliers, more similar to the original data. The original STRESS(eq.(2.1)) is used to quantify the preservation level of the non-outlier data. The STRESS value of the standard NeuroScale model in Figure 5.5(c) is $\texttt{STRESS} = 0.065$. The STRESS of modified NeuroScale in Figure 5.5(d) is almost 0 and the STRESS value of 5.5(e) is 0.001. The results of the STRESS value also reveal the better reconstruction of the modified NeuroScale models over the standard model.



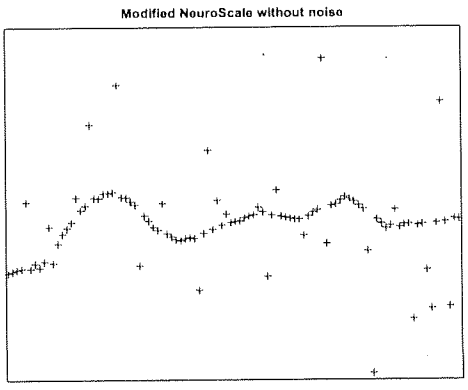
(a)



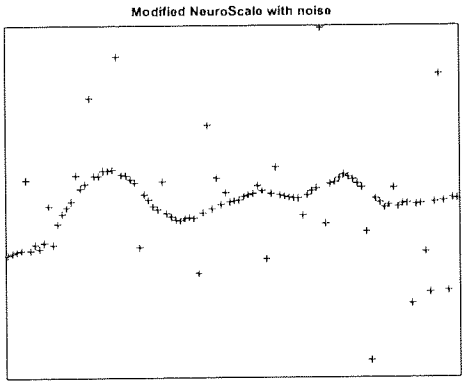
(b)



(c)



(d)



(e)

Figure 5.5: Another synthetic example using data generated in a two-dimensional manifold with curvature. (a) is the original data. (b) is disturbed by noise with outliers. (c) shows the result of reconstruction of the synthetic data by using the standard NeuroScale. (d) uses modified NeuroScale without noise as an extra dimension. (e) is a result of modified NeuroScale with noise.

5.3.4 Synthetic Example on a 2-dimensional plane with non-uniform added noise

This example is similar to the GTM synthetic example in the previous chapter, considering data defined in a 2-dimensional plane with 11 × 11 points and different levels of added noise used to induce distortion into the third dimension. The noise is measured by $\sigma_i^2 = \frac{1}{C_i}$, where C_i is the confidence of each data point measured by $C_i = d_{\text{from plane}} / \max(d_{\text{from plane}})$, where $d_{\text{from plane}}$ denotes the vertical distance a data point is removed from the plane in which the main data is constrained. In this example, the noise is not uniformly distributed for all data points. Some points are more disturbed by noise and the noise is attached as extra information into the NeuroScale approaches suggested previously. In the example shown in Figure 5.6, 20% of the data points (24 points) are selected to be outliers. Those points have very high variance added with $\sigma^2_{\text{outlier}} = 5$. The two-dimensional projection results are shown in Figure 5.6. Figure 5.6(a) shows the projecting result using the standard NeuroScale model while Figure 5.6(b) is the standard NeuroScale result with additional dimensionality of noise. Both projections show similar projections in which high-variance data points (outliers) are scattered away from the majority together with the distortion of data points with low added noise (high certainty data). The results do not preserve the original data structure well in which points are aligned evenly on the plane. On the other hand the modified model result for the heuristic approach is shown in 5.6(c) and Figure 5.6(d) for for the probabilistic approach. By visual perception, Figure 5.6(c) shows improved alignment of non-outliers compared to the standard models both with and without additional noise. Figure 5.6(d) clearly shows the best alignment of the data points. Non-outlier data are only marginally disturbed by the outlier points. The small misalignment are only due to the outlier data. This full probabilistic model shows the best alignment. The results show that using the probabilistic approach preserves the original structure in a more appropriate topographic manner.

To evaluate the modified NeuroScale, the original STRESS measure, using the Euclidean distance, is used to show the quality of structure preservation. However, only non-outlier data will be used to calculate this Stress measure since the idea is not to preserve the outliers but to preserve the structure of the rest. Therefore the

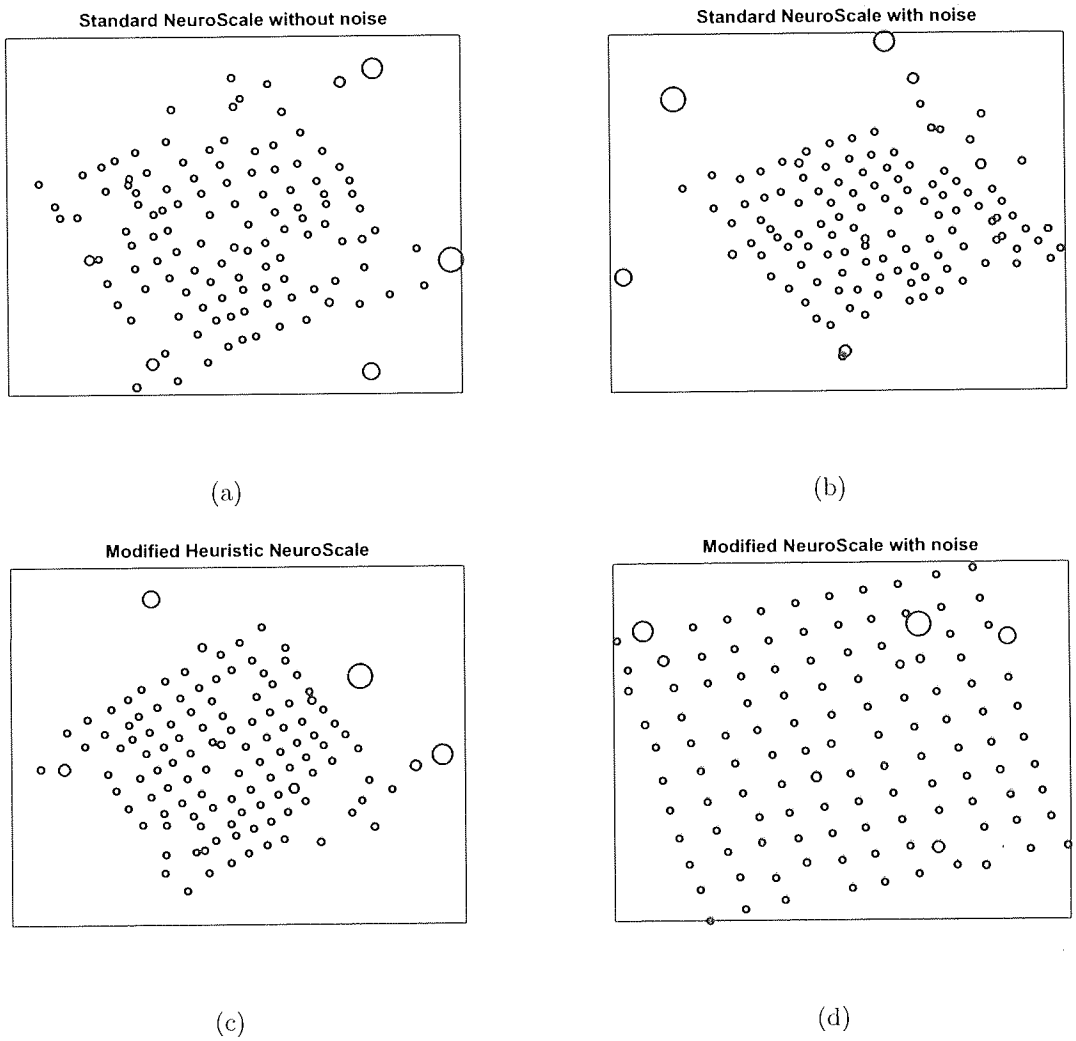


Figure 5.6: A synthetic example using data generated in a two-dimensional plane with 20% noisy outliers (a) the standard NeuroScale, (b) the standard NeuroScale with an extra noise dimension (c) using the heuristic modified cost function (5.4) (d) using the modified shadow target approach with a probabilistic model. The size of the symbols denotes uncertainty levels.

new STRESS is computed by ignoring those outliers. All NeuroScale approaches were trained on the data samples and the original stress function (2.1) was used to measure the error. Table 5.1 shows the STRESS value obtained by using (2.1). The number of outliers is chosen between 10 and 20 percent which are 12 and 24 points respectively. The outlier's variance is tested between 0.5 to 5 averaged over 50 training networks. The average is created from the mean over 50 training examples using different NeuroScale approaches, all of which were initialised by PCA.

Outliers		Average STRESS value(mean)				
Percent	Variance	Standard eq.(2.1)	Standard with noise eq. (2.1)	Prob. eq.(5.21)	Prob. with noise eq.(5.21)	Heuristic eq.(5.4)
10	0.5	25.23	27.94	22.38	22.20	**20.70**
	1.0	35.75	40.17	25.82	25.83	**24.86**
	1.5	52.49	56.69	32.46	**32.30**	33.01
	2.0	56.16	59.21	34.04	**32.47**	36.37
	2.5	63.67	64.93	41.79	**41.40**	44.53
	3.0	80.99	82.05	52.30	**51.67**	58.80
	3.5	101.35	104.38	61.93	**61.07**	72.45
	4.0	102.73	103.61	61.48	**61.20**	73.45
	4.5	121.74	122.30	**69.24**	71.23	90.26
	5.0	107.00	108.65	82.09	**78.77**	81.36
20	0.5	21.59	23.96	16.13	**16.03**	17.08
	1.0	48.78	52.44	**23.25**	23.73	33.40
	1.5	69.76	71.99	34.79	**33.81**	50.00
	2.0	63.73	66.77	37.32	**35.55**	40.46
	2.5	108.68	110.05	**48.80**	50.03	83.71
	3.0	123.85	124.35	53.90	**51.98**	98.02
	3.5	156.86	156.42	77.07	**75.13**	134.52
	4.0	174.93	175.25	98.87	**97.06**	140.16
	4.5	211.28	212.29	263.02	265.14	**157.13**
	5.0	257.58	256.80	323.03	322.24	**196.15**

Table 5.1: A table showing the average evaluated STRESS from different fractions and variances of outliers using different NeuroScale approaches, including the standard NeuroScale, the standard NeuroScale with noise, the fully probabilistic NeuroScale without noise, with noise and the heuristic NeuroScale. The best STRESS value of each row is highlighted in bold. The results show that overall the modified NeuroScale both fully probabilistic and heuristic way which incorporate uncertainty information gave better performance. (The standard models and modified models columns are separated by double lines).

From Table 5.1, the modified NeuroScale approaches show an improvement over

the standard NeuroScale using the STRESS measure. The heuristic approach is sometimes better than the probabilistic approaches, however, most of the time the fully probabilistic approach gave better performances. The error gradually increases as the σ^2 of outliers increase in all models except with 20 % outliers with $\sigma^2 = 4.5$ and $\sigma^2 = 5$ which shows a dramatic drop in the performance of the modified NeuroScale. This is due to some of the networks having very poor convergence which is due to local minima. However, if the outlier networks are ignored and the error is averaged by using median, instead of arithmetic mean, the result of the modified NeuroScale of the 20% outlier improves, the results are shown in Table 5.2

Outliers		Average STRESS value(mode)				
Percent	Variance	Standard eq.(2.1)	Standard with noise eq. (2.1)	Prob. eq.(5.21)	Prob. with noise eq.(5.21)	Heuristic eq.(5.4)
20	0.5	19.77	22.78	13.35	**13.19**	15.05
	1.0	42.29	46.19	**21.37**	21.89	28.03
	1.5	59.57	62.74	29.14	**26.91**	47.27
	2.0	57.58	59.87	32.39	**30.83**	34.39
	2.5	97.75	97.75	**44.19**	46.33	76.30
	3.0	118.60	118.84	**47.75**	47.84	84.72
	3.5	134.21	133.52	63.21	**57.43**	119.72
	4.0	162.31	162.41	73.37	**71.84**	128.35
	4.5	183.82	185.85	**97.42**	104.08	138.58
	5.0	210.31	210.54	117.45	**115.92**	155.66

Table 5.2: A table showing the average of evaluated STRESS by using the median instead of the mean average using different NeuroScale approaches. The best STRESS value of each row are highlighted in bold. The results show that overall the modified NeuroScale gave best performance. (The standard models and modified models columns are separated by double lines)

5.3.5 The real data sample with attached uncertainty

The same microarray data set with uncertainty which was used with the GTM model previously is explored using the modified NeuroScale. Applying modified NeuroScale to this data set, gives the result shown in Figure 5.7. For comparison, Figure 5.8 illustrates the result from standard NeuroScale. Data with different confidence values are shown in different sizes. Larger represent higher uncertainty. The results of the

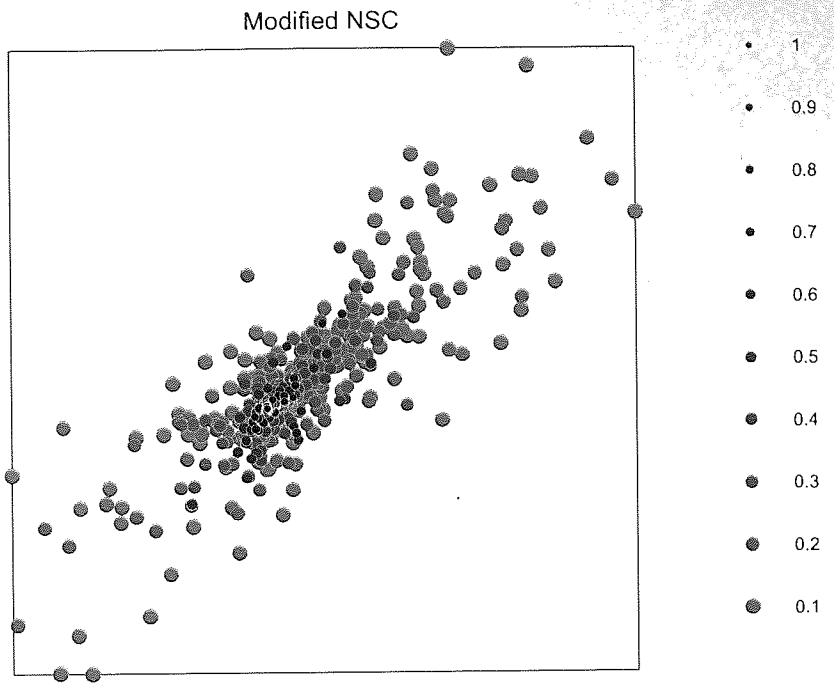


Figure 5.7: The resulting visualisation using modified NeuroScale.

modified models show the high confidence data point are packed together more than the standard models.

Similar to the GTM model, we will focus on the four genes grouped together in the standard model. Those genes are highlighted in Figure 5.9. Their expression values are shown in Figure 5.10. The highlighted gene is gene 'SCO5777' which give the highest expression level.

In modified NeuroScale, those four genes, which previously grouped together, become separated. Their locations in the new visualisation space are highlighted in blue colours shown in figure 5.11. Figure 5.12, Figure 5.13, Figure 5.14 and Figure 5.15 are the gene expressions of those genes with their new neighbours. The neighbours of the previously neighbouring genes are different from those in the modified. This is due to the effect from the confidence value attach to the standard NeuroScale. Although the four genes are spread away from one another, they are still in the bottom left corner of the picture as it was in the standard NeuroScale.

In standard NeuroScale, only highly expressed genes will come close to the edge and genes which have low levels of expression will be close to the centre. This has changed in this modified NeuroScale model. The standard NeuroScale model is better

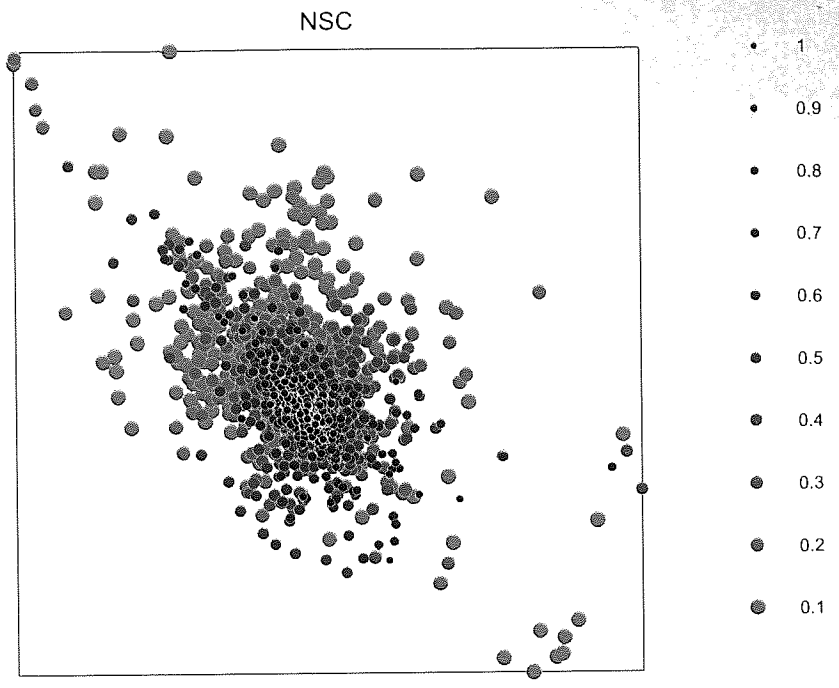


Figure 5.8: The resulting visualisation using standard NeuroScale.

at representing the location of the genes which corresponds to the levels of expression. This modified NeuroScale model is more likely to put higher expressed genes to the middle of the projection. The results show the impact of including the confidence level, or uncertainty using the modified NeuroScale model.

The interpretation of the results using this gene set requires more cooperation of the biologists who are the experts in the information given to verify which models are more useful and applicable for them.

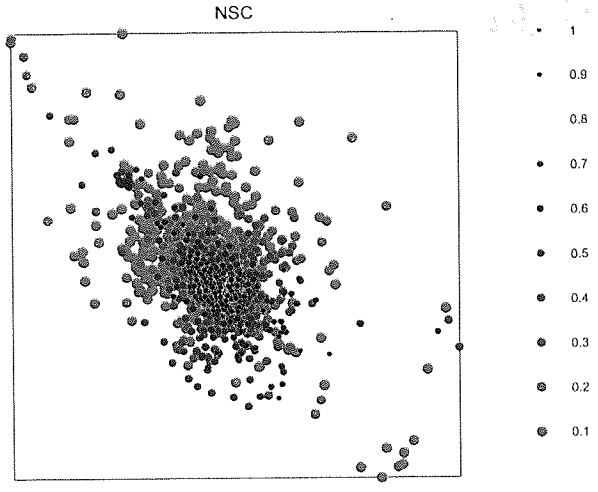


Figure 5.9: The resulting visualisation of the standard NeuroScale model, focusing on the blue circle region that we will use to investigate.

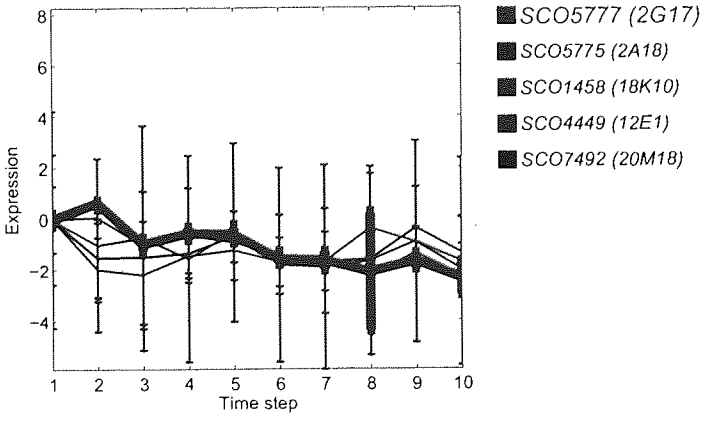


Figure 5.10: The expressions of the selected genes.

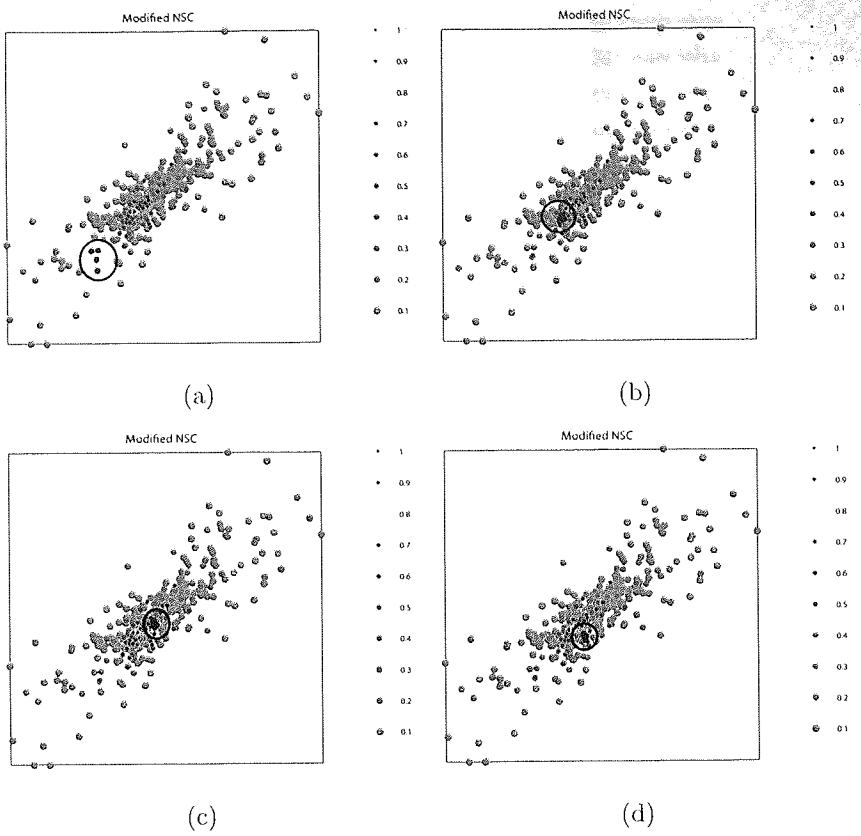


Figure 5.11: The resulting visualisations of modified NeuroScale of those genes that previously grouped together in the standard model. The new locations are highlighted in blue colours. (a) shows the location of 'SCO5777'. (b) shows the location of 'SCO5775'. (c) shows the location of 'SCO4449'. (d) shows the location of 'SCO5487'.

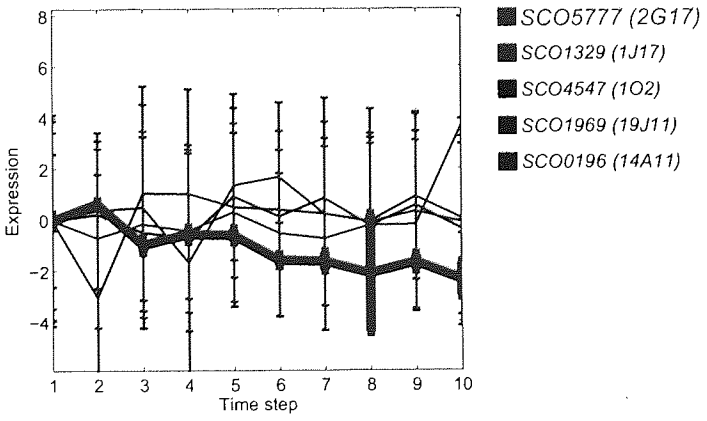


Figure 5.12: The gene expression profiles of gene 'SCO5777' and its neighbours in the modified NeuroScale.

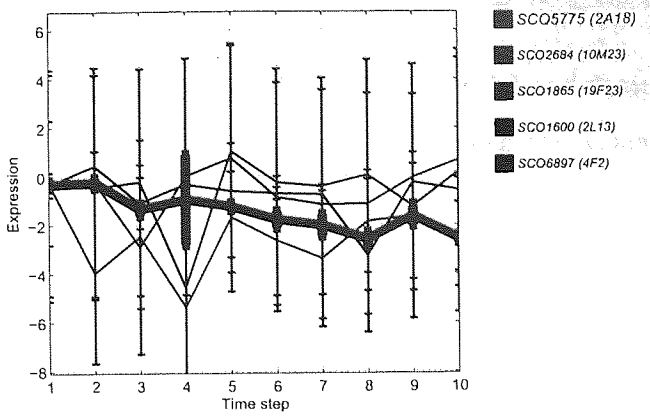


Figure 5.13: The gene expression profiles of gene 'SCO5775' and its neighbours in the modified NeuroScale.

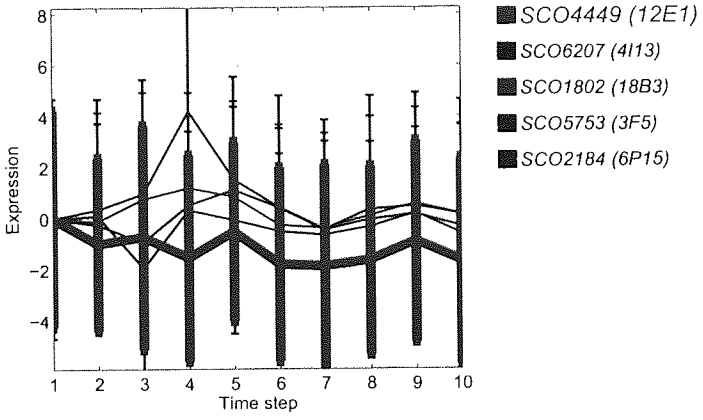


Figure 5.14: The gene expression profiles of gene 'SCO4449' and its neighbours in the modified NeuroScale.

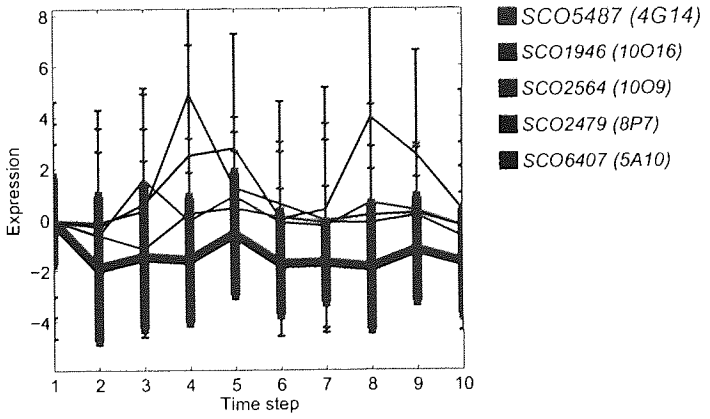


Figure 5.15: The gene expression profiles of gene 'SCO5487' and its neighbours in the modified NeuroScale.

5.4 Conclusion

This chapter has introduced a modified NeuroScale model and optimisation algorithm where data uncertainty is explicitly allowed to influence the visualisation. We have introduced a new probabilistic projection model based on preserving topographic properties between distributions rather than just data points.

We have also introduced a modified shadow target algorithm to optimise the new model more efficiently and have demonstrated its effectiveness on synthetic data problems, showing its superiority to the standard models. Our modifications have not diminished the ability of NeuroScale to produce generalisation maps to novel data.

However, the modified model still has some limitation of this probabilistic method. First, the latent space is assumed to have the same distribution as in the high-dimensional space, which are Gaussian with the same variance in the examples considered so far. However, this is quite optimistic since the projection of the distributions need not necessarily have the same distribution as the data space. In addition, it is prone to local minima problems in the optimisation. Therefore good initialisation is important to obtain meaningful projections. Using a linear method such as PCA to obtain the initial configuration is a common approach. There is still a potential for improvement of this new full probabilistic NeuroScale.

In the next chapter, we compare the probabilistic NeuroScale projection model with the probabilistic generative GTM model applied to the real-world problem of cancer prognosis gene lists, which has motivated much of this thesis.

Chapter 6

Cancer Prognosis Case Study

In the previous chapters, we introduced new approaches for visualisation models incorporating uncertainty. In this chapter the modified models are developed to analyse the van't Veer cancer prognosis problem. However, the cancer data set does not have uncertainty information explicitly attached to the data. Therefore in this chapter we will investigate different uncertainty estimates of the patient data.

6.1 Uncertainty Measures

For the developed models, some measurement of uncertainty is required for the visualisation. For the cancer prognosis case study, we consider each patient measurement vector $\boldsymbol{x}$ as a sample generated from an underlying noise model $\eta(\boldsymbol{x})$ giving rise to an implicit uncertainty in data. This uncertainty therefore needs to be estimated by assuming all the patient data is generated by the common noise model. We therefore use machine learning approaches to estimate the noise model assuming a Normal distribution of unknown parameters. It has been shown that the microarray data can be fitted by Gaussian models [11].

The uncertainty estimate used is based on two different error measures including the 'predictive error bar' [52] which is based on a single network and a committee [63] of networks which combines different networks.

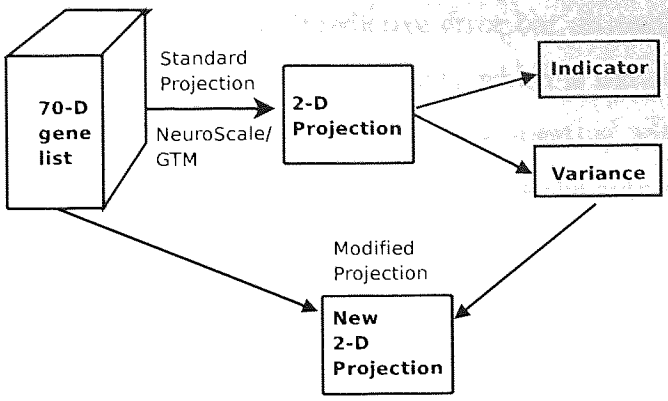


Figure 6.1: How the modified model is created from the original data space. The variance is trained by predictive error bars from the standard projection.

6.1.1 Predictive Error Bar

In this chapter, the van't Veer cancer prognosis example will be used for discussion. In this data set, the samples are sparsely distributed in a very high dimensional space. The number of data samples are almost the same as the dimensionality of the data. Therefore producing error bars based on the original 70 dimensional data is not appropriate. The error bars will be created based on the 2-dimensional projection as obtained from Chapter 3 using the original NeuroScale and GTM projections respectively. The diagram in Figure 6.1 shows the training phase of the modified models where the expected variance $\overline{\sigma}^2$ is estimated from the previous standard models. This datum-specific error bar will show the uncertainty level of how confident each patient belongs to either a good- or a poor-prognosis class. If the claim of the original van't Veer study was correct that List A can be used as a prognosis indicator, therefore the predictive error bars can be used as an error measure of each individual patient to be able to separate both patient groups in the modified visualisation results. That is to say, if the data is noise-free, then perfect classification results should be obtainable.

From Chapter 3, classifiers constructed from the visualisation space gave 2-component class membership probabilities between good and poor prognosis. In addition to the output, the classification error can be estimated. There are many different approaches for error estimation. The traditional approaches include Bayesian error bars [8, 53, 54] and Gaussian processes [97, 98]. However, the suggested way in this thesis for predicting the overall estimated uncertainties of data and model which is also suitable for

estimating the unseen data is by using 'predictive error bar estimation' [44, 52]. The predictive error bar is trying capture the uncertainty which is the result of model uncertainty from trying to classify the two patient groups together with the uncertainty of the data under assuming that the selected gene lists are the appropriate biomarkers for the breast cancer patients.

This approach uses the input $\boldsymbol{x}$ to estimate the variance $\sigma^2 = \|\boldsymbol{t}(\boldsymbol{x}) - \boldsymbol{y}(\boldsymbol{x})\|^2$ by using another neural network to estimate this data dependent variance. To avoid expected variances becoming negative, which is possible for the output of a linear-final-node neural network, $\log(\sigma^2)$ is used as the network targets. The outputs are therefore exponentiated to regain non-negative estimated variances. Variances of both good and poor prognosis indicators give the same output.

As previously discussed in the early chapters, the output data, $\boldsymbol{t}$ can be estimated by using RBF network modelling. In addition, the approximation of $\overline{\sigma}$, the expected error of the prediction, can also be estimated using another RBF network which minimises the square error of the training set to approximate the expected local confidence interval of the of the data set as:

$$\langle \|\boldsymbol{t}_{\boldsymbol{x}} - \langle \boldsymbol{t}_{\boldsymbol{x}} | \boldsymbol{x} \rangle\|^2 | \boldsymbol{x} \rangle, \tag{6.1}$$

where $\boldsymbol{x}$ is the input of the two-dimensional projections of the standard model, $\boldsymbol{t}_{\boldsymbol{x}}$ is the target data of this neural network. For the van't Veer data set, the output of this "error-predicting" network is therefore the log of the square error the network that used to find the good/bad indicators. The predictive error bar is trained on the log value of the error to avoid the negative outputs. The desired target values are $\boldsymbol{t} = [1 \quad 0]$ for a good prognosis group, and $\boldsymbol{t} = [0 \quad 1]$ for the other group, as mentioned earlier in Chapter 3.

The implementation of the models includes two interlocked neural networks which share the same input and hidden nodes but have two different final layer outputs: one for estimating the predicted output and the other is for estimating the log of the expected noise variance. The weight optimisation process has two stages. The first stage determines the weights to minimise the error to fit the desire target values, which represent the prognosis indicator. The second stage keeps the hidden units the same as

before but with another set of output weights redetermined to approximate the square error, σ^2 of the previous network.

6.1.2 The committee network

In the previous method, the RBF models are used to estimate training and validation errors using a single predictive error bar. The disadvantage of the previous method is that it is dependent on the model selection for the predictive variance. However, choosing any single network over the training data does not guarantee the better generalisation. Instead of discarding the rest we perform here a committee averaging of models [63]. The combination of different models will reduce the bias of having different random starts of the networks and reduces the overfitting or local minima, from any single network. This method allows the moderation of any network which does not agree with the majority. Committee averaging reduces the variance across models proportional to the number of models if the errors of the models are uncorrelated. The prognosis results are obtained by averaging the outputs across all models,

$$\mathbf{Y}_{COM} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{y}_i, \tag{6.2}$$

where $\mathbf{Y}_{COM}$ represents the averaged output from M different models and y_i is the output of networks. However, the weighted average can also be used to moderate those outputs obtained from outlier networks that do not agree with the majority [44, 77]. Therefore, the committee output can be obtained by:

$$\mathbf{Y}_{COM} = \sum_{i=1}^{M}\alpha_i\mathbf{y}_i, \tag{6.3}$$

where α_i is the weight of model i and constrained by $\sum_{i=1}^{M}\alpha_i = 1$. The weight of each model can be obtained by either using the prior knowledge or the correlation level of each network estimated using the variance of each model (σ_i^2):

$$\alpha_i = \frac{\sigma_i^{-2}}{\sum_{j=1}^{M}\sigma_j^{-2}}. \tag{6.4}$$

Similarly the committee variance can be obtained by:

$$\sigma^2_{COMM} = \sum_{i=1}^{M}\alpha_i\sigma_i^2. \tag{6.5}$$

Replacing this equation with (6.4) gives

$$\sigma^2_{COMM} = \frac{M}{\sum_{i=1}^{M} \sigma_i^{-2}}, \tag{6.6}$$

where M is the number of models. This result is the harmonic mean of variances across all models.

The predictive error bars are obtained by optimising another RBF model as in the previous section to estimate the conditional variance of the committee, as

$$\overline{\sigma}^2_{COMM} = < \sigma^2_{COMM} | \boldsymbol{x} >, \tag{6.7}$$

which is the predictive error bar across the committee networks.

6.1.3 Prognosis Indicators with uncertainty

From the two-output classifier, the first output is for the good-prognosis and the other for the poor-prognosis group. The higher the value of the first output, the more likely that patient will be diagnosed as good-prognosis patient and less likely to be diagnosed as poor-prognosis. Therefore, showing a single value is sufficient. The upper and lower bars will represent the predicted error bars of the network, defined by:

$$\begin{aligned} U &= G + \overline{\sigma} \\ L &= G - \overline{\sigma}, \end{aligned} \tag{6.8}$$

where $G = y(1)$ is the good prognosis indicator which is equal to the first output of the network. U and L are the upper and lower bounds of the estimated error. The new network complexity is determined similarly to the complexity of the main classifier.

6.2 Modified NeuroScale applied to the van't Veer data set

We first present predictive classification results using RBF models, showing the patient-specific error bars on each predicted prognosis value. The error bar estimation techniques from the previous section represent the level of certainty of the indicator resulting from a classifier superimposed on the standard NeuroScale model in chapter

3. The predictive error, σ can be estimated. Therefore these results can then be employed in our probabilistic visualisation studies where we will show that patient-specific uncertainty influences the global visualisation maps.

6.2.1 Single predictive error bar

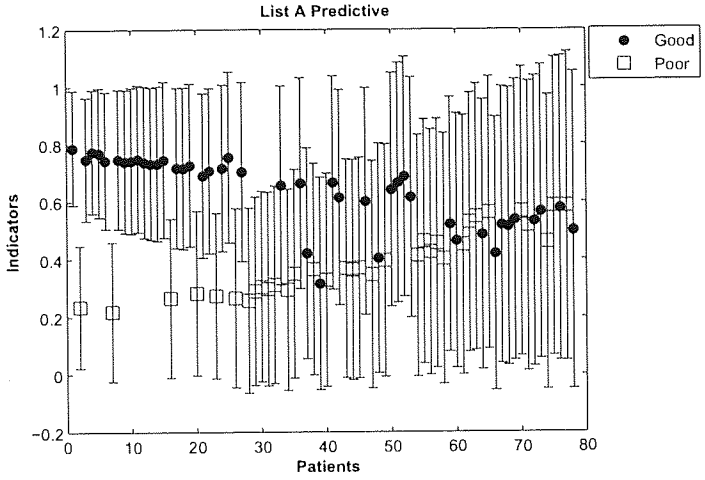
The good prognosis indicator ($y1$) is evaluated using the error bars obtained from using a single predictive error bar as in (6.1). The results of the predictive error for each patient ranked by the level of uncertainty are shown in Fig. 6.2(a) and 6.2(b) for List A and List B respectively. The patient-specific prognosis score is plotted in ascending order of σ values.

Both figures show that patients with high predicted prognosis of being either good- or poor-prognosis have slightly small variance compared to the patients with marginal predictions of being either good/poor prognosis. Very few patients do not have the error bar crossing the random threshold, 0.5.

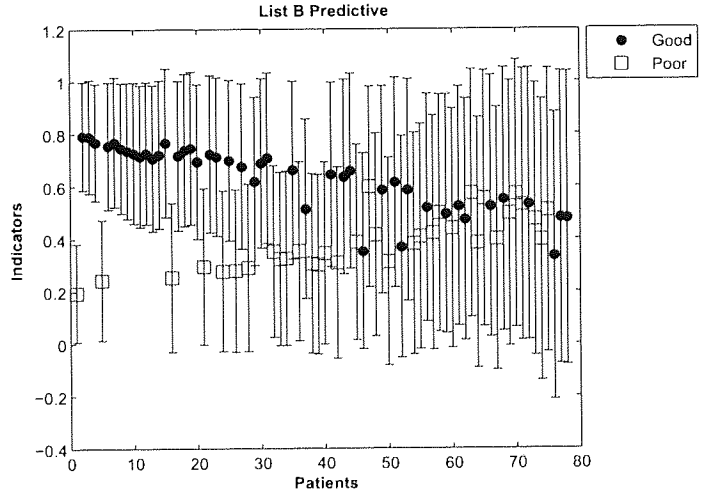
Both gene lists have the same magnitude of error bars to each other despite the same patients not always sharing similar levels of uncertainty between the two gene lists. The results show that neither gene list gives high confidence to any patients' prognosis predictions. This is consistent with some recent studies of this problem [24, 25]

6.2.2 Modified NeuroScale results

We now develop our earlier visualisation models to examine to what extent uncertainty influences topographic visualisation maps. The error bars therefore will be incorporated in the modified NeuroScale model. Figure 6.3(a) shows the result of the modified NeuroScale model using List A and Figure 6.3(b) shows the result using List B using the single predictive error bars as in (6.1). According to the previous section, patients with high indicators usually have lower variances. In the result of the modified NeuroScale map, both figures obviously separate a group of good prognosis patients to one side with slightly smaller variances than the remaining patients. However, almost all poor prognosis patients cannot be separated from the remaining good prognosis patients and form another cluster side. Both figures reveal quite similar structures. $P54$ is still shown as a significant outlier.



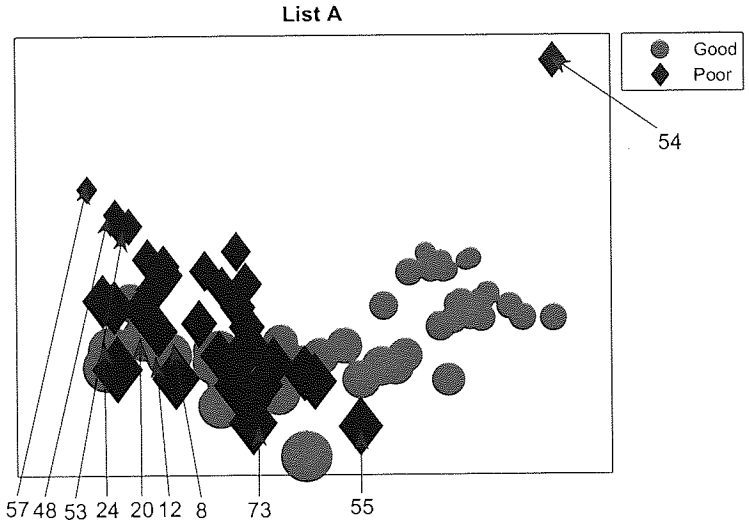
(a) List A.



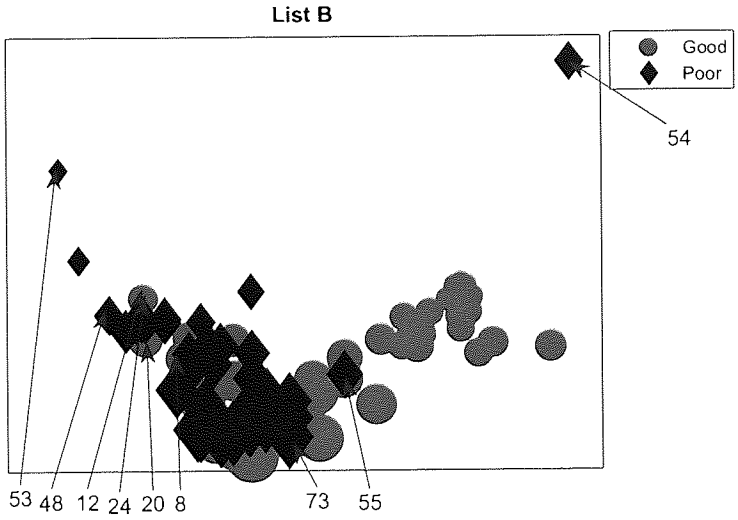
(b) List B.

Figure 6.2: The predictive output of the classifier from the standard NeuroScale projection as in (6.1) using (a) List A, indicating good prognosis value for each patient, ordered by the level of predictive error bar, and (b) List B. Solid circles represent good prognosis patients while open squares represent poor prognosis patients.

Both figures reveal that $P53$ is confidently misclassified with its intrinsic outlying gene structure, which is confirmed from the standard models. The visualisation of four good prognosis patients, $P8$, $P12$, $P20$, $P24$, still show that they project far from the good prognosis cluster. The differences between the standard and the modified models are larger separations between some high confidence good prognosis patients and poor prognosis patients. However, no patients give very high confidence. It can be interpreted that these gene lists give low confidence in identifying the correct prognosis group for these 78 patients using either gene list. The ability to separate some good prognosis patients confirms the prognosis ability of List A as mentioned in the van't Veer study that this list can filter the good prognosis patients that do not need to undergo the adjuvant treatment. However, that gene list is not unique for this prognosis capability as they claimed. Moreover, the literature has not mentioned the confidence level of the patients who could be successfully classified.



(a) List A.

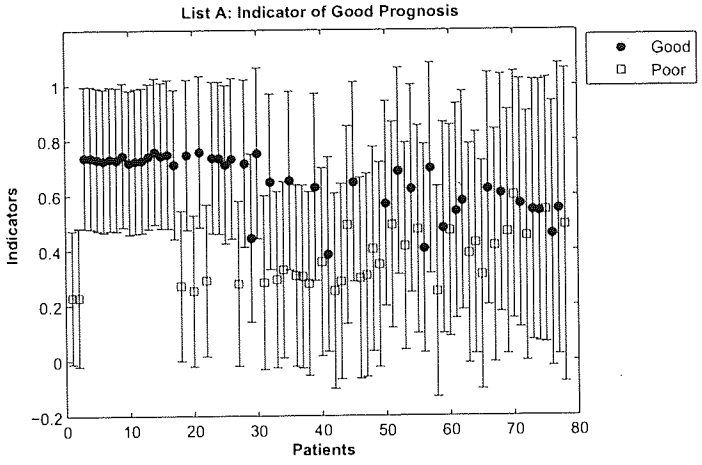


(b) List B.

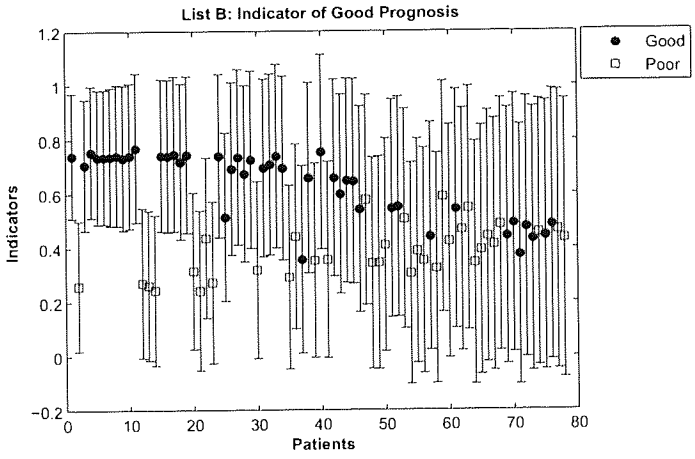
Figure 6.3: Modified NeuroScale results using single network uncertainty. Both figures obviously separate some higher confidence good prognosis patients to one side but most of these patients have high variance. The size of the symbols reflects the magnitude of the predicted uncertainty for that patient. Circles represent good prognosis patients and diamonds represent poor prognosis patients.

6.2.3 Committee predictive error bar results

For comparison we repeat the patient visualisation with our modified NeuroScale results, but using the revised uncertainty values from a committee of network models as in (6.7). RBF models with different complexities are used.



(a) List A



(b) List B

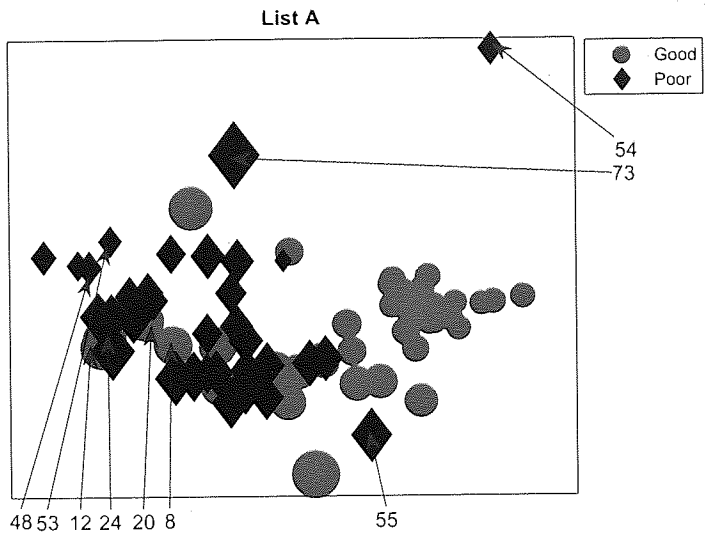
Figure 6.4: The predictive output of the committee classifier from the standard NeuroScale projection using (a) List A and (b) List B indicating good prognosis value for each patient. with the predicted committee variance per patient superimposed. Closed circles represent good prognosis patients and opened squares represent poor prognosis patients.

Figures 6.4(a) and 6.4(b) show the predictive error bar results using a committee for each patient based on averaging 40 committee networks. The predictive error bar is

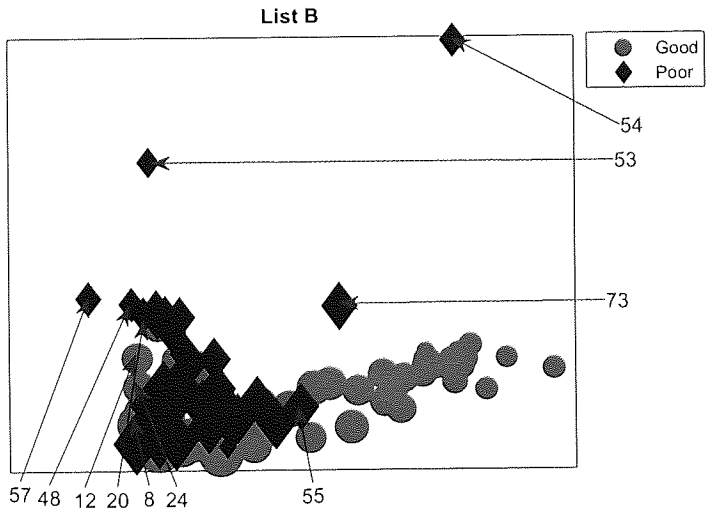
estimated by using another RBF network to produce results of the committee average error bars. The number of committee models is not a very sensitive parameter in this data set. Higher numbers of models do not significantly improve the results [77]. Most networks are correlated; however the average helps to remove any outlier network which had bad initial configuration. Few poor prognosis patients can be correctly classified into their group but not so significantly. Additionally, almost no patients have a small enough uncertainty to reliably locate them into one of the prognosis classes.

6.2.4 Modified NeuroScale results using the van't Veer data set and committee-predictive uncertainty.

Results of modified NeuroScale but using the predicted uncertainty from committee averaging are shown in Figure 6.5. Both results using List A and List B show quite similar results to the previous results of the single networks. The two projections are still showing similar results revealing the equivalent prognosis ability of the two gene lists. $P73$ who is not shown to be an outlier using the single network predictive error bar becomes separated using both gene lists. This reflects the bias that may occur by using just a single predictive model. The new projections confirm that some high confidence patients can be separated but there are still many patients with higher variance who remain unclassifiable.



(a) List A.



(b) List B.

Figure 6.5: Modified NeuroScale using a predictive error bar estimated with committee averaging attached to each patients. (a) denotes the projection of the van't Veer data set using List A. (b) is using List B. The results are similar using both gene lists.

6.3 Modified GTM applied to the van't Veer data set

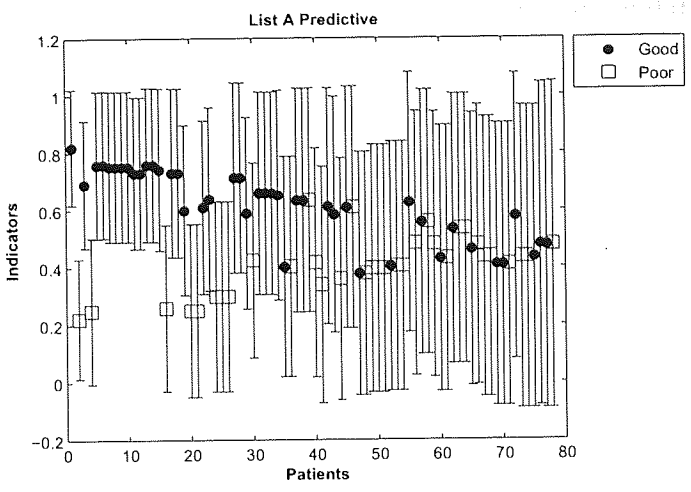
We have just illustrated topographic visualisation results for a projective model which incorporates uncertainty. The same idea of incorporating variances can be applied to the GTM model. The error bar can be obtained by using both single and committee networks based on the standard GTM model. We now illustrate the results of a *generative* model using the modified GTM structure as described in Chapter 4.

6.3.1 Single predictive error bar

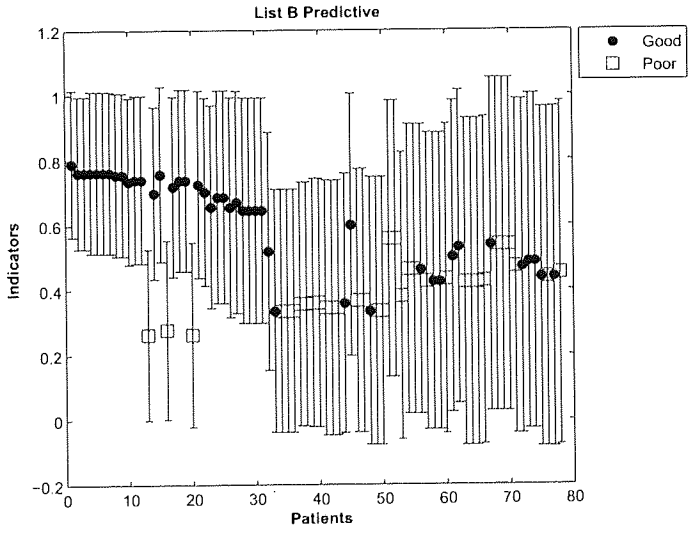
First, we determine predictive error bar estimates as in (6.7) using the level of certainty of the indicator resulting from a classifier superimposed on the standard GTM in chapter 3. The data together with the error bar are projected in Figure 6.6(a) and 6.6(b) using List A and B respectively. Levels of centainty are high. Most patients have error bars crossing the 0.5 boundary.

6.3.2 Modified GTM results using the single predictive error bar

The visualisation results using the patient-specific variance predictions obtained from the predictive error bars and used in the modified GTM are shown in Figure 6.7(a) and Figure 6.7(b) using List A and List B respectively. Compared to the original GTM in chapter 3, both projections produce very similar structures, with a 180 degree rotation of the image for List B. From both gene lists, patients with small σ^2 are mainly good-prognosis patients with very few patients from the poor prognosis group. Specifically, $P53$ from List A is now projected in the boundary of the two patient groups, while it was projected to the edge of the poor prognosis side. Additionally, the boundary where two prognosis groups separate cannot be easily identified if the labels of two patients groups are not used. Figure 6.7(a) and Figure 6.7(b) show the location where it is occupied by multiple patients. There are a few locations in both figures where the many patients are projected to which reveal big overlap of patients in the middle



(a) List A.



(b) List B.

Figure 6.6: The predictive outputs of the classifier from the standard GTM projection using (a)List A. (b) is using List B. Solid circles represent good prognosis patients while open squares represent poor prognosis patients. The results show higher misclassification rate and fewer high confidence patients compared to the NeuroScale results.

of the projection. Obviously by using the projections alone about 20% of those 78 patients cannot be classified with any level of certainty. Furthermore, on List A, $P12$, $P20$, $P24$ are obvious misclassified good prognosis patients. While $P4$, $P8$ are obvious misclassified good prognosis patients for List B. Those are similar to the standard model. $P54$ becomes a more obvious outlier for List B compared to the original GTM. On the other hand $P53$, normally an outlier, is not an outlier in this figure although this patient has a large variance.

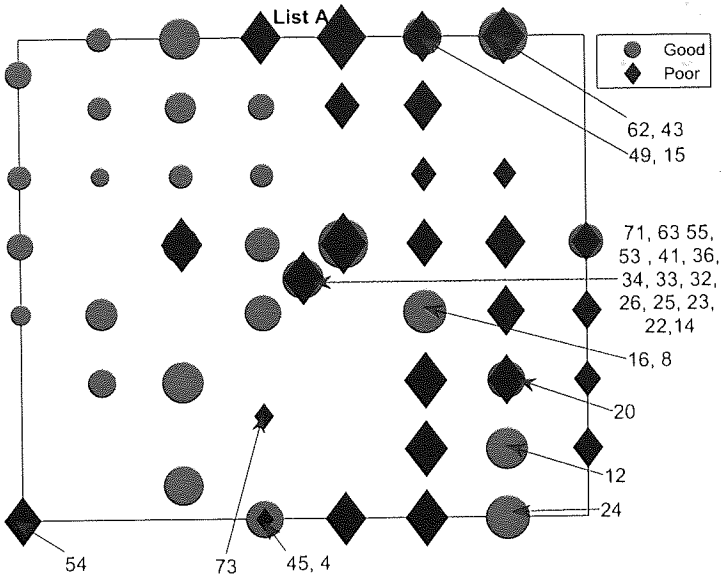
6.3.3 Committee predictive error bar results

Similar to the NeuroScale experiments, committee networks were used to re-estimate the variance by using the committee networks as in (6.7).

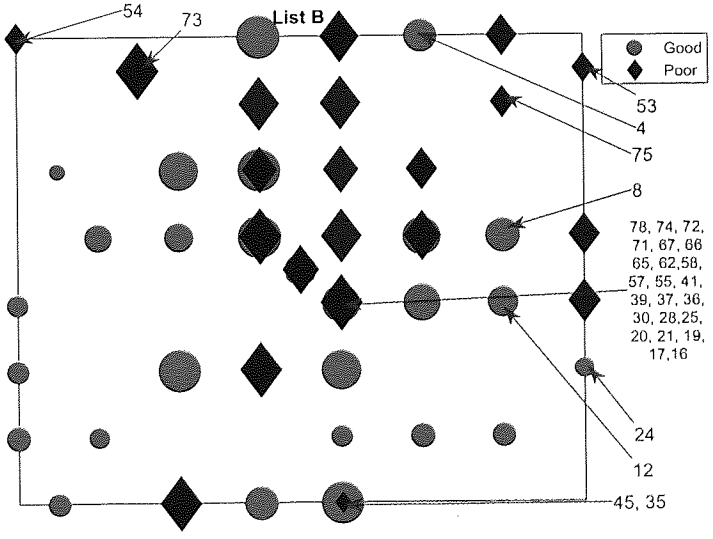
From the original GTM projection from Chapter 3, different numbers of RBF classifiers with different hidden units were used to produce the average output and variance of the GTM models. The results are shown in Figure 6.8. Circles and squares in Fig. 6.8(a) and 6.8(b) represent good-prognosis and poor-prognosis patients respectively, , sorting by the level of variance in an ascending order. Both figures reveal similar interpretations to the ones using the single predictive error bars: Most patients are unclassifiable, if the level of uncertainty is taken into account.

6.3.4 The modified GTM results using the van't Veer data set and committee-predictive uncertainty

The estimated variance from the committee was used for the modified GTM. The results are shown in Figure 6.9(a) and 6.9(b) using List A and List B respectively. The structure of the projections are similar to the ones using the single predictive error bars however, there are fewer overlapped between data points. Poor prognosis spread in a few more locations compared to the single predictive error bar. There are a few differentiation between However, the low variance patients can not be separated from each other. It is shown that the different uncertainty levels give some impacts to the GTM projection.

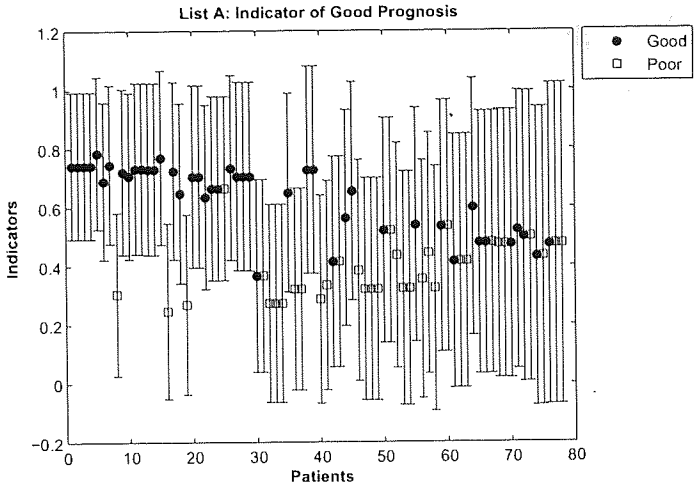


(a) List A.

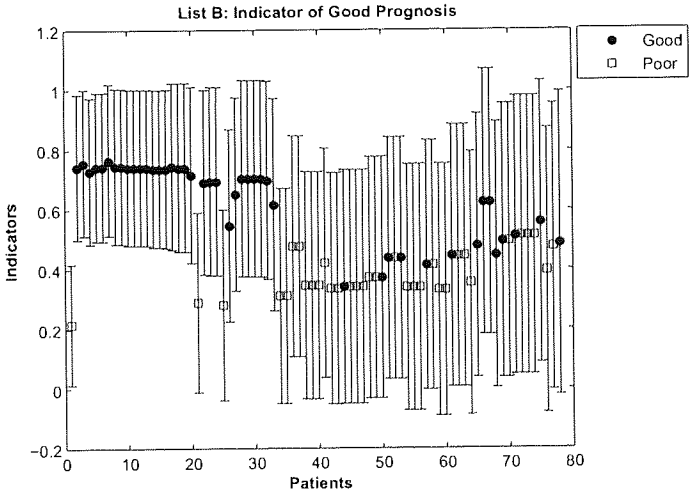


(b) List B.

Figure 6.7: Modified GTM visualisation using the single-model predictive error bar for each patient. (a) denotes projection of the van't Veer data set using List A. (b) is using List B. $P53$ is no longer an outlier with the modified GTM.

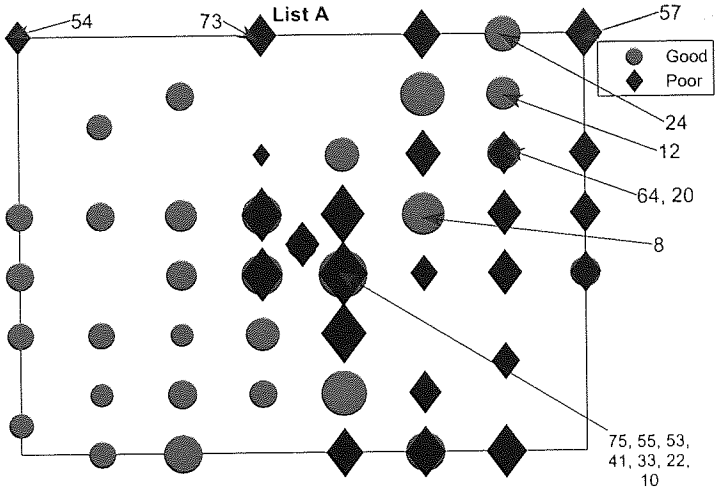


(a) List A.

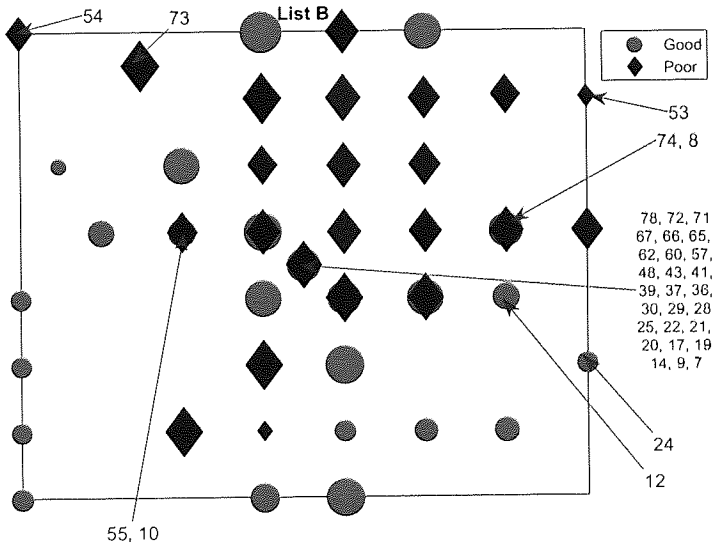


(b) List B.

Figure 6.8: The predictive outputs of the classifier from the standard GTM projection using (a)List A. (b) is using List B estimated with committee averaging. Solid circles represent good prognosis patients while open squares represent poor prognosis patients. The results of the committee networks show high reduction of the error bars, however, the committee output give indicators of many patient close to 0.5.



(a) List A.



(b) List B.

Figure 6.9: Modified GTM visualisation using the committee average predictive error bar for each patient. (a) denotes projection of the van't Veer data set using List A. (b) is using List B.

6.4 Projection of New Patients

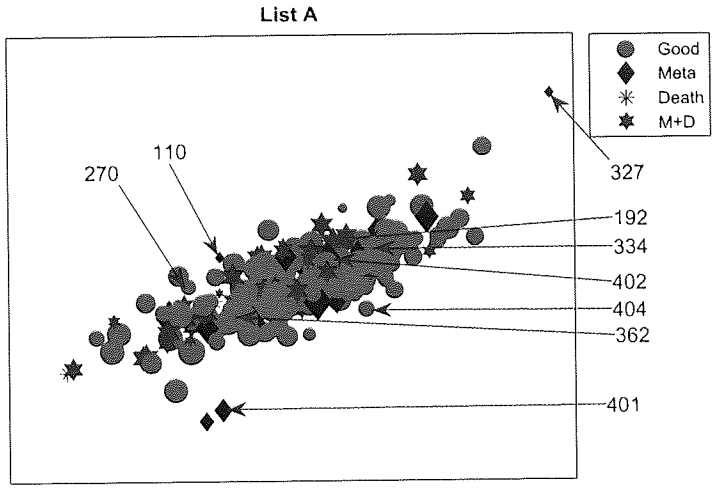
The benefit of both NeuroScale and GTM models is to have the capability to project the new patients who are not used in the training set to make new projections. This ability remains the same with the modified models. The 234 patients from the van Vijver study group were visualised using the modified NeuroScale and GTM obtained earlier. The noise variances used in this modified model are estimated by using the RBF networks trained earlier for variance estimation.

6.4.1 The projection of the new patients using modified NeuroScale

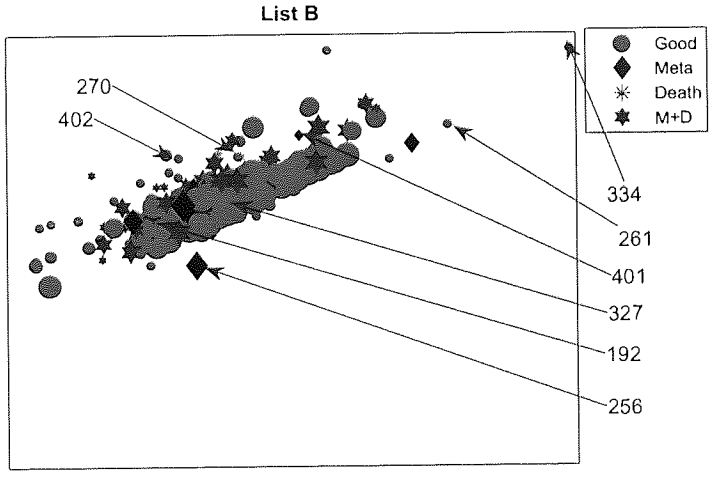
The modified NeuroScale model is used to map the novel 234 patients by the networks trained previously. The projection of the new patients using the NeuroScale models trained on the 78 patients and also using predictive error bar networks to estimate the variances of the new patients. The results of modified NeuroScale using single-predictive error bar networks used in Figure 6.3 for List A and List B.are shown in Figure 6.10. The results show very poor separation of these new 234 patients. Since many patients show quite different patterns compared to the original 78 patients, therefore the original NeuroScale projection created a sparse mapping of these new patients. As a result, many new patients project with high variance. However, Both gene lists show quite similar structures of projections. Only $P334$ and $P261$ seems to give quite high confidence using List B that can be correctly classified.

Variances obtained from committee networks

Figure 6.11 shows the modified NeuroScale projections using variances estimated by the committee average network as used in Figure 6.5.In this figure, some patients have very high variance in both gene lists. Those patients are removed from the projections. Figure 6.11(b) for List B shows a bigger variance of the data points than Figure 6.11(a) for List A. The result using List A is similar to the one using the single predictive error bar while the result using List B is different. The result of the latter is more like one dimension. Nevertheless, with these modified NeuroScale projections, both gene lists



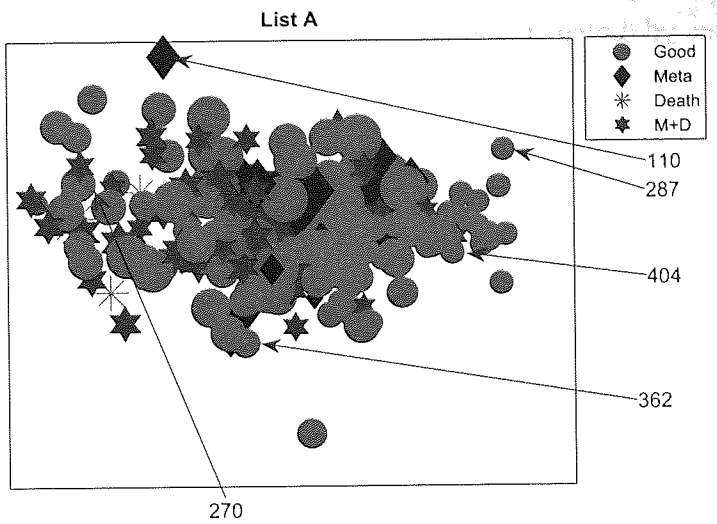
(a) List A.



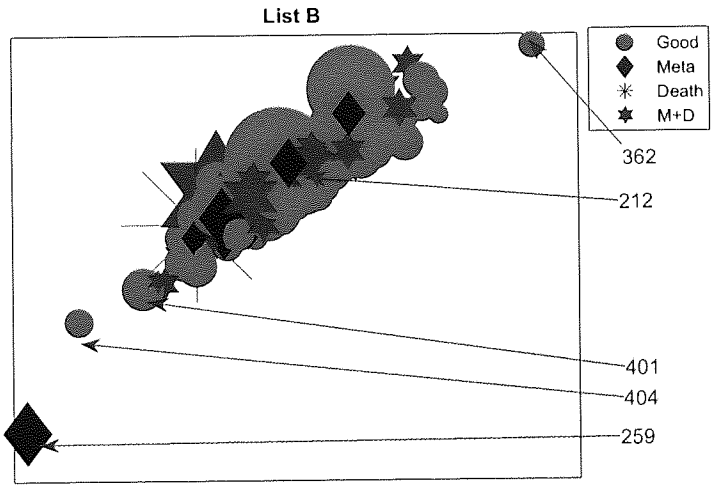
(b) List B.

Figure 6.10: The new 234 patients are projected by using the modified NeuroScale mapping trained in Figure 6.3. The variance are estimated by the same RBF networks as in Figure 6.3. (a) is the projection using List A. (b) shows the projection using List B. Both results show very high variance of the these new projections.

barely give any good confidence on any patients to be correctly classified. Both gene lists show that neither gene lists give enough useful information for separating patients into any prognosis group reliably.



(a) List A.

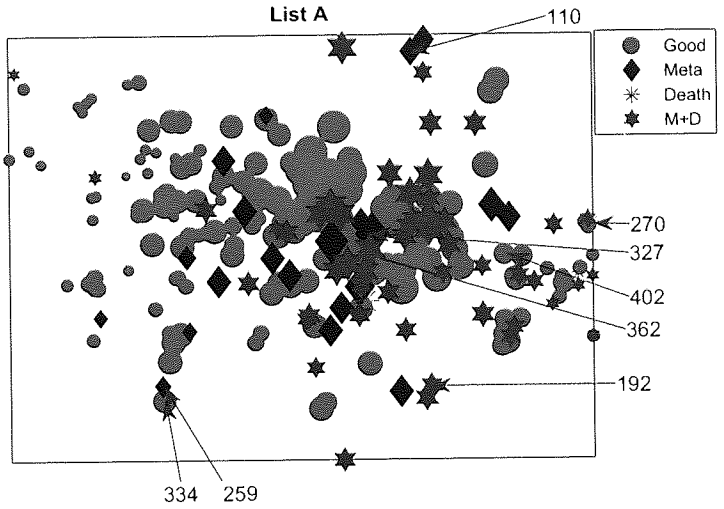


(b) List B.

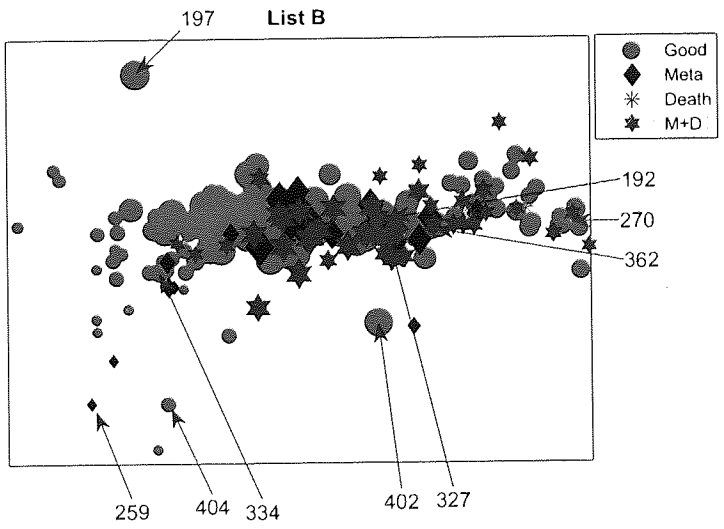
Figure 6.11: The new 234 patients are projected by using the modified NeuroScale mapping trained in Figure 6.5. The variances are estimated by committee average, the same RBF networks as in Figure 6.5.2 (a) is the projection using List A. (b) shows the projection using List B. Both results show very high variance of the new projections. Some patients are removed from this projection because of very high variance since those patients do not give meaningful results. Few patients have low variance hence only a few can be classified using these projections

6.4.2 The projection of the new patients by the modified GTM

Similar projections of the modified GTM can be made on the new 234 patients. The projection of the new patients whose variances are estimated by using the single-predictive error bar networks used in Figure 6.7 for List A and List B are shown in Figure 6.12. The results are slightly similar to the standard GTM, but there are a lot less overlaps between patients.



(a) List A.



(b) List B.

Figure 6.12: The new 234 patients are projected by using the modified GTM mapping trained in Figure 6.7. The variance are estimated by the same RBF networks as in Figure 6.7.

Variances obtained from committee networks

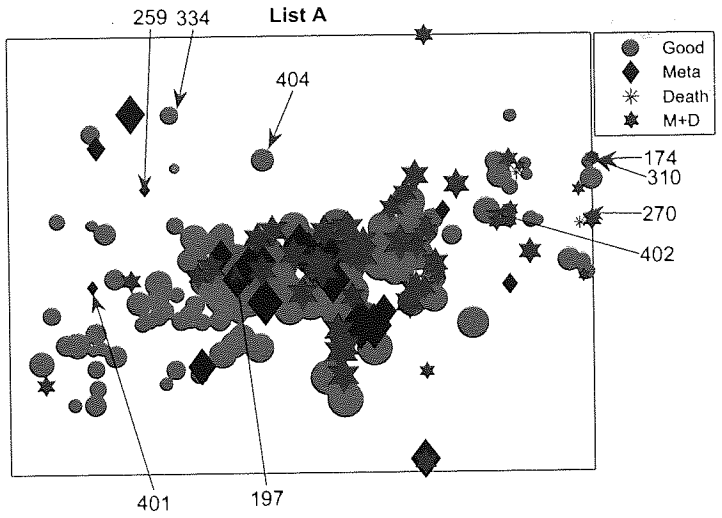
The results for modified GTM using the committee average predictive error bars are shown in Figure 6.13. The results are similar to the results obtained using the single predictive error bars because of high correlations amongst networks. However, both results show that the projections are less sparse but there are still more overlap of patients in both Figure 6.13(a) and 6.13(b), in the middle. Similar to all previous projection, the low confidence patients are projected to the interface of two prognosis groups. Almost no patients can be classified confidently into any prognosis groups.

6.5 Comparison between the probabilistic NeuroScale and the probabilistic GTM

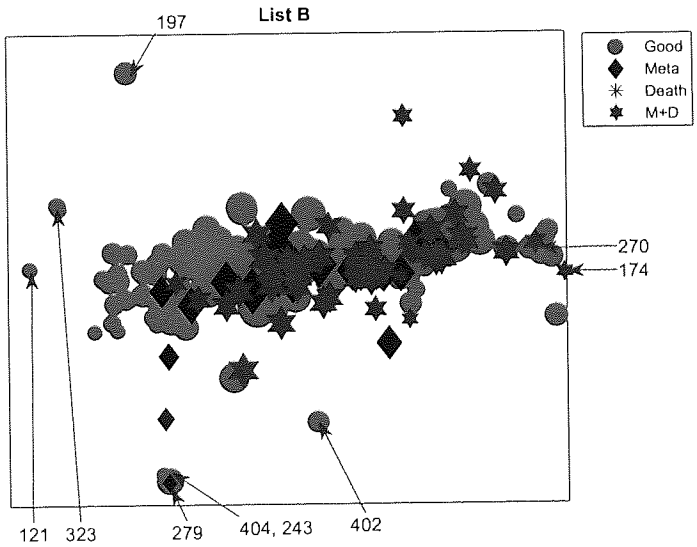
From the results obtained using the van't Veer data sample using both probabilistic NeuroScale and GTM, it is shown that the committee average give a slight differentiation of the projections compared to using only a single predictive model for the modified models. These show the consistency in produce the predictive error bars using different networks and show that no patients have very high confidence.

Nevertheless, both GTM and NeuroScale models show that some higher confidence good prognosis patients can be separated from the remaining patients. While patients with less confidence, both good and poor prognosis, cannot be classified into either good or poor prognosis. The projection of those low confidence patients tend to be overlapped between two prognosis groups. On the other hand, poor prognosis patients with high variance tend to have some outlying good prognosis patients in this groups. The good prognosis patients who appear in the low confidence poor prognosis patients are usually $P8$, $P12$, $P20$ and $P24$. They show intrinsic outliers in the structure of these patients using both gene lists.

For the GTM projection of the new 234 patients, some good prognosis patients are separated in the region where the good prognosis high variance patients in the previous projection(Figure 6.7 and 6.9), are. While for the NeuroScale projection, patients tend to have higher variances and there are few patients with low variance in the area where good prognosis patients used to be separated with the 78 patient projection.



(a) List A.



(b) List B.

Figure 6.13: The new 234 patients are projected by using the modified GTM mapping trained in Figure 6.9. The variances are estimated by committee averaging with the same RBF networks as in Figure 6.9. This figure and Figure 6.12 show very similar projections with fewer overlaps of patients.

6.6 Conclusion

This chapter has demonstrated the use of the modified visualisation models, both for NeuroScale and GTM, which have been proved to successfully retrieve the original structure of the synthetic data as shown earlier in chapters 4 and 5. The modified models are applied to the specific case study, the van't Veer data. The method of estimating confidence levels and projections were introduced in this chapter.

In addition, this chapter has also introduced approaches to estimate the level of uncertainty of each patient by using predictive error bars. The committee average output can give better certainty level of data if different networks created big disagreement. However, since the correlations across different models in the committee network, the improvements of the visualisation cannot be seen clearly.

The modified visualisation models helps show that few cancer patients can be confidently classified into either good or poor prognosis groups. They form a slightly better separation in the trained data than the original projection, for NeuroScale in particular. However, for the estimated variance, there are no patients who can be confidently identified reliably to be either good or bad prognosis patients. The modified models show different views of projections when confidence levels are incorporated.

In the context of the published literature, that there exists a preferential PGL of 70 genes, our studies indicate this to be false which has been shown by the visualisation techniques developed in this thesis. When accounting for uncertainty in patient feature vectors, no reliable separation of patients into prognosis groups can be made.

Chapter 7

Conclusion

This thesis has introduced new models for data visualisation for data with different levels of uncertainty. The new visualisation approaches helps to see a true underlying nature of the data set which involves the variations of data uncertainty. The original visualisation techniques can give false impressions of the nature of data resulting from involving unnecessarily high accuracy on the final projection for low dimensional visualisation. Creating a false interpretation can be dangerous if the data involves biomedical data which could give an incorrect understanding if applied to clinical diagnosis or prognosis. This thesis used the van't Veer data as an illustrative example taken from the domain of breast cancer prognosis throughout the thesis.

7.1 Visualisation for Biomedical Data

Normally in biomedical data, classifiers are mainly used to show the performance of biomarkers in order to diagnose or separate the data samples into pre-defined groups. However, the fact that those biomedical data contain uncertainty information has often been ignored. Performing supervised techniques on these data may give imprecise knowledge of the data. In addition, the pre-defined medical groups normally have clear separations neglecting the fact that some data samples in fact may lie in the overlapped region which is unclassifiable.

Data visualisation approaches applied in this thesis used two dimensional projections of biomedical data, of the van't Veer data in particular, to facilitate better un-

derstanding of the underlying structure of the data. Chapter 3 has performed different unsupervised visualisation techniques, including NeuroScale, LLE, SNE and GTM using the van't Veer data set as an example. The results have shown consistency among all techniques. Furthermore, this chapter has also introduced the concept of an alternative gene list with the same number of genes, List B, which is almost orthogonal to the proposed gene list, List A, introduced by the original study [90] which gives almost similar prognosis ability in this data set. Extracting a small subset of genes from small data samples in such a very high dimensional space as in the microarray data can create random correlations in the feature selection. As a result, the gene list obtained is not the best representation and not unique. The visualisation results throughout this thesis which compared the original and the alternative gene lists are consistent with Ein-Dor [25] where non-uniqueness of small gene lists is suggested. Many predictive gene lists with the same size can create equivalent predictability of the metastasis in breast cancer patients.

7.2 Probabilistic Visualisation

Chapters 4 and 5 have proposed approaches to attach data uncertainty to the standard GTM and NeuroScale. The modifications of the two models are different.

Due to the probabilistic nature of GTM, it is more straightforward to attach uncertainty levels by modifying the variance of the data points. Instead of using a constant σ^2 for all data points, in the modified GTM a datum-specific σ_i^2 proportional to the uncertainty level(C_i) of each individual data point i is derived from confidence. The proportion is controlled by the EM algorithm, therefore the results from different experiments give more similar results than NeuroScale.

Furthermore, the main disadvantage that can be seen obviously is that the GTM model needs to find the appropriate latent shape in order to gain a successful reconstruction of the visualisation. Furthermore, changing the number of Radial Basis Function centres also affects the results of the visualisation in GTM while this change does not affect the NeuroScale model as significantly.

On the other hand NeuroScale requires significant alteration to the basic model to

incorporate uncertainty because of its deterministic nature. Two approaches have been suggested for uncertainty attachment of the NeuroScale model. First, the heuristic approach, which is achieved by modifying the cost function and attaching the uncertainty level to the original STRESS measure by weighting the inter-dimensional distance by the confidence level of each data point. Even though it is quite a straightforward approach, it does not have a proper probabilistic explanation.

The other approach is the more complete probabilistic approach of NeuroScale which involves changing the distance measures of both input and output dissimilarity matrices. Instead of using the point-wise dissimilarity, such as Euclidean distances, the probabilistic approach uses the dissimilarity measures for probability, the Kullback-Leibler distance. Consequently, to accommodate this new dissimilarity measurement, a modification of the learning algorithm is needed. The modified shadow target algorithm has been introduced in this thesis. The shadow target now become $\hat{t}_i = y_i - \eta\sigma_i^2 \frac{\partial E}{\partial y_i}$, which has an additional σ_i^2 factor compared to the standard shadow target [Appendix D].

Synthetic examples have been use to demonstrate the performance of both modified GTM and NeuroScale. The modified models show equivalent performance to the standard models when small noise levels are uniformly applied to the data set. However, with non-uniformly distributed noise both modified GTM and NeuroScale have given better performance than standard models. The modified GTM model gives very consistent results while for the modified NeuroScale approaches, the heuristic NeuroScale gives more consistency among the results [15] than the full probabilistic approach especially with higher noise level. However, the full probabilistic approach gives better images of the underlying structure of the data given a good initialisation and choosing from many different experiments. To sum up, both modified visualisation models give better understanding of outlying structures of the data improving open the standard models as verified by the synthetic examples.

7.3 Intrinsic Uncertainty

Chapter 6 has proposed a way of estimating uncertainty levels of individual patients. Due to the low number of samples, the estimation of uncertainty is deduced from the previous visualisations to restrict any random correlation that may occur. Predictive error bars are estimated by training on the original 78 patients using both a single predictive network and committee average network. The committee network shows a reduction of error bars but still shows that most high certainty patients are those who are in the unclassifiable region.

In addition, the probabilistic visualisation results have shown that even when the uncertainty level has been attached to the data set, both gene lists have the same predictive power. The results of the original patients reveal that only some good prognosis patients with low confidence can be separated from the main patient set which shows a big overlap between the two patient groups. In addition, patients who have high confidence are those who are overlapped in the interface of the two prognosis groups hence they are therefore confidently *unclassifiable*. These results are consistent using both modified GTM and modified NeuroScale models. Moreover, both gene lists failed to separate the two prognosis groups of breast cancers patients in general, as shown in the visualisation of the new 234 patients. In this new patient set, almost no separations of good and poor prognosis patients can be made. As a result, the suggested gene lists do not contain enough information to *confidently* discriminate two patient groups and no single PGL performs better than the other. Therefore, the argument from the original literature which claimed prognosis ability of the reduced gene list is not correct. No reliable prognosis ability based on the gene list can be presented.

The prognosis should not be determined by purely looking at only gene expression, especially with only a small subset extracted from large number of human genes. Many genes are randomly correlated with patient survival. Using gene expression profiles can be one tool to help predicting the outcome of breast cancer patients. However, the predictive uncertainty should also be attached to the results. In addition, trying to classify the unclassifiable patients and claiming them to be a success of the method is unrepresentative.

Our conclusions have been derived on an analysis of just one biomedical example,

but are much more generic: publishing results relating to life-critical decision-support tools need to include at least an estimate of confidence in predictions, especially when based on small patient cohorts. Otherwise unfounded confidence in results could propagate to both scientific and public communities.

7.4 Directions of Future Research

The current probabilistic NeuroScale model is still prone to local minima. Chapter 5 has suggested that in the current form of NeuroScale, several runs of NeuroScale should be performed in order to retrieve the best performance especially for data with a high variance data points. The future can aim for a better version to produce more consistent results than the current modified NeuroScale.

In addition, the current form of NeuroScale used mainly thin-plate splines as basis functions, both in the synthetic data and the van't Veer data set simulations. Many types of basis functions can be used for comparisons between the basis functions in order to see the differentiation of the current probabilistic NeuroScale.

In addition, both modified NeuroScale and GTM models assumed added noise are data-specific Gaussian distributions. Each data point is assumed to have individual Gaussian noise attached to it as data-specific uncertainty. However, it is also possible to attach uncertainty to each dimension of each data point. For the breast cancer data, it could be regard as gene-specific uncertainty. For the extension to gene-specific uncertainty, the suggested models which currently use spherical Gaussian can be modified to use diagonal Gaussian noise models to incorporate different uncertainty levels in different dimensions. In addition, the future research direction can also aim for different types of noise models. Various probabilistic distributions are suitable for different types of data sets.

On the application side, throughout this thesis, only one application has been applied for experiments of the modified probabilistic visualisation. Future research should explore a range of applications throughout different fields. Different uncertainty measures can also be used for experiment to evaluate the reliability of that data set and the current modified visualisation models.

Bibliography

[1] http://www.cambridgebluegnome.com/bluefuse.htm.

[2] D. K. Agrafiotis and H. Xu. A self-organizing principle for learning nonlinear manifolds. *Proceedings of the National Academy of Sciences of the United States of America*, 99(25):15869–15872, 2002.

[3] Y. Balagurunathan, N. Wang, E. Dougherty, D. Nguyen, Y. Chen, M. Bittner, J. Trent, and R. Carroll. Noise factor analysis for cDNA microarrays. *Journal of Biomedical Optics*, 9(4):663–78, 2004.

[4] BBC. The future of cancer treatment. *http://news.bbc.co.uk/1/hi/health/medical_notes/c-d/590699.stm*, 2000.

[5] D. Beer, S. Kardia, C. Huang, T. Giordano, A. Levin, D. Misek, L. Lin, G. Chen, T. Gharib, D. Thomas, M. Lizyness, R. Kuick, S. Hayasaka, and J. Taylor. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature Medicine*, 8(8):816 – 824, August 2002.

[6] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral clustering. *Advances in Neural Information Processing Systems*, 16:177 – 184, 2004.

[7] E. Biganzoli and P. Boracchi. Old and new markers for breast cancer prognosis: the need for integrated research on quantitative issues. *European Journal of Cancer*, 40:1803–1806, 2004.

[8] C. M. Bishop. *Neural networks for pattern recognition.* Clarendon, 1995.

[9] C. M. Bishop, M. Svensén, and C. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[10] F. C. Boogerd, F. J. Bruggeman, J.-H. S. Hofmeyr, and H. V. Westerhoff, editors. *Systems Biology: Philosophical Foundations.* Elsevier, Oxford, UK, 2007.

[11] N. Brändle, H. Bishof, and H. Lapp. A generic and robust DNA microarray image analysis. *Machine Vision and Application*, 15:11–28, 2003.

[12] D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2(3):321–355, 1998.

[13] W. W. Cohen. *Aomputer Scinetist's Guide to Cell Biology.* Springer, Pittsburgh, Philadelphia USA, 1 edition, 2007.

[14] T. Cox and M. Cox. *Multidimensional Scaling*. Academic Press, 2 edition, 2001.

[15] D. D'Alimonte, D. Lowe, I. T. Nabney, and M. Sivaraksa. Visualising uncertain data. In *2nd European Conference on Emergent Aspects in Clinical Data Analysis, EACDA05*, Pisa, Italy, 2005.

[16] D. D'Alimonte, I. Nabney, D. Lowe, V. Mersinias, and C. P. Smith. Exploration of time-series gene expression data using advanced topographic representation. 2005. Sumitted to Bioinformatics.

[17] S. Davies and D. Seale. DNA microarray stochastic model. *IEEE Transactions on Nanobioscience*, 4(3):248–54, 2005.

[18] D. DeMers and G. Cottrell. Non-linear dimensionality reduction. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 580–587. Morgan Kaufmann, San Mateo, CA, 1993.

[19] R. Díaz-Uriarte and S. Alvarez de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1):3, 2006.

[20] A. L. Dixon, L. Liang, M. F. Moffatt, W. Chen, S. Heath, K. C. Wong, J. Taylor, E. Burnett, I. Gut, and M. Farrall. A genome-wide association study of global gene expression. *Nature Genetics*, 39(10):1202 – 1207, 2007.

[21] K. A. Easton, Douglas F.and Pooley, A. M. Dunning, P. D. P. Pharoah, D. Thompson, D. G. Ballinger, J. P. Struewing, J. Morrison, H. Field, R. Luben, N. Wareham, S. Ahmed, C. S. Healey, R. Bowman, K. B. Meyer, C. A. Haiman, L. K. Kolonel, B. E. Henderson, L. Le Marchand, P. Brennan, S. Sangrajrang, V. Gaborieau, F. Odefrey, C.-Y. Shen, P.-E. Wu, H.-C. Wang, D. Eccles, D. G. Evans, J. Peto, O. Fletcher, N. Johnson, S. Seal, M. R. Stratton, N. Rahman, G. Chenevix-Trench, B. G. Bojesen, Stig E.and Nordestgaard, C. K. Axelsson, M. Garcia-Closas, L. Brinton, S. Chanock, J. Lissowska, B. Peplonska, H. Nevanlinna, R. Fagerholm, H. Eerola, D. Kang, K.-Y. Yoo, D.-Y. Noh, S.-H. Ahn, D. J. Hunter, S. E. Hankinson, D. G. Cox, P. Hall, S. Wedren, J. Liu, Y.-L. Low, N. Bogdanova, P. Schurmann, T. Dork, R. A. E. M. Tollenaar, C. E. Jacobi, P. Devilee, J. G. M. Klijn, A. J. Sigurdson, M. M. Doody, B. H. Alexander, J. Zhang, A. Cox, I. W. Brock, G. MacPherson, M. W. R. Reed, F. J. Couch, E. L. Goode, J. E. Olson, H. Meijers-Heijboer, A. van den Ouweland, A. Uitterlinden, F. Rivadeneira, R. L. Milne, G. Ribas, A. Gonzalez-Neira, J. Benitez, J. L. Hopper, M. McCredie, M. Southey, G. G. Giles, C. Schroen, C. Justenhoven, H. Brauch, U. Hamann, Y.-D. Ko, A. B. Spurdle, J. Beesley, X. Chen, A. Mannermaa, V.-M. Kosma, V. Kataja, J. Hartikainen, N. E. Day, D. R. Cox, and B. A. J. Ponder. Genome-wide association study identifies novel breast cancer susceptibility loci. *Nature*, 447:1087 – 1093, 2007.

[22] P. Edén, C. Ritz, C. Rose, M. Fernö, and C. Peterson. "Good old" clinical markers have similar power in breast cancer prognosis as microaray genes expression profilers. *European Journal of Cancer*, 40:1837–1841, 2004.

[23] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman and Hall, 1993.

[24] L. Ein-Dor, I. Kela, G. Getz, D. Givol, and E. Domany. Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics*, 21:171 – 178, Jan. 2005.

[25] L. Ein-Dor, O. Zuk, and E. Domany. Thousands of samples are needed to generate a robust gene list for prediction outcome in cancer. *PNAS*, 103:5923 – 5928, April 2006.

[26] D. Ettinger, G. Bepler, R. Bueno, A. Change, T. Chang, L. Chirieac, T. D'Amico, T. Demmy, S. Feigenberg, and F. Grannis. National Comprehensive Cancer Network(NCCN). Non small lung cancer clinical practice guidelines in oncology. *Journal of National Comprehensive Cancer Natw.*, 4:548–582, 2006.

[27] C. H. Ford. Personalised medicine: Fantasy or a realistic goal? *Kuwait Medical Journal*, 36(4):247–249, 2004.

[28] X. Geng, D. C. Zhan, and Z. H. Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transaction on Systems, Man, and Cybernetics, Part B Cybernetics*, 35:1098– 1107, Dec 2005.

[29] N. Haan and G. Snudden. Microarrays in the real world: Image analysis. *Biopharmaceutical Review*, pages 44–47, 2004.

[30] S. Hautaniemi, O. Yli-Harja, J. Astola, P. Kauraniemi, A. Kallioniemi, M. Wolf, J. Ruiz, S. Mousses, and O. Kallioniemi. Analysis and visualization of gene expression microarray data in human cancer using self-organizing maps. *Machine Learning*, 52:45–66, 2003.

[31] S. Haykin. *Adaptive and learning systems for signal processing, communications, and control.* Wiley, 2000.

[32] R. Herzallah. *Exploiting uncertainty in Nonlinear Stochastic Control Problems.* PhD thesis, Aston University, October 2003.

[33] M. A. Hibbs, N. C. Dirksen, K. Li, and O. G. Troyanskaya. Visualization methods for statistical analysis of microarray clusters. *BMC Bioinformatics*, 6(1):115, 2005.

[34] L. Hilakivi-Clarke. Estrogens, BRCA1, and breast cancer. *Cancer Res.*, 60(18):4993–5001, September 2000.

[35] G. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[36] G. E. Hinton and S. Roweis. Stochastic Neighbor Embedding. *Neural Information Proceeding Systems:Natural and Synthetic*, 15:833–840, December 2002.

[37] E. Huang, S. H. Cheng, H. Dressman, J. Pittman, M. H. Tsou, C. F. Horng, A. Bild, E. S. Iversen, M. Liao, C. M. Chen, M. West, J. R. Nevins, and A. T. Huang. Gene expression predictors of breast cancer outcomes. *Lancet*, 361:1590–1596, 2003.

[38] T. Iwata, K. Saito, N. Ueda, S. Stromsten, T. L. Griffiths, and J. B. Tenenbaum. Parametric embedding for class visualization. *Neural Computation*, 19:2536–2556, September 2007.

[39] B. Jordan. *DNA Microarrays: Gene Expression Applications*. Springer, Heidelberg, Germany, 1 edition, 2001.

[40] S. J.W. Sammon, G. Stephanopoulos, and G. Stephanopoulos. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5):401–409, May 1969.

[41] J. N. Karmeshu. *Entropy measures, Maximum Entropy Principle and Emerging Applications*. Springer, 2003.

[42] R. Kincaid. Vistaclara: an interactive visualization for exploratory analysis of DNA microarrays. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 580–587, 2004.

[43] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[44] A. Krogh and J. Vedlesby. Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing system*, 7:231–238, 1995.

[45] N. Lama, F. Ambrogi, L. Antolini, P. Boracchi, and E. Biganzoli. Some issues and perspectives in microarray data analysis in breast cancer: the need for integrated research. In *EWADP 2004: 7-9 July Milan, 1st European Workshop on the Assessment of Diagnostic Performance.*, 2004.

[46] J. Landgrebe, W. Wurst, and G. Welzl. Permutation-validated principal components analysis of microarray data. *Genome Biology*, 3(4):0019.1–0019.11, 2002.

[47] J. A. Lee and M. Verleysen. Nonlinear projection with the Isotop method. In *Proceedings of the International Conference on Artificial Neural Networks*, volume 2415, pages 933 – 938, London, UK, 2002.

[48] M. T. Lee. *Analysis of Microarray Gene Expression Data*. Kluwer Academic Publishers, 2004.

[49] Y. F. Leung and D. Cavalieri. Fundamentals of cDNA microarray data analysis. *Trends in Genetics*, 19(11):649–659, 2003.

[50] D. Lowe. Radial basis function networks. *The Handbook of Brain Theory and Neural Networks*, pages 779–782, 1995.

[51] D. Lowe and M. E. Tipping. Neuroscale: novel topographic feature extraction using RBF networks. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 543–549, London, UK, 1997.

[52] D. Lowe and K. Zapart. Point-wise confidence interval estimation by neural networks: A comparative study based on automotive engine calibration. *Neural Computing and Applications*, 8(1):77–85, 1999.

[53] D. Mackay. Bayesian interpolation. *Neural Computation*, 4:415–417, 1992.

[54] D. Mackay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992.

[55] D. M. Maniyar and I. T. Nabney. Data visualization with simultaneous feature selection. In *IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, 2006. CIBCB '06.*, pages 1–8, Toronto, Ont.,, September 2006.

[56] S. Michiels, S. Koscielny, and C. Hill. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet*, 365:488 – 492, 2005.

[57] J. Misra, W. Schmitt, D. Hwang, L. Hsiao, S. Gullans, G. Stephanopoulos, and G. Stephanopoulos. Interactive exploration of microarray gene expression patterns in a reduced dimensional space. *Genome Research*, 12:1112–1120, 2002.

[58] M. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533, 1993.

[59] A. Myllari, T. Salakoski, and A. Pasechnik. On the visualization of the DNA sequence and its nucleotide content. *Sigsam Bulletin*, 39(4):131–135, 2005.

[60] I. T. Nabney. *Netlab: Algorithms for Pattern Recognition.* Advances in Pattern Recognition. Springer-Verlag, London, 2002.

[61] National Cancer Institute. Genetic testing for BRCA1 and BRCA2: It's your choice. Technical report, 2002.

[62] D. A. Nix and A. S. Weigend. Learning local error bars for nonlinear regression. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 489–496. The MIT Press, 1995.

[63] M. Perrone and L. Cooper. When networks disagree: ensemble methods for hybrid neural networks. *Artificial Neural Networks for Speech and Vision.*, 4:126–142, 1993.

[64] S. B. Primrose and R. M. Twyman. *Genomics: Applications in Human Biology.* Blackwell, Oxford, UK, 1 edition, 2004.

[65] J. Quiñonero-Candela, I. Dagan, B. Magnini, and F. D'Alché-Buc, editors. *Evaluating Predictive Uncertainty Challenge*, volume 3944 of *Lecture Notes in Computer Science*, Heidelberg, Germany, 2006. Springer.

[66] S. Ramaswamy, K. N. Ross, E. S. Lander, and T. R. Golub. A molecular signature of metastasis in primary solid tumors. *Nature Genetics*, 33:49 – 54, 2002.

[67] M. Rattray, X. Liu, G. Sanguinetti, M. Milo, and N. D. Lawrence. Propagating uncertainty in microarray data analysis. *Briefings in Bioinformatics*, 7(1):37–47, 2006.

[68] A. Renyi. On measures of entropy and information. *Selected papers of Alfred Renyi*, 2:565–580, 1976.

[69] S. Richard, M. D. Radmacher, K. Dobbin, , and L. M. McShane. Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *Journal of the National Cancer Institute*, 95:14–18, 2003.

[70] C. Ritz. Comparing prognostic markers for metastases in breast cancer using artificial neural networks. MSc. thesis, Department of Theoretical Physics, Lund University, March 2003.

[71] S. Roweis and L. Saul. Nonlinear dimensionality reduction by Locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[72] L. K. Saul and S. T. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[73] S. Schiffman, M. Reynolds, and F. Young. *Introduction to multidimensional scaling.* Academic Press, London, 1981.

[74] C. Shannon. A mathematical theory of communication. *Bell system Tech. Journal*, 27:1803–1806, 1948.

[75] M. Sivaraksa and D. Lowe. Gene expression predictors of breast outcome: A visualisation study. In P. Ruusuvuori, T. Manninen, H. Huttunen, M.-L. Linne, and O. Yli-Karja, editors, *Fourth TICSP International Workshop on Computational Systems Biology, WCSB 2006*, pages 53–56, Tampere, Finland, June 2006.

[76] M. Sivaraksa and D. Lowe. Microarray visualisation using neuroscale models. In *Poster session presented at: European Summer School in Biomedical Informatics*, Balatonfured, Hungary, 2006.

[77] M. Sivaraksa and D. Lowe. Unclassifiability in medical prognosis: example using biopattern gene markers. In *Third International Conference on Computational Intelligence in Medicine and Healthcare*, Plymouth, UK, 2007.

[78] M. Sivaraksa and D. Lowe. Predictive gene lists for breast cancer prognosis: A topographic visualisation study. *BMC Medical Genomics*, 1(1):8, 2008.

[79] Society for Women's Health Research. Women's fear of heart disease has almost doubled in three years, but breast cancer remains most feared disease. *http://www.nhlbi.nih.gov/health/hearttruth/press/fear_doubled.htm*, 2003.

[80] T. Sorlie, C. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. Eisen, M. van de Rijn, and S. Jefrey. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceeding National Academic of Science*, 98:10896 – 10874, 2001.

[81] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. Golub. Interpreting patterns of gene expression with self-organizing maps. methods and application to hematopoietic differentation. *Proc. Natl. Acad. Sci.*, 96:2907–2912, 1997.

[82] Y. Teh and S. Roweis. Automatic alignment of hidden representations. In *Advances in Neural Information Processing System*, volume 15, pages 841 – 848, Cambridge, MA, USA, 2002.

[83] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, December 2000.

[84] M. E. Tipping. *Topographic mappings and feed-forward neural networks.* PhD thesis, Aston University, 1996.

[85] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[86] M. E. Tipping and D. Lowe. Shadow targets: a novel algorithm for topographic projections by radial basis functions. *NeuroComputing*, 19(1):211–222, 1998.

[87] V. Trevino, F. Falciani, and H. Barrena-Saldaña. DNA microarray: a powerful genomic tool for biomedical and clinical research. *Molecular Medicine*, 13(9-10):527 – 541, 2007.

[88] J. Tyrer, S. W. Duffy, and J. Cuzick. A breast cancer prediction model incorporating familial and personal risk factors. *Statistics in Medicine*, 23(7):1111–1130, July 2006.

[89] M. J. van de Vijver, D. Yudong, L. J. van't Veer, H. Dai, A. A. M. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, M. J. Marton, M. Parrish, D. Atsma, A. T. Witteveen, A. Aglas, L. D. Elahay, T. van deer Veld, H. Bartelink, S. Rodenhuis, E. T. RutgersS, S. H. Friend, and R. Bernards. A gene-expression signature as a predictor of survival in breast cancer. *The New England Journal of Medicine*, 347:1999–2009, December 2002.

[90] L. J. van't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. M. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, January 2002.

[91] M. Vazirgiannis, M. Halkidi, and D. Gunopulos. *Uncertainty Handling and Quality Assessment in Data Mining.* Springer, 2003.

[92] J. Wang, J. Delabie, H. C. Aasheim, E. Smeland, and O. Myklebost. Clustering of the SOM easily reveals distinct gene expression patterns: Results of a reanalysis of lymphoma study. *BMC Bioinformatics*, 3(36):1471–2105, 2002.

[93] Y. Wang, J. Klijn, Y. Zhang, A. Sieuwerts, M. Look, F. Yang, D. Talantov, M. Timmermans, M. M. van Gelder, and J. Yu. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *The Lancet*, 365:671 – 679, 2005.

[94] M. R. Warrier and G. K. Khurana-Hershey. Genetics of asthma - personalizing healthcare. *US Respiratory Disease*, 1:247–249, July 2006.

[95] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. Olson, J. R. Marks, and J. R. Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 98(20):11462–11467, 2001.

[96] H. Willard, M. Angrist, and G. S. Ginsburg. Genomic medicine: genetic variation and its impact on the future of health care. *Philosophical Transactions of the Royal Society B*, 360:1543 – 1550, July 2005.

[97] C. Williams. Computation with infinite neural networks. Technical report, Neural Computing Research Group, Aston University, 1997.

[98] C. Williams. *Learning in Graphical Models*, chapter Prediction with Gaussian processes from linear regression to linear prediction and beyond, pages 599–621. MIT Press, 1999.

[99] E. Witt and J. McClure. *Statistics for Microarrays: Design, Analysis and Inference*. Wiley, 2004.

[100] World Health Organization International Agency for Research on Cancer. The future of cancer treatment. *World Cancer Report*, 2003.

[101] K. Y. Yeung and L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

Appendix A

The Two Gene Lists

This appendix shows the predictive gene list of both the original gene list, List A and the alternative gene list, List B. Both gene lists have 6 genes in common which are highlighted in bold.

A.1 A List of the original 70 gene list, List A.

AA555029.RC	AB037863	AF052162	AF055033	AF073519
AF201951	AF257175	AK000745	AL080059	AL080079
AL137718	Contig20217.RC	Contig2399.RC	Contig24252.RC	Contig25991
Contig28552.RC	Contig32125.RC	Contig32185.RC	Contig35251.RC	Contig38288.RC
Contig40831.RC	Contig46218.RC	**Contig46223.RC**	Contig48328.RC	Contig51464.RC
Contig55377.RC	Contig55725.RC	**Contig56457.RC**	Contig63102.RC	Contig63649.RC
NM.000127	NM.000436	NM.000599	NM.000788	**NM.000849**
NM.001282	NM.001809	NM.002019	NM.002073	NM.002916
NM.003239	NM.003607	NM.003748	NM.003862	NM.003875
NM.003882	NM.003981	NM.004702	NM.004994	NM.005915
NM.006101	NM.006117	NM.006681	NM.006931	NM.007036
NM.007203	**NM.014321**	NM.014791	NM.014889	NM.015984
NM.016359	NM.016448	NM.016577	NM.018354	NM.018401
NM.020188	**NM.020386**	NM.020974	U82987	X05610

A.2 A List of the alternative 70 gene list, List B.

AA553619.RC	AB023216	AB032954	AF065241	AL050065
Contig11065.RC	Contig14706.RC	Contig14882.RC	Contig15031.RC	Contig31839.RC
Contig34302.RC	Contig35229.RC	Contig37063.RC	Contig37262	Contig39090.RC
Contig42162.RC	Contig46.RC	**Contig46223.RC**	Contig49818.RC	Contig51800
Contig55189.RC	**Contig55377.RC**	**Contig56457.RC**	Contig753.RC	Contig760.RC
Contig8930.RC	Contig8950.RC	NM.000272	NM.000286	NM.000320
NM.000419	NM.000540	**NM.000849**	NM.001879	NM.002624
NM.003686	NM.003778	NM.003858	NM.004087	NM.004273
NM.004336	NM.004456	NM.004701	NM.004791	NM.005008
NM.005087	NM.005744	NM.006260	NM.006547	NM.007359
NM.012177	NM.012261	NM.012310	NM.012406	NM.014093
NM.014264	**NM.014321**	NM.014404	NM.014547	NM.014675
NM.014968	NM.015434	NM.017926	NM.018089	NM.018098
NM.018313	NM.018488	NM.020123	**NM.020386**	NM.021033

Appendix B

Classification Results of the standard models

From chapter 3, where the standard visualisations were applied on the van't Veer data set. The results of the classification superimposed on the visualisation results are shown in this appendix.

B.1 The misclassification matrix from the NeuroScale projection using List A

The misclassification matrix from the NeuroScale projection using List A as shown in Figure 3.1(a). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 83.33%.

	Predict Poor	Predict Good
Actual Poor	27	7
Actual Good	6	38

B.2 The misclassification matrix from the NeuroScale projection using List A with only high confidence patients

The misclassification matrix from the NeuroScale projection using List A with only high confidence patients as shown in Figure 3.1(a). The classification is performed using only 30 high confidence patients whose indicators are either above 0.7 or below 0.3. The classification rate is 100%.

	Predict Poor	Predict Good
Actual Poor	10	0
Actual Good	0	20

B.3 The misclassification matrix from the NeuroScale projection using List B

The misclassification matrix from the NeuroScale projection using List B as shown in Figure 3.1(b). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 83.33%.

	Predict Poor	Predict Good
Actual Poor	28	6
Actual Good	7	37

B.4 The misclassification matrix from the NeuroScale projection using List B with only high confidence patients

The misclassification matrix from the NeuroScale projection using List B with only high confidence patients as shown in Figure 3.1(b). The classification is performed using only 26 high confidence patients whose indicators are either above 0.7 or below 0.3. The classification rate is 100%.

	Predict Poor	Predict Good
Actual Poor	7	0
Actual Good	0	19

B.5 The misclassification matrix from the LLE projection using List A with $K = 5$

The misclassification matrix from the LLE projection using List A with $K = 5$ as in 3.2(a). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 79.49%.

	Predict Poor	Predict Good
Actual Poor	27	7
Actual Good	9	35

B.6 The misclassification matrix from the LLE projection using List A with $K = 5$ using only high confidence patients

The misclassification matrix from the LLE projection using List A with $K = 5$ using only high confidence patients. as shown in Figure 3.2(a). The classification is performed using only 22 high confidence patients whose indicators are either above 0.7 or below 0.3. The classification rate is 100%.

	Predict Poor	Predict Good
Actual Poor	7	0
Actual Good	0	15

B.7 The misclassification matrix from the LLE projection using List B with $K = 5$

The misclassification matrix from the LLE projection using List B with $K = 5$ as shown in Figure 3.2(b). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 87.18%.

	Predict Poor	Predict Good
Actual Poor	31	3
Actual Good	7	37

B.8 The misclassification matrix from the LLE projection using List B with $K = 5$ using only high confidence patients

The misclassification matrix from the LLE projection using List B with $K = 5$ using only high confidence patients as shown in Figure 3.2(b). The classification is performed using 29 high confidence patients whose indicators are either above 0.7 or below 0.3. Again a perfect classification rate is achieve.

	Predict Poor	Predict Good
Actual Poor	8	0
Actual Good	0	21

B.9 The misclassification matrix from the LLE projection using List A with $K = 20$

The misclassification matrix from the LLE projection using List A with $K = 20$ as shown in Figure 3.3(a). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 93.58%.

	Predict Poor	Predict Good
Actual Poor	33	1
Actual Good	4	40

B.10 The misclassification matrix from the LLE projection using List A with $K = 20$ using only high confidence patients

The misclassification matrix from the LLE projection using List A with $K = 20$ using only high confidence patients as shown in Figure 3.3(a). The classification is performed using 41 high confidence patients whose indicators are either above 0.7 or below 0.3. The classification rate is 100%.

	Predict Poor	Predict Good
Actual Poor	12	0
Actual Good	0	29

B.11 The misclassification matrix from the LLE projection using List B with $K = 20$

The misclassification matrix from the LLE projection using List B with $K = 20$ as shown in Figure 3.3(b). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 84.62%.

	Predict Poor	Predict Good
Actual Poor	27	7
Actual Good	5	39

B.12 The misclassification matrix from the LLE projection using List B with $K = 20$ using only high confidence patients

The misclassification matrix from the LLE projection using List B with $K = 20$ using only high confidence patients as shown in Figure 3.3(b). The classification is performed using 28 high confidence patients whose indicators are either above 0.7 or below 0.3. The classification rate is 100%.

	Predict Poor	Predict Good
Actual Poor	12	0
Actual Good	0	16

B.13 The misclassification matrix from the GTM projection using List A

The misclassification matrix from the GTM projection using List A as shown in Figure 3.4(a). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 80.77%.

	Predict Poor	Predict Good
Actual Poor	30	4
Actual Good	10	31

B.14 The misclassification matrix from the GTM projection using List A using only high confidence patients

The misclassification matrix from the GTM projection using List A using only high confidence patients as shown in Figure 3.4(a). The classification is performed using 26 high confidence patients whose indicators are either above 0.7 or below 0.3. The classification rate is 96.15%.

	Predict Poor	Predict Good
Actual Poor	7	1
Actual Good	0	18

B.15 The misclassification matrix from the GTM projection using List B

The misclassification matrix from the GTM projection using List B as shown in Figure 3.4(b). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 79.49%.

	Predict Poor	Predict Good
Actual Poor	28	6
Actual Good	10	34

B.16 The misclassification matrix from the GTM projection using List B using only high confidence patients

The misclassification matrix from the GTM projection using List B using only high confidence patients as shown in Figure 3.4(b). The classification is performed using 21 high confidence patients whose indicators are either above 0.7 or below 0.3. The classification rate is 100%.

	Predict Poor	Predict Good
Actual Poor	2	0
Actual Good	0	19

B.17 The misclassification matrix from the SNE projection using List A with $\sigma = \log(5)$

The misclassification matrix from the SNE projection using List A with $\sigma = \log(5)$ as shown in Figure 3.6(a). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 79.49%.

	Predict Poor	Predict Good
Actual Poor	27	7
Actual Good	9	35

B.18 The misclassification matrix from the SNE projection using List A with $\sigma = \log(5)$ using only high confidence patients

The misclassification matrix from the SNE projection using List A with $\sigma = \log(5)$ using only high confidence patients as shown in Figure 3.6(a). The classification is performed using 16 high confidence patients whose indicators are either above 0.7 or below 0.3. The classification rate is 100%.

	Predict Poor	Predict Good
Actual Poor	4	0
Actual Good	0	12

B.19 The misclassification matrix from the SNE projection using List B with $\sigma = \log(5)$

The misclassification matrix from the SNE projection using List B with $\sigma = \log(5)$ as shown in Figure 3.6(b). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 80.77%.

	Predict Poor	Predict Good
Actual Poor	25	9
Actual Good	6	38

B.20 The misclassification matrix from the SNE projection using List B with $\sigma = \log(5)$ using only high confidence patients

The misclassification matrix from the SNE projection using List B with $\sigma = \log(5)$ using only high confidence patients as shown in Figure 3.6(b). The classification is performed using 17 high confidence patients whose indicators are either above 0.7 or below 0.3. The classification rate is 100%.

	Predict Poor	Predict Good
Actual Poor	8	0
Actual Good	0	9

B.21 The misclassification matrix from the SNE projection using List A with $\sigma = \log(20)$

The misclassification matrix from the SNE projection using List A with $\sigma = \log(20)$ using only high confidence patients as shown in Figure 3.7(a). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 79.49%.

	Predict Poor	Predict Good
Actual Poor	24	10
Actual Good	6	38

B.22 The misclassification matrix from the SNE projection using List A with $\sigma = \log(20)$ using only high confidence patients

The misclassification matrix from the SNE projection using List A with $\sigma = \log(20)$ using only high confidence patients as shown in Figure 3.7(a). The classification is performed using 10 high confidence patients whose indicators are either above 0.7 or below 0.3. The classification rate is 100%.

	Predict Poor	Predict Good
Actual Poor	2	0
Actual Good	0	8

B.23 The misclassification matrix from the SNE projection using List B with $\sigma = \log(20)$

The misclassification matrix from the SNE projection using List B with $\sigma = \log(20)$ using only high confidence patients as shown in Figure 3.7(b). The classification is performed using the original 78 patients with 0.5 prognosis indicator as a threshold boundary. The classification rate is 74.36%.

	Predict Poor	Predict Good
Actual Poor	25	9
Actual Good	11	33

B.24 The misclassification matrix from the SNE projection using List B with $\sigma = \log(20)$ using only high confidence patients

he misclassification matrix from the SNE projection using List B with $\sigma = \log(20)$ using only high confidence patients as shown in Figure 3.7(b). The classification is performed using 14 high confidence patients whose indicators are either above 0.7 or below 0.3. The classification rate is 100%.

	Predict Poor	Predict Good
Actual Poor	2	0
Actual Good	0	12

B.25 The misclassification matrix from the NeuroScale projection on the new 234 patients using List A

The misclassification matrix from the NeuroScale projection on the new 234 patients using List A as shown in Figure 3.8(a). The classification is performed using the new 234 patients with 0.5 prognosis indicator as a threshold boundary 70 gene set provided by List A. The overall classification rate is 58.97%

	Predict Poor	Predict Good
Actual Poor	56	19
Actual Good	77	82

B.26 The misclassification matrix from the NeuroScale projection on the new 234 patients using List A with only high confidence patients

The misclassification matrix from the NeuroScale projection on the new 234 patients using List A with only high confidence patients as shown in Figure 3.8(a). The classification is performed using only the 78 high confidence from the new 234 patients whose indicators are either above 0.7 or below 0.3 using 70 gene set provided by List A. The overall classification rate improves to 60.26%

	Predict Poor	Predict Good
Actual Poor	24	3
Actual Good	28	33

B.27 The misclassification matrix from the NeuroScale projection on the new 234 patients using List B

The misclassification matrix from the NeuroScale projection on the new 234 patients using List B as shown in Figure 3.8(b). The classification is performed using the new 234 patients with 0.5 prognosis indicator as a threshold boundary 70 gene set provided by List A. The overall classification rate is 56.40%

	Predict Poor	Predict Good
Actual Poor	32	43
Actual Good	59	100

B.28 The misclassification matrix from the NeuroScale projection on the new 234 patients using List B with only high confidence patients

The misclassification matrix from the NeuroScale projection on the new 234 patients using List B with only high confidence patients as shown in Figure 3.8(b). The classification is performed using only the 92 high confidence from the new 234 patients whose indicators are either above 0.7 or below 0.3 using 70 gene set provided by List A. The overall classification rate improves to 67.40%

	Predict Poor	Predict Good
Actual Poor	13	21
Actual Good	9	49

B.29 The misclassification matrix from the GTM projection on the new 234 patients using List A

The misclassification matrix from the GTM projection on the new 234 patients using List A as shown in Figure 3.9(a). The classification is performed using the new 234 patients with 0.5 prognosis indicator as a threshold boundary 70 gene set provided by List A. The overall classification rate is 59.40%

	Predict Poor	Predict Good
Actual Poor	60	15
Actual Good	80	79

B.30 The misclassification matrix from the GTM projection on the new 234 patients using List A with only high confidence patients

The misclassification matrix from the GTM projection on the new 234 patients using List A with only high confidence patients as shown in Figure 3.9(a). The classification is performed using only the 51 high confidence from the new 234 patients whose indicators are either above 0.7 or below 0.3 using 70 gene set provided by List A. The overall classification rate improves to 60.26%

	Predict Poor	Predict Good
Actual Poor	8	1
Actual Good	11	31

B.31 The misclassification matrix from the GTM projection on the new 234 patients using List B

The misclassification matrix from the GTM projection on the new 234 patients using List B as shown in Figure 3.9(b). The classification is performed using the new 234 patients with 0.5 prognosis indicator as a threshold boundary 70 gene set provided by List A. The overall classification rate is 47.86%

	Predict Poor	Predict Good
Actual Poor	64	11
Actual Good	111	48

B.32 The misclassification matrix from the GTM projection on the new 234 patients using List B with only high confidence patients

The misclassification matrix from the GTM projection on the new 234 patients using List B with only high confidence patients as shown in Figure 3.9(b). The classification is performed using only the 13 high confidence from the new 234 patients whose indicators are either above 0.7 or below 0.3 using 70 gene set provided by List B. The overall classification rate improves to 84.62%

	Predict Poor	Predict Good
Actual Poor	0	2
Actual Good	0	11

Appendix C

Traditional Visualisation Approaches

C.1 Principal Component Analysis

Principle Component Analysis (PCA) is the most common method for dimensionality reduction.

Suppose that we are trying to map $\mathbf{x}$ in D-dimensional space to a subspace d. For visualisation, d is usually 2 so that the vector $\mathbf{x}$ can be represented as a linear combination of a set of d orthonormal vectors u_i onto vector $\mathbf{z}$

$$\boldsymbol{x} = \sum_{i=1}^{D} z_i \mathbf{u}_i \tag{C.1}$$

$\boldsymbol{x}$ can then be represented as

$$\boldsymbol{x} = \sum_{i=1}^{d} z_i \boldsymbol{u}_i + \sum_{i=d+1}^{D} z_i \boldsymbol{u}_i. \tag{C.2}$$

Suppose that we retain only a subset d of the basis vectors u_i, so that we use only the first d coefficients z_i to approximate $\boldsymbol{x}$. The remaining coefficients will be replaced by constant b_i. Vector $\boldsymbol{x}$ is approximated by:

$$\tilde{\boldsymbol{x}} = \sum_{i=1}^{d} x_i \boldsymbol{u}_i + \sum_{i=d+1}^{D} b_i \boldsymbol{u}_i. \tag{C.3}$$

The error of the projection is given by

$$\boldsymbol{x}^n - \tilde{\boldsymbol{x}}^n = \sum_{i=d+1}^{D} (z_i^n - b_i) \boldsymbol{u}_i. \tag{C.4}$$

The best approximation is defined to be that which minimises the sum of the square

errors over the whole data set. Thus we minimise

$$E = \frac{1}{2}\sum_{n=1}^{N} \|\boldsymbol{x}^n - \tilde{\boldsymbol{x}}^n\|^2$$
$$= \frac{1}{2}\sum_{n=1}^{N}\sum_{i=d+1}^{D}\sum_{j=d+1}^{D}(x_i^n - b_i)(x_j^n - b_j)\mathbf{u}_i^T\mathbf{u}_j \tag{C.5}$$

since u_i is orthornormal, it satisfies

$$\boldsymbol{u}_i^T\boldsymbol{u}_j = \delta_{ij}, \tag{C.6}$$

where δ is the Kronecker delta.

Hence (C.5) can be rewritten as:

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{i=M+1}^{d}(x_i^n - b_i)^2. \tag{C.7}$$

If we set the derivative of E with respect to b_i to zero we find

$$b_i = \frac{1}{N}\sum_{i=1}^{N} x_i^N, \tag{C.8}$$

which is the mean vector $\overline{\mathbf{x}}$ with respect to the coordinate $u_1, \ldots, u_d$. This is equal to $u_i^T\overline{\mathbf{x}}$. The error term can be written as

$$E = \frac{1}{2}\sum_{i=M+1}^{d}\sum_{n=1}^{N}\{\boldsymbol{u}_i^T(\boldsymbol{x}_n - \overline{\boldsymbol{x}})^T\}\{(\boldsymbol{x}_n - \overline{\boldsymbol{x}})\boldsymbol{u}_i\}$$
$$= \frac{1}{2}\sum_{i=M+1}^{d}\boldsymbol{u}_i^T\boldsymbol{\Sigma}\boldsymbol{u}_i \tag{C.9}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the set of vectors$\{x_n\}$ and is given by

$$\boldsymbol{\Sigma} = \sum_{n=1}^{N}(x_n - \overline{x})^T(x_n - \overline{x}). \tag{C.10}$$

Minimise E with respect to the choice of basis vectors u_i, occur at the eigenvectors of $\boldsymbol{\Sigma}$, so that $\boldsymbol{\Sigma}\boldsymbol{u}_i = \lambda_i\boldsymbol{u}_i$. Substitute into (C.9), we get

$$E = \frac{1}{2}\sum_{i=d+1}^{D}\lambda_i. \tag{C.11}$$

Thus, the minimum error is obtained by choosing the $d - M$ smallest eigenvalues. This means that the best signal is obtained by projection onto the d eigenvectors with the largest eigenvalues. These set of eigenvectors corresponding to the ranked set of d largest eigenvalues are called the first M principal components.

C.2 Multidimensional Scaling

Multidimensional Scaling [14, 73] is a mathematical tool for representing the similarities of the object as in a map. It also can be a useful tool for doing a dimensionality reduction of the data. Metric MDS uses the direct similarity (or distance) measure as an input, without requiring the original position of each object. For dimensionality purposes, MDS usually uses the Euclidean distance as a measurement of the dissimilarity of objects.

The classical example of multidimensional scaling is to use distances between a group of cities and uses MDS to reconstruct a location map of the cities.

Classical multidimensional scaling which is commonly used to reconstruct the coordinate space is described below.

C.2.1 Classical multidimensional scaling

This classical multidimensional scaling treats the dissimilarities precisely as Euclidean distances.

Suppose there are n points in a p-dimensional space with coordinates $\mathbf{x}_i (i = 1, \ldots, n)$ where $\mathbf{x}_i = (x_1, \ldots, x_p)^T$. Then the Euclidean distance between two points, i and j is given by,

$$\begin{aligned} d_{ij}^2 &= (\mathbf{x}_i - \mathbf{x}_j')^T(\mathbf{x}_i - \mathbf{x}_j) \\ &= \mathbf{x}_i^T\mathbf{x}_i + \mathbf{x}_j^T\mathbf{x}_j - 2\mathbf{x}_i^T\mathbf{x}_j. \end{aligned} \tag{C.12}$$

Suppose $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$ and place the centre of $\mathbf{X}$ at the origin. Hence

$$\sum_{i=1}^{n} \mathbf{x}_i = 0.$$

Summing (C.12) gives

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} d_{ij}^2 &= \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i^T\mathbf{x}_i + \mathbf{x}_j^T\mathbf{x}_j, \\ \frac{1}{n}\sum_{j=1}^{n} d_{ij}^2 &= \mathbf{x}_i^T\mathbf{x}_i + \frac{1}{n}\sum_{j=1}^{n} \mathbf{x}_j^T\mathbf{x}_j, \\ \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2 &= \frac{2}{n}\mathbf{x}_i^T\mathbf{x}_i. \end{aligned} \tag{C.13}$$

Then, let $\mathbf{B}$ be the inner product where,

$$[\mathbf{B}]_{ij} = b_{ij} = \mathbf{x}_j^T\mathbf{x}_j. \tag{C.14}$$

Substitute (C.13) into (C.12) gives

$$\begin{aligned} b_{ij} &= x_i^T x_j \\ &= -\frac{1}{2}\left(d_{ij}^2 - \frac{1}{n}\sum_{i=1}^{n} d_{ij}^2 - \frac{1}{n}\sum_{j=1}^{n} d_{ij}^2 + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2\right). \end{aligned} \tag{C.15}$$

Define $\mathbf{A}$ as $[\mathbf{A}]_{ij} = -\frac{1}{2}d_{ij}^2$, then

$$\mathbf{B} = \mathbf{HAH}, \tag{C.16}$$

where $\mathbf{H}$ is the *centring* matrix.

$$\mathbf{H} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^T, \tag{C.17}$$

with $\mathbf{1} = (1, 1, \ldots, 1)^T$, a vector of n ones. After finding $\mathbf{B}$, use $\mathbf{B}$ to recover the coordinate

$$\mathbf{B} = \mathbf{X}\mathbf{X}^T. \tag{C.18}$$

Since $\mathbf{X}$ is an $n \times p$ matrix of coordinates, the rank of $\mathbf{B}$, $r(\mathbf{B})$, is then

$$r(\mathbf{B}) = r(\mathbf{X}\mathbf{X}^T) = r(\mathbf{X}) = p.$$

Now $\mathbf{B}$ is symmetric, positive semi-definite and of rank p and hence has p non-negative eigenvalues and $n - p$ zero eigenvalues. $\mathbf{B}$ can now be rewritten as

$$\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T, \tag{C.19}$$

where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ with $\lambda_1 > \lambda_2 > \cdots > \lambda_n$ and $\mathbf{V} = [v_1, \ldots, v_n]$ From $B = \mathbf{X}\mathbf{X}^T$, the coordinate $\mathbf{X}$ can be calculated by

$$\mathbf{X} = \mathbf{V}_1\mathbf{\Lambda}_1^{\frac{1}{2}}, \tag{C.20}$$

where $\mathbf{\Lambda}_1^{\frac{1}{2}} = \text{diag}(\lambda_1^{\frac{1}{2}}, \ldots, \lambda_p^{\frac{1}{2}})$. The coordinates of the points have projection been recovered from the distances between the points.

The dimensionality used to represent the dissimilarities has been indicated previously in λ_i. If $\mathbf{B}$ is positive semi-definite then the number of non-zero eigenvalues gives the number of dimensions required. If $\mathbf{B}$ is not positive semi-definite then the number of positive eigenvalues is the appropriate number of dimensions. However, these give the maximum numbers of dimensions. In practice, to make the dimensionality small simply choose the first p eigenvalues and eigenvectors of $\mathbf{B}$ to get a small dimensional space (usually p=2 or 3).

In fact Classical MDS or Metric MDS is closely related to Principal Component Analysis(PCA) [14].

C.2.2 Non-metric multidimensional scaling

Non-metric multidimensional scaling does not necessarily preserve the distances within the solutions; however it preserves the rank-order of the original distances [14]. The non-metric model only requires the new distances in the projection space to be monotonic with respect to the original ones. The introduction of a nonmetric model is because in some application the magnitude of the data is unreliable [14]. The ranking order., however, is more reliable. The dissimilarity has the following form:

$$\delta_{ij} = f(d_{ij}), \tag{C.21}$$

where f is a monotone function such that

$$d_{ij}d_{i'j'} \Rightarrow f(d_{ij}) < f(d_{i'j'})$$

for all i, i', j and j'.

This method will reconstruct the projected images so that the distance estimates preserves the ranking of dissimilarity δ_{ij}. The distance of the projected images is normally measured by the *Minkowski distance function*:

$$d_{ij} = (\sum \|x_{ik} - x_{jk}\|^p)^{\frac{1}{p}}. \tag{C.22}$$

Any p can be chosen to best fit the data, however the most common p is $p = 2$ which is equivalent to the Euclidean distance.

The STRESS measured introduce by Kruskal has two forms: STRESS formula one (S1):

$$S_1 = \left[\frac{\sum_{(i,j)}(\hat{\delta_{ij}} - \hat{d_{ij}})^2}{\sum_{(i,j)} \hat{d}_{ij}^2}\right]^{\frac{1}{2}}. \tag{C.23}$$

and STRESS formula two (S2):

$$S_2 = \left[\frac{\sum_(i,j)(\hat{\delta_{ij}} - \hat{d_{ij}})^2}{\sum_{(i,j)} \hat{d_{ij}} - \hat{d}_{..})^2}\right]^{\frac{1}{2}}, \tag{C.24}$$

where

$$\hat{d}_{..} = \frac{1}{IJ}\sum_{(i,j)} \hat{d_{ij}}. \tag{C.25}$$

Two formulae are similar with the only difference being on the normalising denominator.

C.2.3 MDS Example

An example of MDS is mapping relative locations of cities from their distances, as shown in table C.1. The classical MDS algorithm uses the distances to compute the A and B matrices in order to gain the relative positions. Figure C.1 shows the positions gained from the MDS algorithm. Finally, to verify the result Table C.2 shows the distances between the cities calculating from using the distances between the location of points in figure C.1. The result shows similar distances to the original distances with small errors.

	Brighton	London	Birmingham	Nottingham	Newcastle	Edinburgh
Brighton	0	51.5	167.6	189	325	463
London	51.5	0	120	130	280	403.5
Birmingham	167.6	120	0	50	198	284
Nottingham	189	130	50	0	154	253
Newcastle	325	280	198	154	0	120.8
Edinburgh	463	403.5	284	253	120.8	0

Table C.1: A table of distances between various cities with units in miles.

Classical MDS

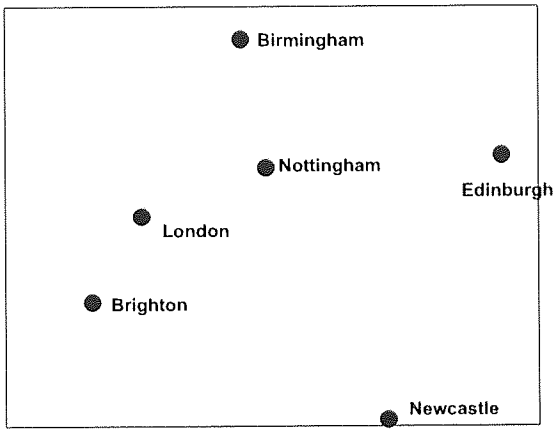


Figure C.1: A possible reconstructed map of British cities reconstructed from the relative journey between the cities. This result preserves the distance between the cities although it is rotated and reflected from the original map.

	Brighton	London	Birmingham	Nottingham	Newcastle	Edinburgh
Brighton	0	60.89	183.6	199.5	334.5	463.03
London	60.89	0	122.77	140.76	283.11	405.77
Birmingham	183.6	122.77	0	46.14	197.72	295.33
Nottingham	199.5	140.76	46.14	0	154.83	265.35
Newcastle	334.5	283.11	197.72	154.83	0	149.01
Edinburgh	463.03	405.77	295.33	265.35	149.01	0

Table C.2: A table of reconstructed distances using Classical MDS between various cities with units in miles.

Non Metric MDS

The result in Figure C.2 shows relative positions of the cities calculated using relative positions between the location of points with non-metric MDS.

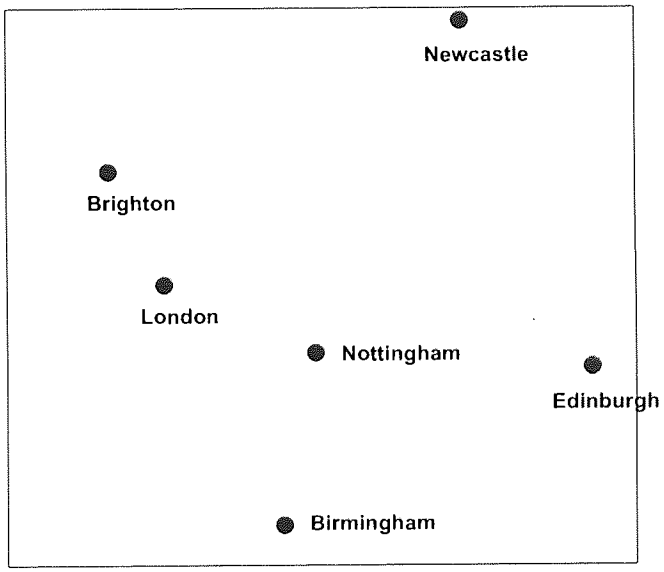


Figure C.2: A possible reconstructed map of British cities reconstructed from the relative journey distances between the cities. This result preserves the ranking of the distances between the cities although it is rotated and reflected from the original map.

	Brighton	London	Birmingham	Nottingham	Newcastle	Edinburgh
Brighton	0	0.0341	0.3370	0.4054	1.1434	2.1437
London	0.0341	0	0.1566	0.2108	0.8361	1.6678
Birmingham	0.3370	0.1566	0	0.0230	0.4054	0.8761
Nottingham	0.4054	0.2108	0.0230	0	0.2446	0.6947
Newcastle	1.1434	0.8361	0.4054	0.2446	0	0.2108
Edinburgh	2.1437	1.6678	0.8761	0.6947	0.2108	0

Table C.3: A table of reconstructed distances using non-matric MDS between various cities.

Appendix D

Standard Shadow Targets

The standard shadow targets algorithm was originally used in the standard NeuroScale. The shadow target algorithm is a modification of the Radial Basis Function(RBF) network which was originally used for regression problems in order to be able to corporate in the visualisation field. The RBF network is defined as

$$\mathbf{Y} = \mathbf{\Phi W}, \tag{D.1}$$

where $\mathbf{W}$ is the network weights, $\mathbf{\Phi}$ is the design matrix which contains basis function networks and $\mathbf{Y}$ is the output vectors of the network. In the original RBF training algorithm, the target values will be provided therefore solving the weight vectors in the RBF is staightforward using supervised training algorithm.

However, in the topographic visualisation the target vectors is not explicitly present. The training algorithm is called *relative supervision* by using pairs of data points to control the target in order to perserve the relative distances between those pairs.

The topographic mapping minimising the STRESS measure between data and projection spaces of N data samples:

$$E = \sum_{i}^{N} \sum_{j}^{N} (d_{ij}^{*} - d_{ij})^2, \tag{D.2}$$

where d_{ij} and d_{ij}^{*} are the measures of the projection space and the original space which are measured by Euclidean distances. This function minimises the inter-point distances between data points in the two spaces. The simplest approach to training the model is simply to compute the partial derivatives of E (from (D.2)) with respect to the network weights (D.1) [86],

$$\frac{\partial E}{\partial w_{kr}} = \sum_{i}^{N} \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial w_{kr}}. \tag{D.3}$$

By differentiating (D.3), it is easy to see that

$$\frac{\partial E}{\partial y_i} = -2 \sum_{i \neq j} \left(\frac{d_{ij}^{*} - d_{ij}}{d_{ij}} \right) (y_i - y_j). \tag{D.4}$$

From equation (D.4), the gradient of y_i can be defined as

$$\frac{\partial y_i}{\partial w_{kr}} = \delta_{ir} z_k, \tag{D.5}$$

where z_k is the output of the kth hidden unit and δ_{ij} is the Kronecker delta. These derivatives are all that is needed to train an RBF network to perform a topographic projection.

In a supervised problem, the error E is given by

$$E = \frac{1}{2}\sum_i \|y_i - t_i\|^2. \tag{D.6}$$

This equation (D.6) may be combined with $\partial E/\partial y_i$, given by (D.4) in the relative supervision algorithm to get *estimated targets* $\hat{t}_i$.

$$\hat{t}_i = y_i - \frac{\partial E}{\partial y_i}, \tag{D.7}$$

$$= y_i + 2\sum_{j\neq i}\left(\frac{d^*_{ij} - d_{ij}}{d_{ij}}\right)(y_i - y_j). \tag{D.8}$$

The vectors $\hat{t}_i$ represent the exact target values of the RBF network. However, these target values are not fixed but are dependent upon the current outputs of the network, y_i and also the weights. For fixed targets $\hat{t}_i$, the least squares problem can be solved directly:

$$W = \mathbf{\Phi}^\dagger \hat{T}. \tag{D.9}$$

A logical approach is to repetitively estimate the shadow targets at each step. However, in the early stages of training, the targets given by (D.7) may be poor and may lead to an increase in STRESS. A more effective approach is to introduce an additional parameter η, which is initially small, and estimate the target as

$$\hat{t}_i = y_i - \eta\frac{\partial E}{\partial y_i}, \tag{D.10}$$

and increase η as STRESS decreases during the training problem. It is clear that η should be restricted to the range (0,1).

1. Initialise the network weights, $\mathbf{W}$ to small random values.
2. Initialise η to some small positive value.
3. Initialise $\mathbf{\Phi}$ by selecting appropriate network centres and variances.
4. Calculate the pseudo-inverse of the activation function $\mathbf{\Phi}^\dagger$.
5. Use (D.10) to compute estimated targets $\hat{t}_i$.
6. Calculate E from (D.2) and compare with the previous value
   (a) If E has increased, which means that η_y is too large, set $\eta_y = \eta_y \times k_{\text{down}}$, where k_{down} is a small predefined value to reduce the η_y value. Restore previous values of W.
   (b) If E has decreased, which means η may be too small, set $\eta = \eta \times k_{\text{up}}$. Generally $k_{\text{down}} < k_{\text{up}}$.
7. Solve for the weights $\mathbf{W} = \mathbf{\Phi}^\dagger \hat{t}_i$.
8. If (D.2) has not converged, return to Step 5.

In the thesis we modify this algorithm to incorporate uncertainty.