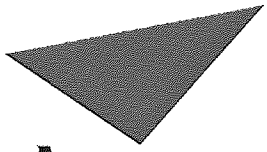
Exploratory data analysis with non-linear and missing data in geochemistry

MARTIN SCHROEDER

Doctor Of Philosophy



Aston University

– ASTON UNIVERSITY –

October 2009

ASTON UNIVERSITY

Exploratory data analysis with non-linear and missing data in geochemistry

MARTIN SCHROEDER

Doctor Of Philosophy, 2009

Thesis Summary

Exploratory analysis of data seeks to find common patterns to gain insights into the structure and distribution of the data. In geochemistry it is a valuable means to gain insights into the complicated processes making up a petroleum system. Typically linear visualisation methods like principal components analysis, linked plots, or brushing are used. These methods can not directly be employed when dealing with missing data and they struggle to capture global non-linear structures in the data, however they can do so locally.

This thesis discusses a complementary approach based on a non-linear probabilistic model. The generative topographic mapping (GTM) enables the visualisation of the effects of very many variables on a single plot, which is able to incorporate more structure than a two dimensional principal components plot. The model can deal with uncertainty, missing data and allows for the exploration of the non-linear structure in the data.

In this thesis a novel approach to initialise the GTM with arbitrary projections is developed. This makes it possible to combine GTM with algorithms like Isomap and fit complex non-linear structure like the Swiss-roll. Another novel extension is the incorporation of prior knowledge about the structure of the covariance matrix. This extension greatly enhances the modelling capabilities of the algorithm resulting in better fit to the data and better imputation capabilities for missing data.

Additionally an extensive benchmark study of the missing data imputation capabilities of GTM is performed. Further a novel approach, based on missing data, will be introduced to benchmark the fit of probabilistic visualisation algorithms on unlabelled data.

Finally the work is complemented by evaluating the algorithms on real-life datasets from geochemical projects.

Keywords: Generative Topographic Mapping, Data Imputation, Data Visualisation, Covariance Matrix, Chemometrics

Contents

1 Introduction **18**
- 1.1 Data Exploration/Visualisation and GTM . . . 20
- 1.2 Petroleum Exploration . . . 22
- 1.3 Motivation . . . 23
- 1.4 Project Aims . . . 23
- 1.5 Report Overview . . . 24
- 1.6 Publications resulting from this thesis . . . 25
- 1.7 Notation . . . 27
- 1.8 Abbreviations . . . 28

2 Geochemistry **30**
- 2.1 Petroleum Generation and Entrapment . . . 31
- 2.2 Petroleum Geochemistry . . . 34
- 2.3 Gas Chromatography-Mass Spectrometry (GC-MS) . . . 36
- 2.4 Chemometrics in Geochemistry . . . 40
- 2.5 Summary . . . 45

3 Toy data sets used in this report **46**
- 3.1 S-shaped data . . . 47
- 3.2 Swiss-roll data . . . 47
- 3.3 Multi phase oil flow data . . . 47
- 3.4 20 and 60 dimensional toy data . . . 48

4 Data Modelling and Exploration **56**
- 4.1 EM Algorithm . . . 58
- 4.2 Mixture Models . . . 59
- 4.3 Gaussian Mixture Models . . . 60
- 4.4 Generative Topographic Mapping . . . 62
  - 4.4.1 Data Visualisation using GTM . . . 65
  - 4.4.2 Initialising GTM . . . 68
- 4.5 Other visualisation algorithms . . . 71
  - 4.5.1 PCA . . . 71
  - 4.5.2 Probabilistic PCA . . . 71
  - 4.5.3 Kernel PCA . . . 71
  - 4.5.4 Gaussian Process Latent Variable Model (GPLVM) . . . 71
  - 4.5.5 MDS . . . 72

4.5.6 Neuroscale 72
4.5.7 Isomap 72
4.6 Summary 72

5 Extensions to GTM **75**
5.1 Block Extension to GTM (B-GTM) 79
5.1.1 Heuristics to stabilise the EM algorithm 80
5.1.2 Variable Block Determination using Optimal Leaf Ordering (OLO) 81
5.1.3 Variable Block Determination using Quick Bayesian Correlation Estimation (QuickBCE) 83
5.2 GTM Visualisation Space Reverse Mapping Initialisation (GTM-VSRMI) 85
5.3 Assessing the novel Extensions 87
5.3.1 B-GTM 89
5.3.2 QuickBCE vs. OLO 99
5.3.3 GTM-VSRMI 101
5.4 Summary 103

6 Missing Data **105**
6.1 Single Imputation 109
6.1.1 Mean Imputation (MI) 109
6.1.2 Weighted Mean Imputation (WMI) 110
6.1.3 Sequential Multiple Regression Imputation (SRI) 110
6.1.4 Multiple Regression Imputation with Mean initialisation and Correlation Cut (MRI) 111
6.1.5 Probabilistic PCA with Missing Data (Bayesian PCA or BPCA) 112
6.1.6 EM for Missing Data in Mixture Models 113
6.1.7 EM for Missing Data in Gaussian Mixture Models 113
6.1.8 Extension of GTM for Missing Data Imputation using EM (GTMI) 115
6.1.9 Extension of B-GTM for Missing Data Imputation using EM (B-GTMI) 117
6.1.10 Performance Indicator 118
6.1.11 General behaviour of the performance indicator and the imputation methods 118
6.1.12 Benchmark 122
6.1.13 Projection Results 122
6.2 Missing data as a way to assess the model fit in unsupervised learning 124
6.3 Conclusion 127

7 Working with real data **129**
7.1 Data pre-treatment 130
7.2 Exploring the non-linear mapping of GTM 131

7.3 Integration of GTM and PCA with pIGI for data exploration 138
7.4 Case Study: Barents Sea Data 138
7.5 Benchmark Study 144
7.5.1 African Data 147
7.5.2 North Sea Data 151
7.6 Performance Study: Speed of the methods 154
7.7 Summary 156

8 Discussion and Future Work **158**
8.1 Discussion 159
8.2 Future Work 161
8.3 Summary 163

9 References **165**

A Additional Graphics **177**
A.1 Chapter 5 178
A.2 Chapter 6 180

B Data Modelling **185**
B.1 PCA 185
B.2 Probabilistic PCA 187
B.3 Kernel PCA 188
B.4 Gaussian Process based data visualisation 189
B.4.1 Gaussian Processes 189
B.4.2 Gaussian Process Latent Variable Model (GPLVM) 191
B.5 MDS 192
B.6 Neuroscale 193
B.7 Isomap 194

List of Figures

2.1 The accumulation of organic material can only happen if it is not decomposed and thus preserved. The material can originate from land plants, aquatic plants or bacteria in the ground. *With permission of IGI Ltd.* 33

2.2 The maturation of source rocks is a complex chemical and physical reaction which is influenced by the depth and temperature of the rock. Depth and temperature are highly related and thus can be plotted on the same axis. Certain depth and temperature windows are associated with the formation of oil, gas and graphite. *With permission of IGI Ltd.* 33

2.3 Oil and gas movement in the subsurface occurs in various stages: (1) A newly generated molecule moves away from a kerogen particle down a pressure and/or concentration gradient into a micropore in the source rock. (2) The molecules accumulate to an oil droplet and move through fracture or intergranular porosity or perhaps by diffusion through the kerogen network. (3) Following carrier beds the oil moves to the surface as an oil seep or gets entrapped in a basin. *With permission of IGI Ltd.* 35

2.4 In geochemistry cross and ternary plots are common to interpret the geochemical compositions of samples and to help with the determination of their origin and possible alteration processes. a) Ternary plot of sterane composition to resolve terrestrial, lacustrine and marine sources. b) Cross plot of sterane ratios to reveal differences in secondary migration. *With permission of IGI Ltd.* 37

2.5 Schematic of the GC-MS: The GC-MS works by injecting the sample into a carrier phase (usually an inert gas) which splits the molecules over time by letting them travel through a column and by heating them up and thus eluting them at different points in time. Attached to the end of the GC is a mass spectrometer which is used to detect the different molecules. *Source: Wikipedia* 38

2.6 Examples of typical GC-MS charts running over the retention time of the analysis. The different spikes mark the detection of chemical molecules which have been picked up with different levels of intensity. *Source: http://geology.gsapubs.org/content/37/10/875/F2.large.jpg* 39

2.7 Measuring partially resolved peaks is done by drawing a baseline between the baseline on the fully-resolved side of the peak and the valley of the partially resolved peak. The distance is then measured from the apex of the peak to the red dotted baseline. *With permission of IGI Ltd.* 41
2.8 Schematic of peak measurements. *With permission of IGI Ltd.* 42
2.9 Baseline signature of a blank sample run: (A) Gradual then rapid increase in the baseline through column bleed. (B-C) Increase of bleed with increasing temperature until maximum temperature programme is reached. (D) The baseline stays stable. (E) Sudden drop of baseline when the oven rapidly cools down at end of the temperature programme. *With permission of IGI Ltd.* 42

3.1 (a) The S-shaped data in 3D. (b) The PCA projection or scores plot for PC1 vs PC2. (c) The PCA projection or scores plot for PC1 vs PC3. (d) The loadings plots shows the contribution of each dimensions (labelled D1, D2, D3) to the first two principal components. A very high or low value on the axes denoted by the principal component translates to a high or low contribution. (e) Percentage of the original variance in the data explained by principal components. 50
3.2 (a) The Swiss-roll data in 3D. (b) The PCA projection or scores plot for PC1 vs PC2. (c) The PCA projection or scores plot for PC1 vs PC3. (d) The PCA projection or scores plot for PC2 vs PC3. (e) Percentage of the original variance in the data explained by principal components. 51
3.3 Description of the possible phases for the multi phase oil data by Bishop as well as a plot of the correlation coefficients to visualise the covariance structure in the data. 52
3.4 The plots show the PCA analysis of the multi phase oil flow data. (a) The PCA projection or scores plot for PC1 vs PC2. (b) The PCA projection or scores plot for PC1 vs PC3. (c) The PCA projection or scores plot for PC2 vs PC3. (d) Percentage of the original variance in the data explained by principal components. 52
3.5 Representation of the covariance structure by plotting the correlation coefficients. (a) The 20D and (b) the 60D data set created by GTMs according to the specifications in Table 3.1. 53
3.6 The plots show the PCA analysis of 20D toy data set. (a) The PCA projection or scores plot for the first two principal components. (b) The PCA projection or scores plot for PC1 vs PC3. (c) The PCA projection or scores plot for PC2 vs PC3. (d) Percentage of the original variance in the data explained by principal components. 54

3.7 The plots show the PCA analysis of 60D toy data set. (a) The PCA projection or scores plot for the first two principal components. (b) The PCA projection or scores plot for PC1 vs PC3. (c) The PCA projection or scores plot for PC2 vs PC3. (d) Percentage of the original variance in the data explained by principal components. 55

4.1 The non-linear function $\Xi(\mathbf{x}, \mathbf{W})$ defines a manifold S embedded in the data space given by the image of the latent variable space under the mapping $\mathbf{x} \rightarrow \mathbf{y}$. 62

4.2 (a) The S-shaped 3D test data. (b) A 15x15 GTM with 16 RBF centres which was fit to the S-shaped test data after 50 iterations with the EM algorithm and initialisation with PCA. The GTM manifold has aligned itself to the structure of the data and fits it nearly perfectly. 66

4.3 Projection of simple data sets using the GTM algorithm which was initialised with PCA. The structure of the S-shaped data in (a) is captured and one can clearly see that the class structure is preserved. This is not the case with the Swiss-roll data in (b) where GTM fails to preserve the structure of the classes. 67

4.4 GTM is initialised with PCA and fitted to a simple sine function. The GTM fails to capture the structure of the data regardless of the number of iterations of the EM algorithm. The green circles indicate the uncertainty/variance around the GTM and their size indicates as well that the GTM has trouble to fit the structure of the data. The green line visualises the actual position and structure of the GTM manifold and it is obvious that it is not fitting the data at all. The black line indicates the initialisation which was used at the beginning of the algorithm, which is the first principal component in this case. 69

4.5 GTM is initialised with a beneficial random initialisation and manages to fit the data with convergence of the EM algorithm after 15 iterations. The green circles indicate the uncertainty/variance around the GTM and their size indicates as well that the GTM fits the data very closely. The green circles indicate the uncertainty/-variance around the GTM and their size indicates that the fit is very good. The green line visualises the actual position and structure of the GTM manifold and it is perfectly aligned with the structure of the data. The black line indicates the initialisation which was used at the beginning of the algorithm, which in this case was chosen by tying random vectors until a good fit was achieved. 70

5.1 Changing the leaf order by an internal node flip. The node is marked with a red ring. 82

5.2 Schematic showing the high level design of the VSRMI algorithm. . 86

5.3 Schematic showing the low level design of the VSRMI algorithm: a) Active node and 5 closest points chosen for initialisation of the node. b) Placing of the active node in the data space through an appropriate choice of closest points. c) Overview for all nodes in the data space. 87
5.4 General schematic of a boxplot according to McGillan 1978. 89
5.5 The nearest neighbour label error on the artificial test data with **high (ST=20) and low (ST=2) structure** for the GTM model with different covariance structures. S=Spherical, B=Block, F=Full GTM. PCA=(blue, dotted line with big dot). S-GTM=(green, constant line with X). B-GTM=(red, dashed line with diamond). F-GTM=(black, dashed and dotted line). 92
5.6 The root mean square error for imputation on the artificial test data with **high (ST=20) and low (ST=2) structure** for the GTM model with different covariance structures. S=Spherical, B=Block, F=Full GTM. MI=(blue, dotted line with big dot). S-GTM=(green, constant line with X). B-GTM=(red, dashed line with diamond). F-GTM=(black, dashed and dotted line.) 93
5.7 The negative log likelihood on the artificial test data with **low (ST=2) structure** for the GTM model with different covariance structures. The box plots show the variation of the negative log likelihood based on 100 different and randomly created datasets respectively for each combination of parameters (blocks and dimensions). S=Spherical, B=Block, F=Full and O=Original (thus creating) GTM. The very large box plots in the cases (e) and (f) show that the B-GTM and F-GTM, dependant on the number of blocks, are unstable and show incongruent behaviour with 70 dimensions. 94
5.8 The negative log likelihood on the artificial test data with **high (ST=20) structure** for the GTM model with different covariance structures. The box plots show the variation of the negative log likelihood based on 100 different and randomly created datasets respectively for each combination of parameters (blocks and dimensions). S=Spherical, B=Block, F=Full GTM. The very large box plots in the cases (c),(d),(e) and (f) show that the B-GTM and F-GTM, dependant on the number of blocks and dimensions, are unstable and show incongruent behaviour with 40 and 70 dimensions respectively. 95
5.9 The negative log likelihood on the artificial test data with **high (ST=20) structure** and 30 dimensions for the GTM model where different amounts of variables were *shuffled* into wrong groups. S-GTM=(green, constant line with X), B-GTM=(red, slashed line with diamond). 75% confidence intervall areas are marked by light gray (B-GTM) and dark gray (S-GTM). 97

5.10 The nearest neighbour label on the artificial test data with **high (ST=20) structure** and 30 dimensions for the GTM model where different amounts of variables were *shuffled* into wrong groups. S-GTM=(green, constant line with X), B-GTM=(red, slashed line with diamond). 75% confidence intervall areas are marked by light gray (B-GTM) and dark gray (S-GTM). 97
5.11 The root mean square error on the artificial test data with **high (ST=20) structure** and 30 dimensions for the GTM model where different amounts of variables were *shuffled* into wrong groups. S-GTM=(green, constant line with X), B-GTM=(red, slashed line with diamond). 75% confidence intervall areas are marked by light gray (B-GTM) and dark gray (S-GTM). 98
5.12 The heat-maps of the correlation coefficients for the oil flow data. a) Sorting by the OLO algorithm. b) Sorting by using the grouping of the BCE algorithm. 100
5.13 The heat-maps of the correlation coefficients for the 20D data. a) Sorting by the OLO algorithm. b) Sorting by using the grouping of the BCE algorithm. 100
5.14 The heat-maps of the correlation coefficients for the 60D data.a) Sorting by the OLO algorithm. b) Sorting by using the grouping of the BCE algorithm. 100

6.1 Behaviour of the RMSE and the distribution of the maximal differences between the original and imputed values. The boxplots show the distribution of results for different proportions of missing data on the oil flow data. Each boxplots describes an experiment with 50 different missing data patterns given the proportion of missing data and the used imputation method. The RMSE behaves as expected with an decrease in average performance as well as variance when the proportion of missing data increases. The distribution of the maximum difference (i.e. worst result) also behaves as expected and the errors get worse the more data are missing with a constant variance. 120
6.2 Distribution of the imputed values: little bias is shown. 121
6.3 Performance of the imputation methods with different proportions of missing data $p_i = [0.05, \ldots, 0.6]$ on different data sets. (a) In the case of the oil flow data B-GTM is as good or better than all other methods for little to medium amounts of missing data $p_i = [0.05, \ldots, 0.4]$. (b-c) The GTM generated toy data show a clear advantage for BPCA regardless of the proportion of missing data with B-GTM being the second best method. 123
6.4 Example projection on the multi-flow oil data, with $p = 0.2$. In the PCA projections (a) and (b) it is not possible to distinguish between the classes. In the GTM projections (c) and (d) the performance has deteriorated and the class boundaries are not clear. 125

6.5 Example projection on the multi-flow oil data, with $p = 0.6$. It is impossible to distinguish between the classes in all classes (a)-(d) and in the case of of spherical GTM (c) the algorithm broke down completely. 126

7.1 Schematic illustrating the set-up of a parallel coordinate plot with 5 variables and 2 samples. The values of the samples for each variable are plotted on the dotted line and joined by lines in their respective colours. 132
7.2 a) 2D Structure of the Swiss-roll with outlier group. b) 3D Structure of the Swiss-roll. c-e) Visualisation using PCA, where the structure is visible after looking at cross plots of all possible combinations of available principal components. 134
7.3 Visualisation of Swiss-roll using GTM: a) The projection where 3 sub classes have been marked (identifiable by bigger and red/brown/yellow coloured dots) . b) Parallel coordinates plot of the 3 sub classes. 136
7.4 Visualisation of Swiss-roll using GTM: a) Magnification factors map with the projection plotted into the map. b) Mode projection with distance of points to mean projection. Mean and and corresponding mode joined by a line. 137
7.5 pIGI Utility 139
7.6 pIGI PCA Utility. 140
7.7 a) GTM visualisation for the Barents Sea oils. b) Corresponding magnification factors for the GTM visualisation. In the GTM visualisation one can clearly distinguish between the three classes of oils depending on their origin (Jurassic, Permian, Triassic). When looking at the magnification factor plot it is further apparent that the sample in the upper right of visualisation is a clear outlier. . . . 142
7.8 Parallel coordinates plot for the different clusters identified in the GTM visualisation. In the plot one can identify that the Pristane (Pr) and Phitane (Ph) ratios can be used to distinguish the Permian from the other two classes. Similarly the sterane ratios St29S/R and St29I/R can be used to discriminate between the Jurassic and the other two classes. 143
7.9 a) The scores plot for the first two principal components showing a similar distinction of classes as the GTM. b) The scores plot for the first and third principal component. 145
7.10 a) Histogram showing the contribution of each principal component towards the variance. b) The loadings plot showing how much each variable contributed towards the first two principal components relative to the other variables. 146

7.11 African Data: (a)Average imputation results for different amounts of missing data. (NL) stands for the initialisation with Isomap in case of the GTM. BPCA is shown for the cases where two and 71 principal components are retained. (b) Area of interest in plot (a). It is apparent that BPCA retaining all principal components outperforms all other methods with B-GTM becoming second. BPCA retaining two principal components becoming third at least in the cases where less than 30% of the data are missing. 149
7.12 African data: boxplots for different proportions of missing data p_i showing the spread of the RMSE for the different imputation methods. They verify that the results given by Figure 7.11 are not skewed due to unnatural outliers in the average performance. . . . 150
7.13 North Sea Data: (a)Average imputation results for different amounts of missind data. (NL) stands for the initialisation with Isomap in case of the GTM. BPCA is shown for the cases where 2 and 66 principal components are retained. (b) Area of interest in plot (a). It is apparent that BPCA retaining all principal components and B-GTM perform equally well on this data set, with a slight advantage for B-GTM once higher amounts of data are missing. BPCA retaining only two principal components performs even worse than MRI and deterioates quickly once the amount of missing data goes beyond 30%. 152
7.14 North Sea data: boxplots for different proportions of missing data p_i showing the spread of the RMSE for the different imputation methods. They verify that the results given by Figure 7.13 are not skewed due to unnatural outliers in the average performance. . . . 153

A.1 Plots used to test for the convergence of the QBCE algorithm. a) The distribution of the mean. b) The plots show the energy of the distribution of the mean and one can see that the MCMC algorithm converged for these parameters. 179
A.2 Plots used to test for the convergence of the QBCE algorithm. The plots show the distribution and the energy of sigma and one can see the MCMC algorithm converged for these parameters. 180
A.3 Projection of the 20D toy data set data, with $p = 0.2$ as proportion of missing values. They show that only in the case of B-GTM (c) it is possible to distinguish class boundaries in the projection. 181
A.4 Projection of the 20D toy data set data, with $p = 0.6$ as proportion of missing values. They show that in no case it is possible to distinguish class boundaries in the projection. 182
A.5 Projection of the 60D toy data set data, with $p = 0.2$ as proportion of missing values. They show that the class boundaries a very smeared in all 4 cases but that distinctions still can be made. 183

A.6 Projection of the 60D toy data set data, with $p = 0.6$ as proportion of missing values. They show that in no case it is possible to distinguish class boundaries in the projection. 184

B.1 Deconstruction of data space into subspaces (projections times loadings $pc_1 \times u_1$) and in the case of pruning (i.e. the omitting of negligible eigenvectors) with an error matrix E for the residuals. 186

B.2 a) The two principal components in a two dimensional data cloud. b) The two principal components for the S-shaped data marked as red and yellow bar. 196

B.3 a-b) The plane which is spanned by the two principal components for the S-shaped data from different angles. 197

B.4 Demonstration of the projection result of the PCA algorithm on simple data. The structure of the S-shaped data in (a) is captured and one can clearly see that the class structure is preserved. This is not the case with the Swiss-roll data in (b) where PCA fails to preserve the structure of the classes. 198

B.5 Demonstration of the projection result of the GPLVM algorithm on simple data. The structure of the S-shaped data in (a) is captured and one can clearly see how that the class structure is preserved. This is also the case with the Swiss-roll data in (b) where GPLVM, thanks to the Isomap initialisation, manges to capture the structure of the data correctly. As comparison the original Isomap projection used to initialise GPLVM can be found in Figure B.7. 199

B.6 Triangle with points A,B,C and the distances between them marked by (AB) and (BC). Given that in this connected graph there are only edges between A,B and B,C the geodesic distance between A and C is given by (AB) + (BC). 200

B.7 Demonstration of the projection result of the Isomap algorithm on simple data. The structure of the S-shaped data in (a) is captured and one can clearly see how the class structure is preserved. This is also the case with the Swiss roll data in (b) where Isomap manges to capture the structure of the data correctly. 201

List of Tables

2.1 Summary of the different kerogen types. 32

3.1 Summary of the specifications for the GTM respectively generating the 20D and 60D toy data. 49

4.1 Overview of the characteristics of the different algorithms. A 'Y' indicates the algorithms exhibits the quality, an 'E' indicates that there is an extension or heuristic to the algorithm that exhibits the quality. The different characteristics are: *Probabilistic:* Is the method based on a probabilistic framework? *Missing Data:* Can the method deal with missing data? $\mathbf{Y} \rightarrow \mathbf{X}$: Does the method provide a mapping function from data to visualisation space? $\mathbf{X} \rightarrow \mathbf{Y}$: Does the method provide a mapping from the visualisation to the data space? *Non-Linear:* Does the method allow for non-linear mappings? *Local Structure:* Does the method allow to explore local structures based for example on a connected graph? 73

5.1 Results after training GTM with different initialisations. The best results for each model type are marked in bold. The results for the BGTM D60 data are included just for completeness since all algorithms broke down when processing them as can be seen by the number of iterations. The results indicate that in all cases the VSRMI initialisation is superior to the old initialisation using the first two principal components of the data. The difference in the results is very relative and thus the important aspect of the table is the clear trend, which is in favour of the VSRMI. 102

7.1 RMSE for African data leave-one-out-cross-validation. The information in brackets relates to the initialisation in case of GTM and to the number of retained principal components in case of BPCA. . 148

7.2 RMSE for North Sea data leave-one-out-cross-validation. The information in brackets relates to the initialisation in case of GTM and to the number of retained principal components in case of BPCA. 151

7.3 Runtime until termination of the algorithm for the different methods in seconds on the different data sets. 155

7.4 Table showing the average runtime in a single step for two different parts of the GTM algorithm. The analysed parts are the two most computationally intense parts, namely the EM-Step and the added heuristic to stabilise the algorithm. It is apparent that the heuristics take up most of the time when the dimensionality of the data increases. 156

To my family and everyone who supported me.

Acknowledgements

First of all I want to thank my two supervisors Ian T. Nabney and Dan Cornford for their constant support, guidance and encouragement. The achievement of finishing this thesis within 3 years was possible only because of their continuous efforts. Especially Ian encouraged me to keep the work to the essentials and within the focus. Both of them always had an open ear for my ideas, helping me to distinguish the sensible and practical ideas from the ones which are better left for later. Our regular meetings helped me to order my thoughts and in particular the long drives down to Devon with Dan helped to clarify some of my ideas. The team of the two of them made the last 3 years a very rewarding and enjoyable experience.

Another big thank you goes to all my fellow peers and friends in the NCRG, across the school of engineering and Aston University. They made my Ph.D. the most enjoyable time of my life. My office mates Alexis Boukouvalas, Erik Casagrande, Harry Goldingay, Jack Raymond and Michel Randrianandrasana were always open for constructive discussions. Especially at the beginning of my Ph.D. they helped me greatly to understand the complicated subject matter. A very special notes has to go to Paul Knowles, Terence Broderick , Antonio Geraldo de Paula Oliveira and Jan Duracz who helped me to found the Engineering Postgraduate Research Society. In our countless committee meetings, which regularly went off topic, they supported me and listened to the occasional rants about certain problems with software libraries, system administrators or the other things which regular stress the life of the common Ph.D. student. Our work in the society was a very welcome deviation from my own research, great fun and challenge at the same time. Not to mention many brilliant events we organised, which made not only my Ph.D. a far more enjoyable experience. Also Ben Tocher should not be forgotten for his legendary email.

I want to thank everyone at IGI Ltd. for their continuous support and efforts. My stays in Hallsannery in lovely Devon were very fruitful, inspiring and thanks to the remote location and lovely countryside felt more like a vocation than work. A special thanks goes to Chris Cornford, Paul Farrimond and Andy Mort for their patience when explaining to me the workings of geochemistry and for the countless discussions about the possible applications of my ideas in geochemistry.

I want to thank my family for their continuous encouragement, backing of my ideas and enforcing some slight pressure from time to time to make sure I will finish in time.

I want to thank my many friends who always fully supported me. They kept me balanced and pulled me out of my work and back into the "normal" world on a regular basis. Of all these friends there are some to which I am especially thankful which includes Cord Drews, my flatmate and the first to know about whatever thrilled or upset me, Khalid Mia, who got me into fencing while being at Aston, Irini-Alexia Terzakis, which helped me to found the German Society, Caro Randolf, for starting Kitesurfing with me, and finally the Latinmotion crew with whom I spend many of my nights enjoying Salsa music and dance.

I am very grateful to Vicky Bond, NCRG Co-ordinator and star behind the scenes at the NCRG. Without her it would have been impossible for me to deal with all the small and big administrative tasks. Regardless of the non-scientific problem I was facing at the University, Vicky had the answer.

I want to thank both examiners of my Ph.D., David Lowe and Olav Kvalheim. Their input and and suggestions greatly improved the structure and presentation of this thesis.

Finally I want to thank the EPSRC and IGI Ltd. for funding my studentship under the CASE scheme.

1 Introduction

CONTENTS

1.1 Data Exploration/Visualisation and GTM 20
1.2 Petroleum Exploration 22
1.3 Motivation 23
1.4 Project Aims 23
1.5 Report Overview 24
1.6 Publications resulting from this thesis 25
1.7 Notation 27
1.8 Abbreviations 28

In all sciences the amount of available data is steadily growing. Increasing capabilities of analysis methods and decreasing costs for the capture, processing and storage of data are likely to further increase this trend in the future. For example, in chemistry and geology modern technologies allow samples to be automatically analysed for chemical composition or biomarkers. In molecular biology, micro-array analysis allows access to large quantities of data. The method is used to decode the information stored in the DNA of living organisms and for large scale experiments to test the reactivity of biologically active compounds helping with the development of drugs or in understanding and fighting diseases. Financial markets are getting ever more complex with vast amounts of different economic indicators, products and price quotes allowing for arbitrage opportunities if one can find and distinguish new and emerging patterns and trends early enough. The internet created a new challenge for semantic data mining: for example the tasks of classification, categorisation and exploration of the huge amounts of documents in databases like Google Scholar, personalised profiles on social networking sites like Facebook or short paragraphs of content like in Twitter.

To cope with the vast amount of information and find patterns as well as underlying processes, rules and structures the available data are analysed with the aid of methods from statistics and mathematics to help and guide the analysts. This analysis supports the discovery of regularities or irregularities, which are hard to find when looking at the raw data in tables of numbers, symbols, text or images. In statistics common methods like multidimensional scaling or principal component analysis are quite sophisticated however they lack the ability to account for noise in the data and cannot deal appropriately with missing values. Missing values occur for different reasons but exist in most real world data sets. Some reasons for missing data are:

- incomplete or censored recordings;
- errors in automated machines or algorithms;
- mistreatment of samples or contaminated samples;
- export from analog to digital;
- diverse treatment of samples (using different analysis techniques);
- lack of response (from humans in surveys);
- other forms of human failure.

Many existing mathematical and statistical analytical algorithms cannot treat these data without prior processing. The most common method to deal with these incomplete samples is to either delete the sample itself or exclude a whole category/variable from the analysis if too many samples are missing this entry (Schafer, 1997). Generally this is neither sensible nor desirable since a lot of valuable information is lost and, far worse, one might introduce a serious bias to the

end results if the underlying process, which generated these missing values is just ignored (Little and Rubin, 2002). In the scientific literature better methods have been proposed but mainly with reference to specific domains like census problems, population surveys or genetics while neglecting their possible application to other sciences and the impact on the field of data visualisation (Raghunathan *et al.*, 2001; Oba *et al.*, 2003).

1.1 Data Exploration/Visualisation and GTM

Capturing structures in high-dimensional data is a complex and tricky task. Human visual understanding of geometry ends at three or arguably with the help of advanced plotting mechanisms like colour, size and shape of the plotted points, at up to six dimensions. Thus finding structures with direct visualisation alone is not possible on data sets with more than three to six dimensions.

One way to get over this dilemma is to project this high-dimensional data onto a low-dimensional representation while preserving as much information about the structure as possible. This low-dimensional representation is usually two-dimensional to be representable on screen or paper and is often called the *visualisation space* or *latent space*. This way the human analyst can look at the data and spot eye-catching structures. There are many possible ways to obtain such a low-dimensional representation and the best choice depends on many factors like the data, the application and the practitioner. A general introduction to these methods is given in chapter 4.

Another way to find structures is to use clustering algorithms like K-means clustering (Hartigan and Wong, 1979) or Gaussian Mixture Models (Nabney, 2002). These algorithms try to find groups in the data set and cluster the points to them accordingly. The major problem of these algorithms is the definition of good clustering and complexity criteria since it is usually neither known how many clusters there are in the data nor how the borders between the clusters should be drawn.

In conjunction with cluster algorithms, visualisation methods can help to specify the parameters like the number of classes. Conversely cluster algorithms may help to validate the visualisation. Ideally one would employ both methodologies in the exploration of unlabelled data. However in this thesis the focus will be on visualisation algorithms.

A method of particular interest is the Generative Topographic Mapping (GTM) introduced in chapter 4. It can be described as a constrained Gaussian Mixture Model where the Gaussians are connected via a two-dimensional flexible grid. A good analogy for the GTM is a flexible rubber-sheet which stretches and bends itself to fit the data as good as possible. The result is a probabilistic latent trait model for data visualisation which indicates topographic rather than real distances between points. It was developed by Bishop *et al.* (1996) at the NCRG, Aston University with the goal of improving the Self Organising Map (SOM) (Kohonen, 1995) by a more principled probabilistic method. The algorithm has a

successful track record in various applications, including the following examples:

- A study about semantic space models utilised multiple randomly initiated GTM models to create a set of new lower dimensional spaces which were analysed for common structures and patterns (Lowe, 2001).
- Comparison of dimensionality reduction methods for wood surface inspection where GTM was found to be preferable for the purpose of interactive classification by humans (Niskanen and Silven, 2003).
- Outlier detection in scatterometer data used for predicting wind vectors (Bullen *et al.*, 2003).
- Dimension reduction in speech analysis using voice morphing technology to enable users to transform one person's speech pattern into a different pattern with distinct characteristics while preserving the original meaning (Orphanidou *et al.*, 2003).
- Condition monitoring and fault diagnosis of a gearbox (Liao and Shi, 2004).
- Analysis of microarray data (Grimmenstein *et al.*, 2004).
- Training of an AI player to play pong using a GTM latent space obtained from observation data recorded from games played by humans (Leen and Fyfe, 2005).
- Identification and visualisation of clusters formed by motor unit action potentials (MUAPs) to aid with investigations seeking to explain the control of the neuromuscular system (Andrade *et al.*, 2005).
- Data visualisation during the early stages of drug discovery (Maniyar *et al.*, 2006a).
- Handling outliers in brain tumour magnetic resonance spectroscopy (MRS) data analysis through robust topographic mapping (Vellido and Lisboa, 2006).
- An application involving missing data and a decision support system to assist water managers with their decision making tasks when exploring the ecological status of human altered streams (Vellido *et al.*, 2007).
- Word segmentation of handwritten text using supervised classification techniques (Sun *et al.*, 2007).
- Investigation of the existence of abandonment routes (ways customers leave the actual provider) in the Brazilian telecommunications market, according to the customers' service consumption pattern where GTM was used to learn families to segment and visualise the data, as well as to identify typical churn routes (García *et al.*, 2007).

This thesis centres around the use of visualisation tools to enhance the analyses of geochemical data in the presence of missing values. GTM is the method of choice due to its probabilistic and non-linear nature allowing it to be expanded and modified for the needs of the practitioners. The model further allows the use of different diagnostics to extensively analyse the obtained visualisation. Two major extensions for GTM will be proposed and the actual utility of the method as well as outstanding issues and problems will be highlighted and discussed.

1.2 Petroleum Exploration

Crude oil and refined petroleum products are some of the most important resources in the modern industrial world. They include fuels in cars, aircrafts, ships or electricity generators, lubricants in machinery, asphalt for road building and the manufacturing of all sorts of synthetic materials in the chemical industries like plastics. They play an important role in our modern society and the demand is rising steadily together with price and the need for the exploration of more oil fields.

In ancient times oil was collected at the earth's surface. Today oil is being produced from accessible underground reservoirs and it is becoming increasingly difficult to locate new areas which could have produced hydrocarbon-impregnated rocks and thus oil. Geologists all over the world look at aerial and satellite images, examine rocks and take samples to detect if there is a chance of finding oil producing source rocks in a certain area. Then geophysicists study the physical properties of the subsurface using various methods like gravimetry and magnetometry to decide if the underlying strata are likely to contain traps or faults that could be filled with hydrocarbons.

Then all the results are accumulated and studied to decide if it is sensible to build a well and drill into the earth to collect samples of cuttings as well as cores to analyse them further. At this point the geochemical analysts get involved. They analyse the composition of the cuttings and cores to estimate if the sampled rocks have the potential to be source rocks. If this is the case the next step is the modelling of the basin based on the stratigraphy. The goal is to predict possible reservoirs for oil and gas and based on these predictions to drill new wells. As soon as a well finds oil or gas the work is not yet done. To commercially exploit any reservoir one has to determine how big it is and answer the question how and where the oil was generated. Depending on the structure in the subsurface and the composition of the oil one might be able to find more oil reservoirs when one can trace the source or sources.

For example in one possible scenario the source rock has expelled oil into different directions and one has found only one of many traps with oil entrapped in it. In another scenario the oil in the actual trap is being loaded by two or more different source rocks, where scenario one gets even more likely. To answer these questions a geochemist tries to perform an oil-oil or oil-source rock correlation.

1.3 Motivation

A geochemist performs exploratory analysis of petroleum geochemical data to find common patterns to distinguish between different source rocks, oils and gases. The aim is to explain the source of the petroleum together with its maturity and any intra-reservoir alteration. However, at the outset, the geochemist is typically faced with a large matrix of samples each with a range of molecular and isotopic properties, with a spatially and temporally unrepresentative sampling pattern, noisy data, and often a large number of missing values. This inhibits analysis using conventional statistical methods. Typically visualisation methods like principal components analysis are used but these methods are not easily able to deal with missing data and they struggle to capture global non-linear structure in the data.

Another approach to discovering complex, non-linear structure in the data is through the use of linked plots, or brushing, while ignoring the missing data. These approaches do not make the most use of the available information and further are likely to miss possible non-linear relations and structures.

The realm of machine learning and probabilistic inference offers a number of well studied frameworks to fit a probabilistic non-linear model to data, which can cope naturally with the missing data, for example exploiting the EM algorithm. These methods offer the ability to deal with the above mentioned problems. The motivation of this project was to complement the known chemometrical approaches on dealing with the issue of missing data and non-linearity in visualisation by employing and enhancing well known machine learning algorithms. For example one obvious approach is to integrate geochemical expert knowledge into these models via a range of mechanism including the covariance structure of the data. By doing so we hope to support the geochemical analyses and this motivated the project.

1.4 Project Aims

The overall goal of this project has been to develop new and improved statistical methods for the analysis of complex geochemical data. The aim of this thesis is to provide an outlook on their possible application and expected performance on geochemical data with and without missing values. More specifically the aims can be split into the following objectives:

- Assess the the capabilities of GTM to deal with missing data and compare these with competing alternative methods.
- Assess the implication of missing data on the visualisation capabilities.
- Automate, improve and assess the initialisation of GTM.
- Evaluate the possibility of including prior information into GTM and assess if this results in improved model fit and quality of the projection.

- Review extensions of the GTM regarding their usefulness in modelling and exploring geochemical data.
- Apply GTM and its known and newly developed extensions to geochemical data and assess if the usage of these methods improves the exploration and modelling of this data.
- Explore how GTM and other multivariate methods can be used to help with the exploration of geochemical data in a real-life environment

To meet these objectives the Netlab (Nabney, 2002) toolbox was used and where possible modified. For the following known and public algorithms no source code was available and the algorithms were therefore implemented as part of this project (EM for missing data (GMM and GTM) and Bayesian Correlation Estimation). The novel extensions and heuristics were all implemented in Matlab and also partly in C++ for the usage in pIGI, a computer programme employed by IGI to explore geochemical data. IGI is a geochemical consulting company and the industrial CASE partner for this project.

1.5 Report Overview

This thesis is structured into 8 chapters, the remainder of which are organised as follows:

Chapter 2 introduces the basics of geochemistry. It gives an overview of the process for obtaining the data and the commonly used techniques to analyse the data. It concludes with a overview of the usage of multivariate and statistical methods in the subject area.

Chapter 3 establishes and explains the different toy data sets which are used to illustrate, analyse and benchmark the different algorithms and methods introduced in this thesis.

Chapter 4 defines multiple models and algorithms for data exploration. Their general benefits are discussed and their usability and differences are demonstrated using the toy data sets established in chapter 3. The list of models includes GTM, PCA, PPCA, KPCA, GPLVM, Neuroscale and Isomap.

Chapter 5 proposes two novel extensions of the GTM algorithm. The chapter starts with a general overview of all known extensions of GTM. It continues with two novel extensions for GTM. The first extension deals with the embedding of prior knowledge about the covariance structure into the algorithm and is called block GTM (B-GTM). A modification for the EM algorithm is discussed as well as two techniques to obtain this information. The second extension called VSRMI, visualisation space reverse mapping initialisation,

enables the GTM algorithm to utilise any 2D mapping of a dimension reduction algorithm to initialise itself on the data. This allows GTM for example to avail itself of local geometry embedding techniques like Isomap to fit complex data which can not be fit by using the conventional initialisation via PCA.

Chapter 6 comprises of a benchmark study of GTM, the block extensions to GTM and selected imputation algorithms to assess their capabilities when dealing with missing data. First the theoretical basis of treating missing data is reviewed. Then a short summary of the most commonly used imputation methods is given. Afterwards selected imputation methods are introduced in more detail which cover a broad range of available techniques. The extension of the EM algorithm in the presence of missing data is then outlined and consequently extended for GTM and B-GTM. Finally all methods are benchmarked against each other using the toy data sets and artificially created missing data patterns with 10% to 60% missing data.

Chapter 7 outlines the application of the discussed methods on real geochemical data sets. It starts off with discussing the structure of the data, their pre-processing and possible problems that one may encounter. Then a short overview is given about the actual implementation and integration of PCA and GTM into pIGI and other utilities. This is followed up by an introduction about how one can use the non-linear mapping of GTM for exploratory analysis of data in real applications. Subsequently three real geochemical data sets are used. Due to confidentiality issues these are discussed in varying degrees of detail. The first data set comprises of Barents Sea oils from the public domain where GTM was used to identify three clusters of oils from biomarker data. The second and third data sets are then used for a benchmark and performance study to assess the novel extensions of GTM.

Chapter 8 concludes the thesis with a discussion and summary of the main results. Finally a survey of the open questions and possible future research motivated by this thesis is presented.

1.6 Publications resulting from this thesis

Refereed international journal papers

- M. Schroeder, D. Cornford, P. Farrimond, and C. Cornford 2008. Addressing missing data in geochemistry: A non-linear approach. *Organic Geochemistry* 39, 1162- 1169.

Refereed international conference papers

- M. Schroeder, D. Cornford and I.T. Nabney, 2009. Data visualisation and exploration with prior knowledge. International Conference On Engineering Applications Of Neural Networks 2009, *CCIS 43*, pp. 131-142, 2009.

Conference talks

- On "Exploring Geochemical Data with Missing Values" at the *Natural Computing Applications Forum Meeting January 2008.*
- On "Data Visualisation and Exploration with prior knowledge about grouping in the covariance structure" at the *Young Statisticians' Meeting 2009.*

Conference poster presentations

- On "Exploring geochemical data using non-linear projection methods" at the *International Meeting on Organic Geochemistry 2007.*
- On "High-dimensional Data Imputation and Visualisation: a Geochemical Case Study" at the *Centre for Research in Statistical Methodology workshop on Bayesian Analysis of High Dimensional Data April 2008.*
- On "High-dimensional Data Imputation and Visualisation: Application in Geochemistry" at the *Royal Statistical Society Conference 2008.*
- On "Data Visualisation and Exploration in Geochemistry with prior knowledge about grouping in the covariance structure" at the *International Meeting on Organic Geochemistry 2009.*

Technical reports

- M. Schroeder and D. Cornford 2007. Data Visualisation with Missing Data: A Non-Linear Approach. *Technical report NCRG/2007/04*, Aston University, Birmingham.
- M. Schroeder, I.T. Nabney and D. Cornford. Block GTM: Incorporating prior knowledge of covariance structure in data visualisation. *Technical report NCRG/2008/006*, Aston University, Birmingham, 2008.

1.7 Notation

To ease reading a consistent notation is used throughout the thesis. A bold upper-case letter $\mathbf{M}$ is used to indicate a matrix of n rows and m columns where each element in the matrix is indicated through the subscripts $i = 1, \ldots, n$ and $j = 1, \ldots, m$ therefore M_{ij} is the element in row i and column j of the matrix.

A bold lower case letter $\mathbf{x}$ is used to indicate a vector of size n, where each element in the vector is indicated through the subscript $i = 1 : n$ and therefore x_i is the i-th element of the vector. However $\mathbf{x}_j$ is the j-th vector or data point. In the case where a list of multi-dimensional data points needs to be indicated a matrix is used with data points are aligned as row vectors.

The superscripts o (observed) and m (missing) are used for matrices and vectors alike and indicate the observed or missing values respectively. These may be different for different vectors, matrices or data points, e.g. $(\mathbf{M}^o, \mathbf{x}^o)$.

Functions are indicated by an upper-case Greek letter like Θ while the corresponding variables or parameters are indicated by lower-case Greek letter like θ. Further $\mathbf{x}$ is used to indicate location in the unobserved latent space while $\mathbf{y}$ is used to indicate data in the observed feature space.

Short Version:

$\mathbf{M}$ = matrix
$\mathbf{Y}$ = List of vectors
$\mathbf{x}$ = vector
$\mathbf{x}^o$ = observed part
$\Theta(\theta)$ = Function of θ
$\mathbf{x}$ = latent space values
L = dimension of latent space
K = number of Gaussians or centres
B = number of Radial Basis Functions (RBFs)
$l(\theta)$ = data likelihood depending on θ

M_{ij} = element of matrix
$\mathbf{y}_j$ = single vector
x_k = element of vector
$\mathbf{x}^m$ = missing part
θ = variable/parameter
$\mathbf{y}$ = data space observations
D = dimension of data space
N = number of data points

$-L = -\ln[l(\theta)]$ (Negative data log likelihood)

1.8 Abbreviations

The text contains a lot of abbreviations to make the text more concise. The following table contains a list of all used abbreviations:

ARMSE	Average Root Mean Square Error
BCE	Bayesian Correlation Estimation
B-GTM	Block Generative Topographic Mapping
BPCA	Bayesian Principal Component Analysis
DVMS	Data Visualisation and Modeling System
EM	Expectation Maximisation
F-GTM	Full Generative Topographic Mapping
GC-MS	Gas Chromatography Mass Spectrometer
GC	Gas Chromatography
GMM	Gaussian Mixture Model
GPLVM	Gaussian Process Latent Variable Model
GTM	Generative Topographic Mapping
GTMI	Generative Topographic Mapping Imputation
IGI	Integrated Geochemical Interpretation
KPCA	Kernel Principal Component Analysis
MCMC	Markov Chain Monte Carlo
MDS	Multi Dimensional Scaling
MI	Mean Imputation
MRI	Multiple Regression Imputation
MRS	Magnetic Resonance Spectroscopy
NCRG	Non-linearity and Complexity Research Group
NIPALS	Non-linear Iterative Partial Least Squares
NL	Non-linear
NLL	Negative Log Likelihood
NNLE	Nearest Neighbour Label Error
MLP	Multilayer Perceptron
OLO	Optimal Leaf Ordering
PCA	Principal Component Analysis
PLS	Partial Least Squares
PPCA	Probabilistic Principal Component Analysis
QBCE	Quick Bayesian Correlation Estimation
RBF	Radial Basis Function
RMSE	Root Mean Square Error
S-GTM	Spherical Generative Topographic Mapping
SCG	Scaled Conjugate Gradient
SOM	Self Organising Maps
SRI	Sequential Multiple Regression Imputation
ST	Structure (Standard Deviation around a Gaussian)
VSRMI	Visualisation Space Reverse Mapping
WMI	Weighted Mean Imputation

2 Geochemistry

CONTENTS

2.1 Petroleum Generation and Entrapment 31
2.2 Petroleum Geochemistry 34
2.3 Gas Chromatography-Mass Spectrometry (GC-MS) 36
2.4 Chemometrics in Geochemistry 40
2.5 Summary 45

Geochemistry (oldgreek $\gamma\eta$ ge = Earth , $\gamma\eta\omega$- geo- = concerning Earth, $\chi\eta\mu\epsilon\iota\alpha$ chemeia = Chemistry) addresses the material composition, distribution, stability as well as the cycle of chemical elements and their isotopes in minerals, rocks, soil, water and the atmosphere. This applied science combines geosciences as the object of investigation with chemistry as the examination method.

2.1 Petroleum Generation and Entrapment

This thesis focuses on the application of visualisation algorithms to petroleum geochemistry and thus enhance the existing analyses methodology.

The aim of this chapter is to give the reader a short introduction into the area of petroleum geochemistry. The modern geochemist is faced with evermore data due to the advances of technology. To help him to explore his data was the major motivation of this research. This requires a basic understanding of the processes involved in the generation and accumulation of petroleum to appreciate the questions a geochemist faces and wants to answer.

A precondition for the existence of oil is the generation of hydrocarbons if one neglects the possibility of abiological sourcing of oil and gas (Gold, 1985). In geochemistry hydrocarbons refer to decomposed biological tissue, consisting mainly of carbon, hydrogen and oxygen with often significant quantities of sulphur, nitrogen, trace metals, and other elements. They are generated within organic-rich sediment, for example coal and bituminous shale. These sedimentary rocks are called *petroleum source rocks* if they have the potential to generate significant quantities of hydrocarbons. Many factors can affect the deposition of a potential source rock. As illustrated in Figure 2.1, a suitable type and a sufficient quantity of organic matter needs to be produced, be transported, survive and be buried. The mixture and types of organic chemical compounds in the source rock will, under the right maturity conditions, result in the production of bitumen and kerogen. The type of kerogen determines if it will release gas or oil. Kerogen based on marine organic matter is more prone to oil generation than those of terrestrial origins. A more detailed explanation can be found in Table 2.1.

After the accumulation of organic matter the next important step in the generation of petroleum is the thermal maturation of the kerogen as illustrated in Figure 2.2. Three broad stages, called diagenesis, catagenesis and metagenesis (Horsfield and Rullkotter, 1994), are involved in this processes.

The diagenesis or early maturation processes occur at low temperatures. The alteration of the organic tissue is mediated primarily by biological rather than thermal processes. Decomposer communities rapidly recycle amino acids, simple peptides and carbohydrates. Larger, insoluble proteins and polysaccharides need to be broken down by bigger microorganisms first, before they can be assimilated back into the biomass. In this way, degradation products from one group of organism pass through as food for other groups and in the end anaerobic archaebacteria, the methanogens, digest simple organic compounds and produce methane, biogenic gas. This results in the production and release of volatile prod-

Kerogen Types		
Type	Composition	Potential
I (Liptinite)	alginite, amorphous organic matter, cyanobacteria, freshwater algae, and land plant resins	very oil prone
II (Extinite)	pollen and spores, terrestrial plant cuticle, terrestrial plant resins and animal decomposition resins, terrestrial plant lipids and marine algae	oil and gas prone
III (Vitrinite)	terrestrial plant matter that is lacking in lipids or waxy matter	gas prone, low oil potential
IV (Inertite)	decomposed organic matter	no potential

Table 2.1: Summary of the different kerogen types.

ucts like water, carbon dioxide and lesser amounts of carbon monoxide. Another product of this stage is kerogen. It is an insoluble residue and consists of resistant biomacromolecules like cutan, algaenan and lignin and some of the simple alteration products from biological degradation which escaped consumption to undergo further maturation during diagenesis.

Catagenesis follows diagenesis, and in this phase kerogen is subject to high temperatures over a long period of time. This maturation stage involves oil formation and subsequently wet-gas generation, due to the breaking of chemical bonds in kerogen which leads to the break off of smaller molecules from the bulk. These small molecules eventually become petroleum and natural gas. Increasing pressure and compaction also causes various physical changes of the kerogen which accompany the chemical alterations. Some of these changes can be measured later on and allow the geochemist to judge the extent to which kerogen maturation has proceeded.

Finally during metagenesis one observes gas production at high thermal stress. As the kerogen is buried deeper in the Earth, temperature and pressure will rise which results in the oil to gas condensate at 165°C, the remaining condensate is converted to dry thermogenic gas at 175°C to 300°C, which is the theoretical top of the thermodynamic stability field for methane.

From a petroleum explorers point of view, hopefully the oil never reaches the stage of metagenesis but instead escapes from its source rock. This step is called expulsion and is directly followed by migration where the oil travels from the source rock to a reservoir or the surface. The migration can be broken down into three sequential processes (Roberts and Cordell, 1980) as illustrated in Figure 2.3 (based on Mann *et al.* (1991)). During primary migration the hydrocarbons are expelled from kerogen particles into silty or sandy liminae or fractures within the source rock followed by the drainage of aggregated hydrocarbons into the carrier beds. During secondary migration the hydrocarbons travel from the source rock

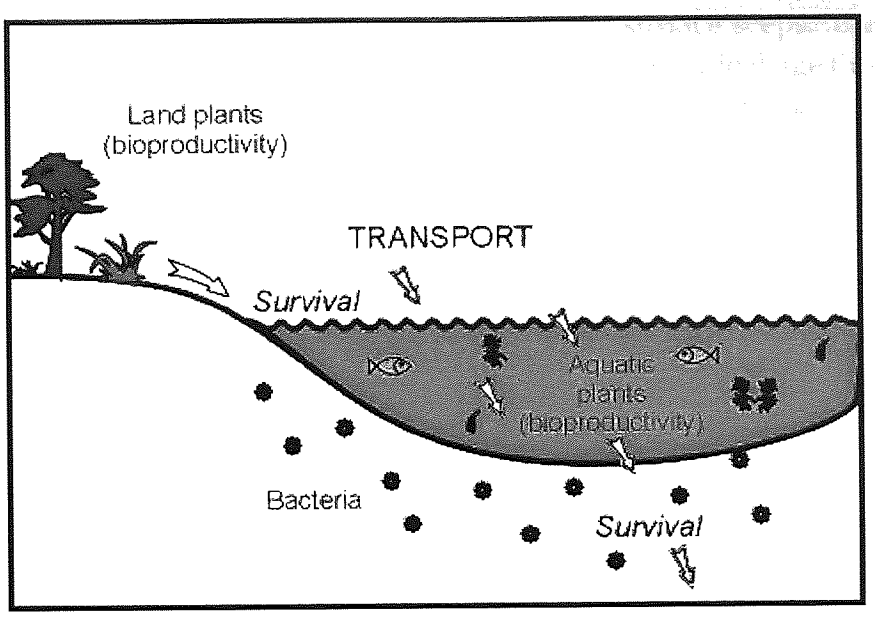


Figure 2.1: The accumulation of organic material can only happen if it is not decomposed and thus preserved. The material can originate from land plants, aquatic plants or bacteria in the ground. *With permission of IGI Ltd.*

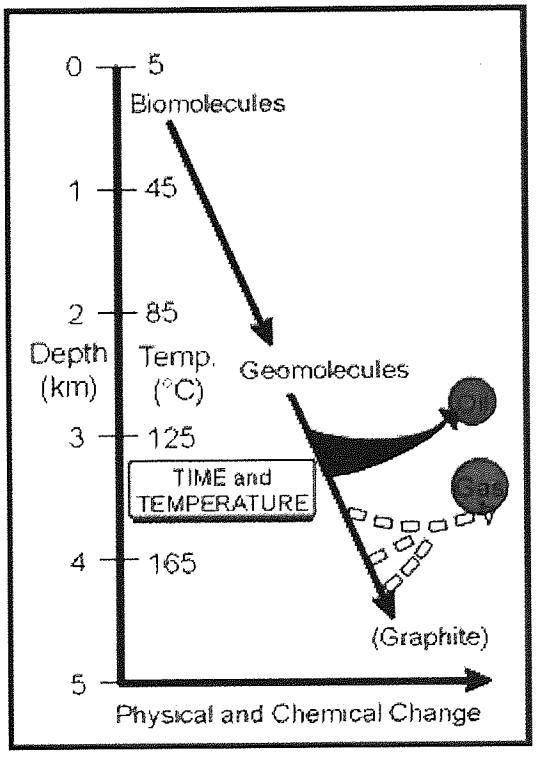


Figure 2.2: The maturation of source rocks is a complex chemical and physical reaction which is influenced by the depth and temperature of the rock. Depth and temperature are highly related and thus can be plotted on the same axis. Certain depth and temperature windows are associated with the formation of oil, gas and graphite. *With permission of IGI Ltd.*

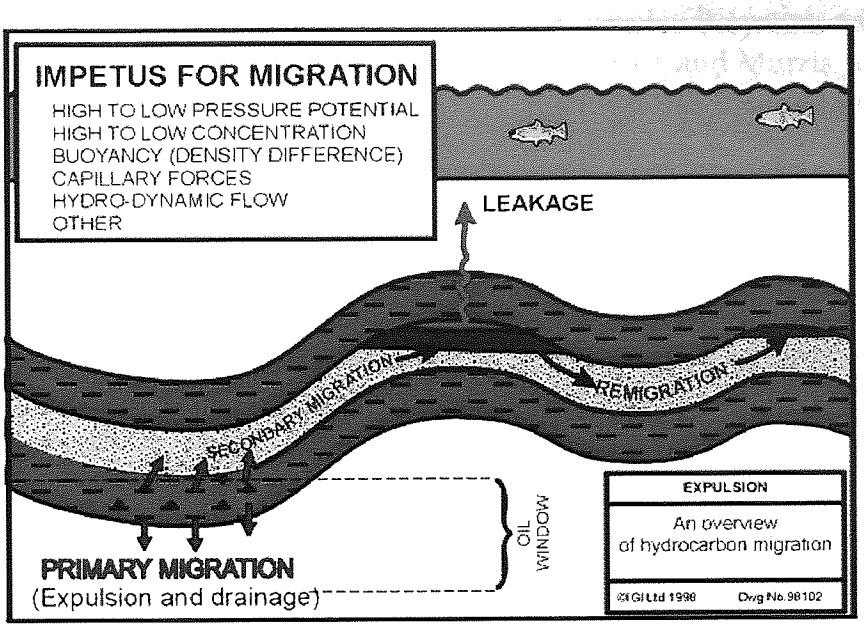
unit via more porous carrier beds to the reservoir or surface seeps. During tertiary migration the hydrocarbons again leave the reservoir by leakage through the seal, or by spill or by displacement by subsequent hydrocarbon charges to other reservoirs or to surface seeps. These reservoirs are usually made of sandstones and fractured limestones in which hydrocarbons accumulate in commercially exploitable quantities, and which usually have no source-rock potential.

2.2 Petroleum Geochemistry

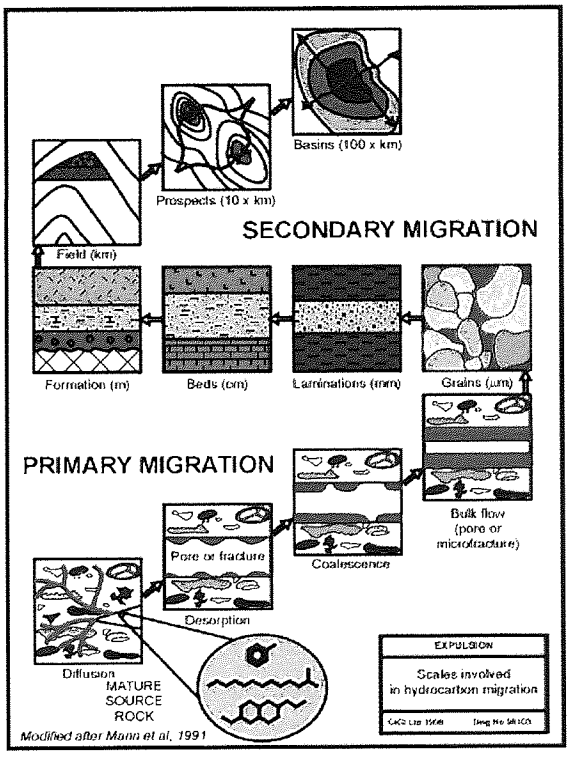
Petroleum geochemistry has a range of applications in the exploration for oil and gas reserves. Some of the issues it helps to address are:

- **Petroleum potential**: Identifying source rock potentials and the type and quantities of kerogen.
- **Petroleum generation**: Assessing the source rock maturity to narrow the possible period of hydrocarbon generation.
- **Petroleum expulsion**: Estimating how much petroleum is expelled from the source rock
- **Petroleum migration**: Predicting possible losses and alteration of hydrocarbons on route to the reservoir.
- **Petroleum entrapment**: Recreating the filling history of reservoirs and trap formation in relation to the timing of petroleum generation.
- **Petroleum survival**: Analysing conditions to find out where hydrocarbons were or were not preserved against the destructive forces of oxidation, cracking and biodegradation.

To deal with these questions the petroleum geochemists are usually provided with a range of data obtained through different sampling and screening techniques. They then use bulk oil properties, such as total petroleum hydrocarbon content, visual comparison of gas chromatography-mass spectrometry (GC-MS) chromatograms, concentrations of source-specific markers, bar plots of the distributions of oil characteristic pentacyclic aromatic hydrocarbons and lists and cross plots of diagnostic ratios like the hydrogen carbon index, to make deductions about the maturity, origin, composition and correlation of oils and source rocks. In this thesis the main focus will be on biomarker data. Biomarkers are obtained through GS-MS which is explained later in this chapter. In geochemistry, biomarkers are molecules which indicate the existence of past living organisms. They are reported to give information on source, maturation, migration and biodegradation (Seifert and Moldowan, 1981). This makes them a valuable tool in oil-source-rock correlation which helps to link a petroleum family to a stratigraphic unit, facies and/or locality containing the source kerogen (Curiale, 1994).



(a) Migration and Entrapment.



(b) Primary and Secondary Migration.

Figure 2.3: Oil and gas movement in the subsurface occurs in various stages: (1) A newly generated molecule moves away from a kerogen particle down a pressure and/or concentration gradient into a micropore in the source rock. (2) The molecules accumulate to an oil droplet and move through fracture or intergranular porosity or perhaps by diffusion through the kerogen network. (3) Following carrier beds the oil moves to the surface as an oil seep or gets entrapped in a basin. *With permission of IGI Ltd.*

This correlation is an essential part of analysing a petroleum system and assists in the identification of undiscovered resources (Demaison and Murris, 1984).

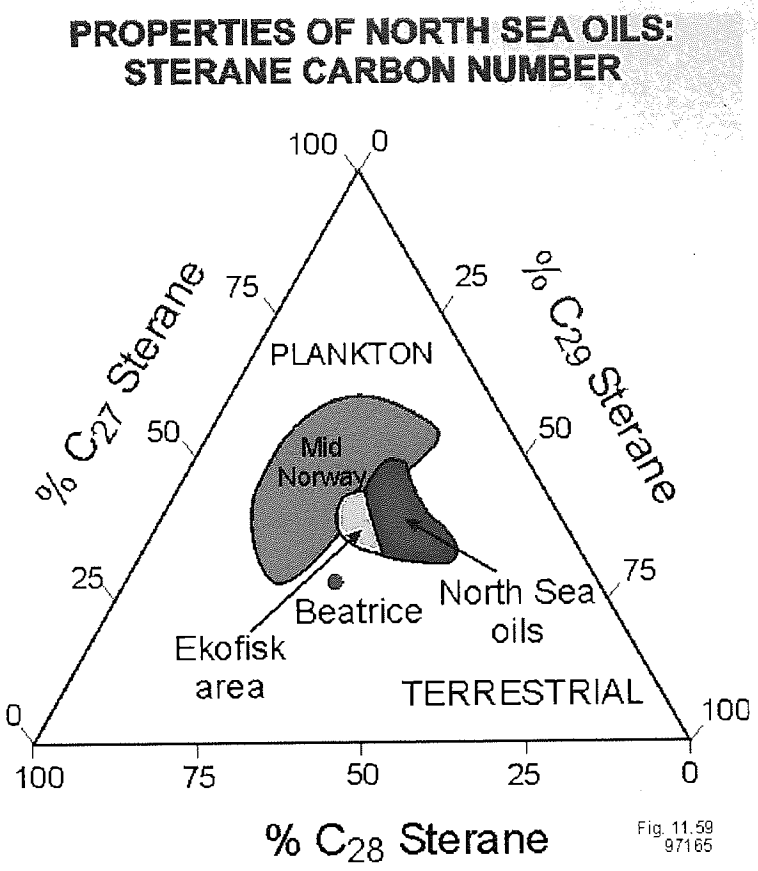
There are a variety of different biomarkers, the most used being the steranes and triterpanes. A discussion of these is beyond the scope of this thesis but the standard reference is the biomarker guide by Peters and Moldowan (1993). A common way to analyse and look at them are key ratios. These ratios are based on prior expert knowledge due to research and previous experience in the field.

These ratios are then used in cross plots or data tables. One example is the use of sterane ratios to reveal differences in secondary migration (Seifert and Moldowan, 1981) as shown in Figure 2.4b. Another example is the usage of key variables in a ternary plot, such as C_{27}, C_{28}, C_{29} which can be used to resolve terrestrial, lacustrine and marine sources (Huang and Meinschein, 1979) as demonstrated in Figure 2.4a.

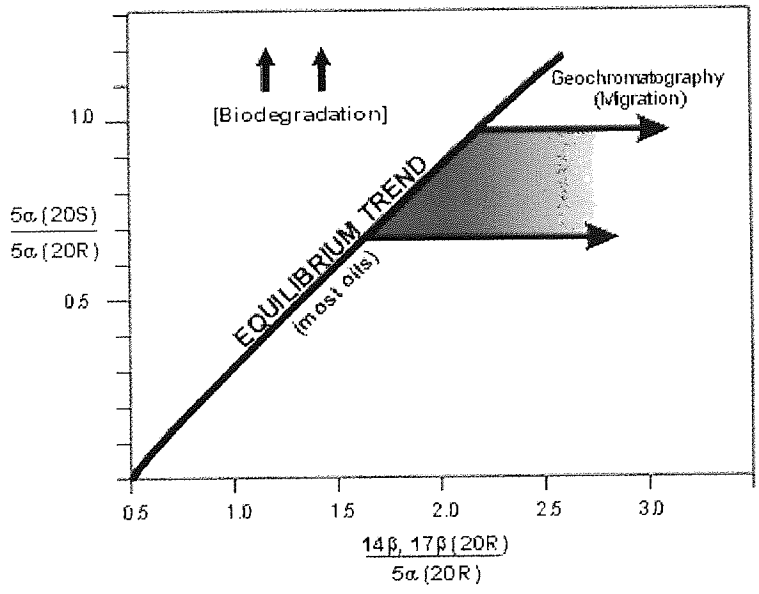
Most of these ratios, cross plots and ternary plots are based on empirical evidence and practical experience. Their use has proven to be successful and the impracticality of doing cross plots for all the available variables tempts the practitioner to neglect most of the other available variables. For example even if one only measures 70 variables one ends up with (70x69)/2=2450 possible bi-plots (cross plots). Another problem with bi-plots is the constraint to only two, or in the case of ratios, 4 variables. Complex patterns might stretch across multiple variables and it might not be possible to identify these when just using bi-plots. This motivates most of the research in this thesis since the usage of the latest multivariate methodologies in data exploration and visualisation might provide benefits for practitioners in geochemistry who wish to explore their data more thoroughly.

2.3 Gas Chromatography-Mass Spectrometry (GC-MS)

GC-MS is one of the most powerful and widespread analytical techniques available to identify biomarker components in hydrocarbon samples. The method combines the features of gas-liquid chromatography and mass spectrometry to extract and identify chemical components within a test sample. The GC-MS procedure is separated into two processes as illustrated in Figure 2.5. In the first process, the gas chromatograph, the tested substance is mixed into a solubilising phase and then travels through the capillary column under constant heating. One has to note that the set up, i.e. column dimensions (length, diameter, film thickness), will influence the end result as well as the solubiliser properties (e.g. 5% phenyl polysiloxane). While travelling through the column the differences in solubility and diffusivity of different molecules in the mixture will separate the molecules over time. This time difference is measured when the different molecules leave or elute from the gas chromatograph and is called the retention time. In the second process a mass spectrometer captures, ionises, accelerates, deflects, and detects the ionised molecules once they elute from the gas chromatograph. This is done by breaking each molecule into ionised fragments and



(a) Ternary plot



(b) Cross plot

Figure 2.4: In geochemistry cross and ternary plots are common to interpret the geochemical compositions of samples and to help with the determination of their origin and possible alteration processes. a) Ternary plot of sterane composition to resolve terrestrial, lacustrine and marine sources. b) Cross plot of sterane ratios to reveal differences in secondary migration. *With permission of IGI Ltd.*

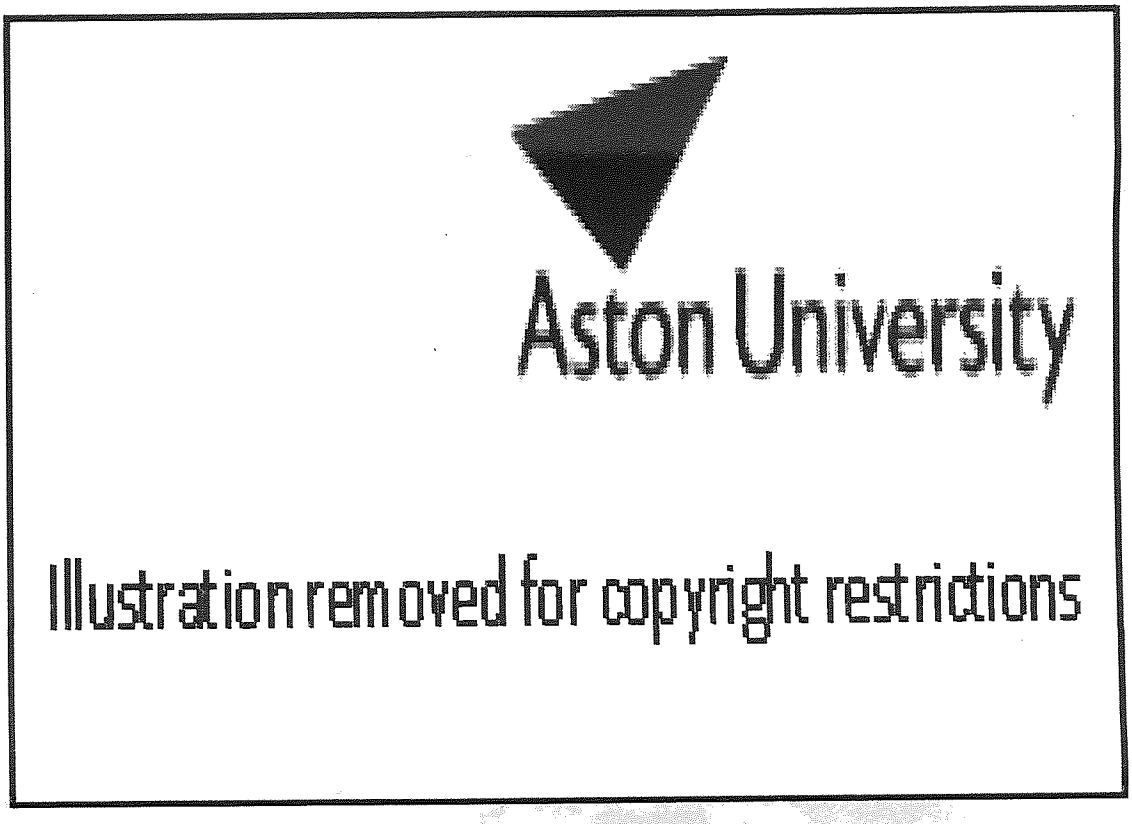


Figure 2.5: Schematic of the GC-MS: The GC-MS works by injecting the sample into a carrier phase (usually an inert gas) which splits the molecules over time by letting them travel through a column and by heating them up and thus eluting them at different points in time. Attached to the end of the GC is a mass spectrometer which is used to detect the different molecules. *Source: Wikipedia*

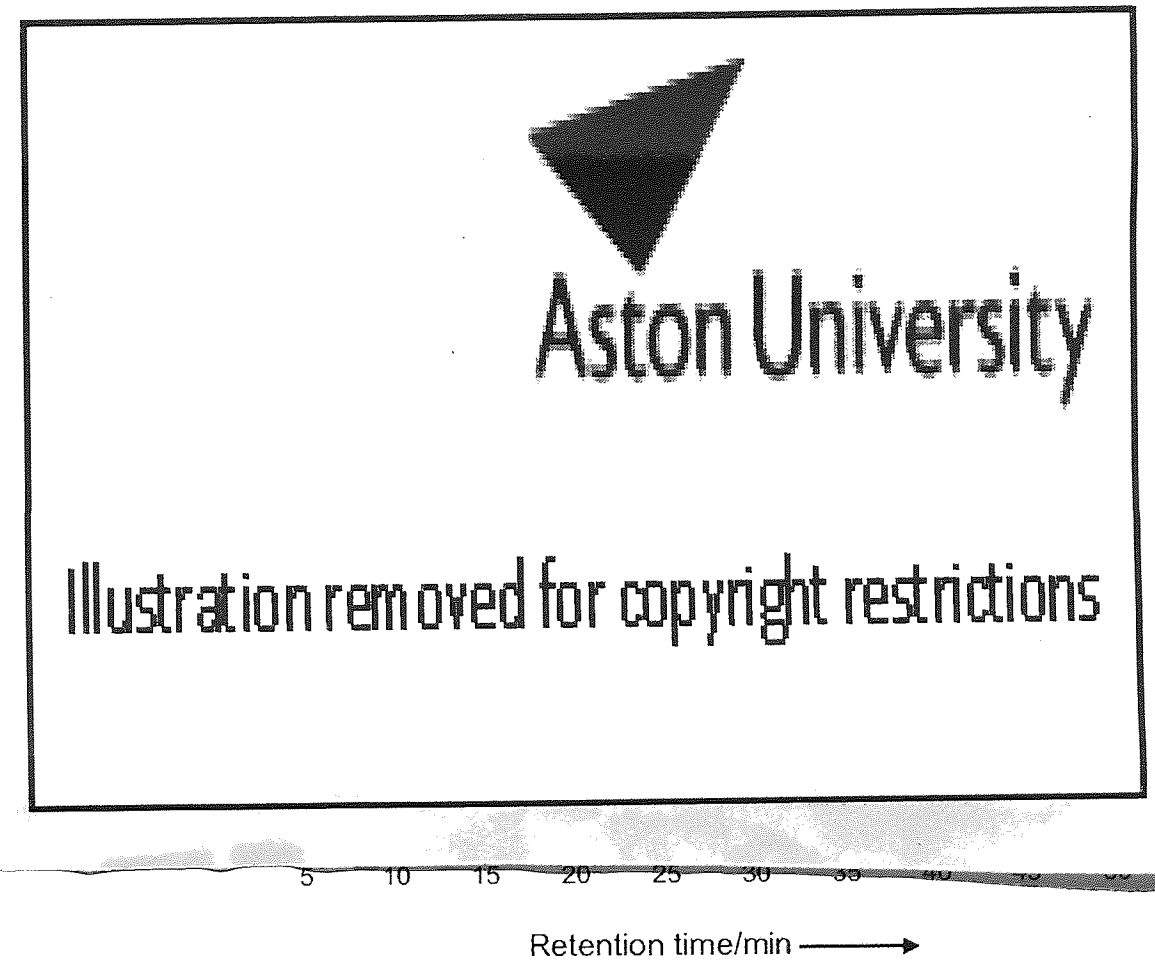


Figure 2.6: Examples of typical GC-MS charts running over the retention time of the analysis. The different spikes mark the detection of chemical molecules which have been picked up with different levels of intensity. *Source: http://geology.gsapubs.org/content/37/10/875/F2.large.jpg*

detecting these fragments according to their mass:charge (m/z) ratio. The result of this detector is amplified and fed either to a chart recorder or, more commonly, to a computer for storage. A sample of this chart or chromatogram can be seen in Figure 2.6.

Peak identification and quantification: Given identical GC conditions (i.e. column length, stationary-phase type and thickness, carrier gas type and flow rate, oven temperature programme) and using a standard oil sample one can tentatively identify the components in a chromatogram using standard oil samples analysed under identical conditions. There are ways to automate this process (Christensen *et al.*, 2004) however processes like biodegradation may complicate the analysis.

Once the components in the samples are matched with the peaks one can try to quantify the relative amount of these components in the given sample. This can be done in two ways: either by measuring the height of a peak or the area underneath it. In general **peak area measurement**, as shown in Figure 2.8, provides the most accurate results if all components are fully resolved. **Peak height measurement** may be preferred when resolution is relatively poor or when only a chromatogram is available. In the case of partially resolved peaks IGI Ltd. takes the height of each peak as the distance between the apex of the peak and a line drawn half-way between the baseline on the fully-resolved side of the peak and the valley between the two partially resolved peaks as demonstrated in Figure 2.7. In the case where absolute concentrations are required it is necessary to add an internal standard to the sample; i.e. a known amount of a compound that does not occur naturally in the sample and which is fully resolved from other components in the chromatogram. However this has to be done before one starts the GC-MS. Another issue is the baseline of a chromatogram which is usually not flat throughout. This is illustrated in Figure 2.9. (A) First there is a gradual then rapid increase in the baseline through column bleed of the liquid stationary phase which gives a background signal. (B-C) The amount of bleed increases with increasing temperature until maximum temperature programme is reached. (D) Over the time of the programme the baseline stays stable, (E) it suddenly drops until when the oven rapidly cools down at end of the temperature programme.

2.4 Chemometrics in Geochemistry

Analysis of geochemical data is an important part of the oil exploration process. The science of extracting information from chemical systems by data-driven means is called chemometrics. Chemometrics primarily involves the use of multivariate statistical methods for the analysis of analytical chemistry data (Brereton, 2007). Multivariate statistical methodologies can help to deal with extensive amounts of compound-specific data and might reveal interesting patterns to help to spot anomalies which might be overseen if one relies only on key ratios and known cross plots. With regards to geochemistry these methods are used ex-

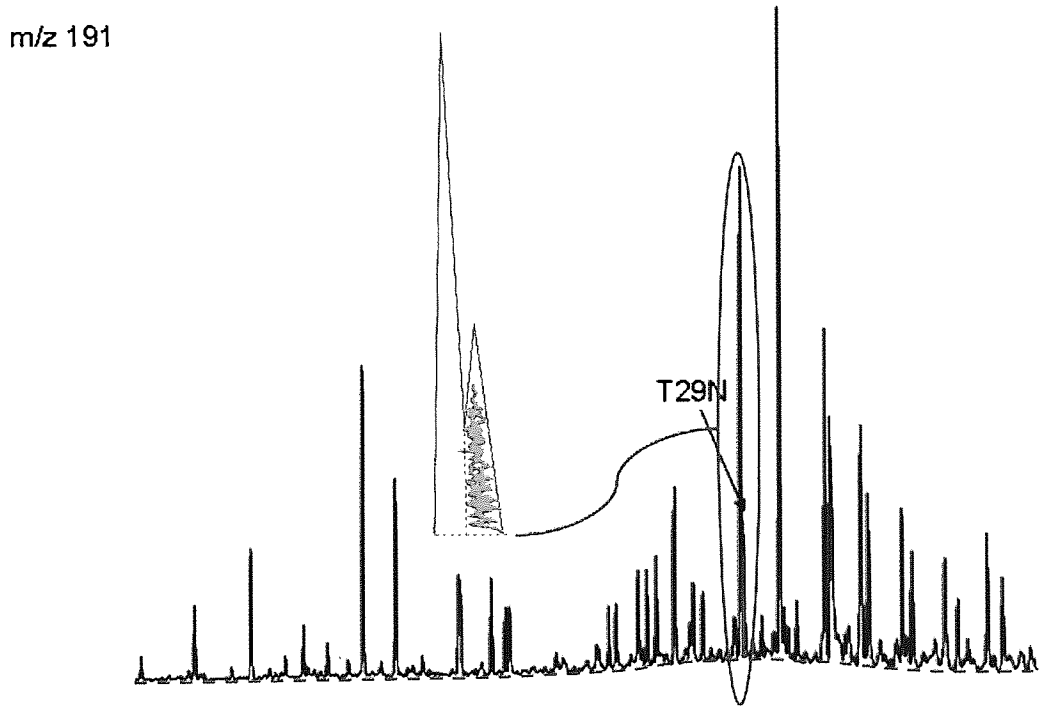


Figure 2.7: Measuring partially resolved peaks is done by drawing a baseline between the baseline on the fully-resolved side of the peak and the valley of the partially resolved peak. The distance is then measured from the apex of the peak to the red dotted baseline. *With permission of IGI Ltd.*

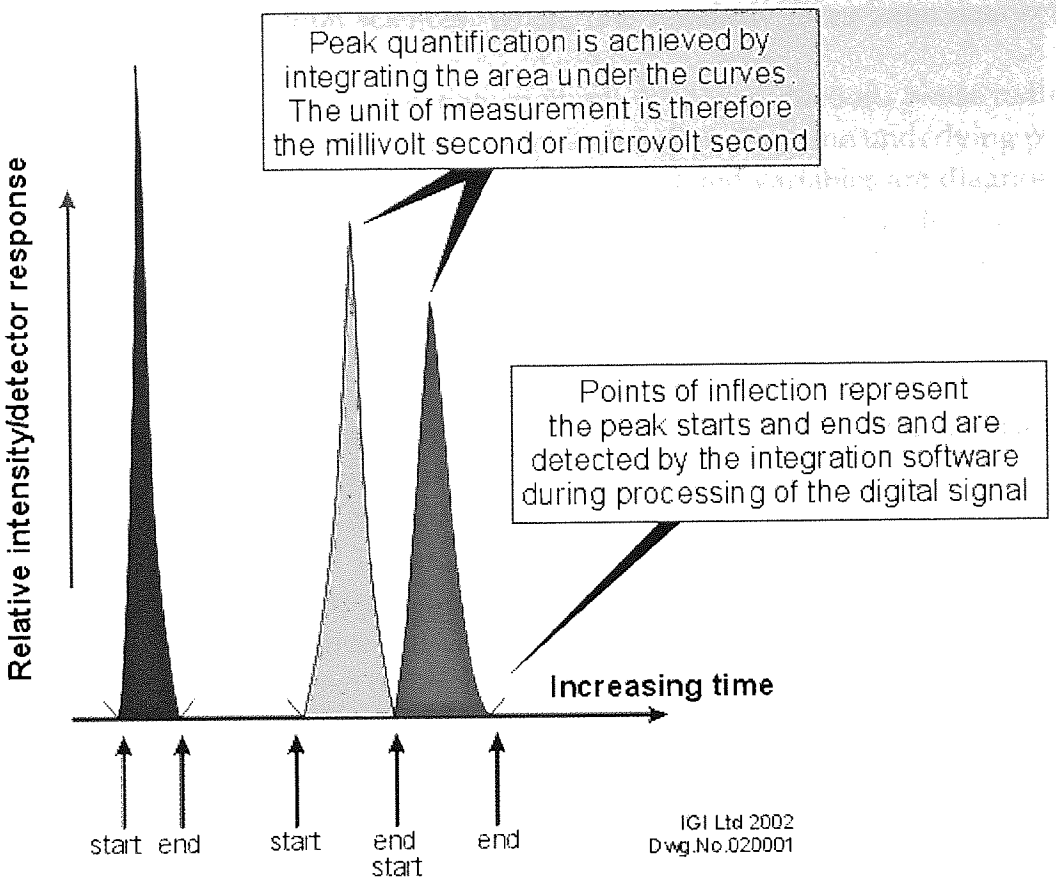


Figure 2.8: Schematic of peak measurements. *With permission of IGI Ltd.*

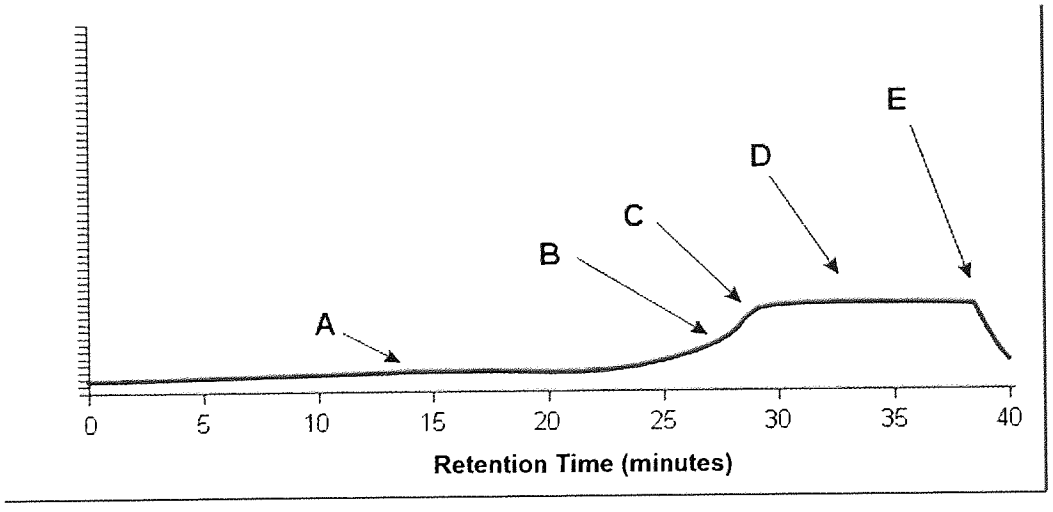


Figure 2.9: Baseline signature of a blank sample run: (A) Gradual then rapid increase in the baseline through column bleed. (B-C) Increase of bleed with increasing temperature until maximum temperature programme is reached. (D) The baseline stays stable. (E) Sudden drop of baseline when the oven rapidly cools down at end of the temperature programme. *With permission of IGI Ltd.*

tensively in environmental sciences, where one regularly faces large data sets, Kettaneh *et al.* (2005).

Another advantage of multivariate methods is noise reduction. Noise reduction is obtained when more than one variable describes the same underlying process (i.e. interrelated variables). Examples of correlated variables are diagnostic ratios like heptadecane/pristane and octadecane/phytane, which describe the same process, namely biodegradation. Likewise if several variables relate to the same process (e.g., thermal maturity, depositional environment and in-reservoir degradation) the combination of them will be less affected by noise, so long as the noise on each is uncorrelated.

One of the main tools used for data visualisation is principal component analysis (PCA) (Jolliffe, 1986). PCA is based on multivariate theory (Kvalheim and Karstang, 1987; Kvalheim, 1987a) and is very popular within the chemometrics community (Kvalheim and Telnaes, 1986a; Kvalheim and Telnaes, 1986b). In general PCA has three main applications. The first is the usage of PCA for the preprocessing of data to reduce the dimensionality and consecutively analyse this reduced data set. In a recent example PCA was used as a pre-processing step before clustering wells and springs for groundwater. This helped to showed similar geographical trends in the trace element chemistry of the wells (Farnham *et al.*, 2000).

The second is the usage of PCA to identify trends and clusters in the correlations of the used variables. An example is the work of Mead *et al.* (2005) who tried to assess the sources of organic matter in sediments and soils of sub-tropical wetland and estuarine systems, in Florida Coastal Everglades. They used PCA loadings to obtain a better resolution of input changes regarding the organic matter along the landscape.

The third is the usage of PCA as a visualisation utility where cross plots between the different principal components are used to find patterns in the distribution of the samples. An example is the chemotaxonomic classification of fossil leaves by Lockheart *et al.* (2000) where PCA was used to emphasise the differences between genera and individual specimens. In another example Walker *et al.* (2005) used PCA to differentiate between sources of polycyclic aromatic hydrocarbons when investigating the contamination of coastal sediments in the Elizabeth River, VA, USA.

These categories are not mutually exclusive. In a study about sources and distribution of organic mater, in sediments of the Atchafalaya river in the northern gulf of Mexico, the loading plots of PCA were used to identify peculiar clusters of biomarkers. Together with the score plots this helped to identify differences and trends in the samples (Gordon and Goñi, 2003). In a source rock study Odden and Kvalheim (2000) used PCA as pre-processing step to remove non-discriminating variables. They then used score plots to identify and remove outliers in their dataset. The loading plots then were used to identify the hydrocarbon components which were most robust and significant when separating the two source rocks.

There is also research on the handling of missing data which can generally be split between the treatment of genuinely missing unknown data and the treatment of "zeros" in compositional data where concentration of measured quantities could not be detected because they were below the detection limit of the analytical machines (i.e. censored data).

Work on the latter problem has been performed by Farnham *et al.* (2002) who looked at the treatment of non-detects in ground water data and imputing zero, the detection limit or half the detection limit for missing values. Other approaches dealing with rounded zeros include the use of Markov Chain Monte Carlo Methods (MCMC) (Thió-Henestrosa and Martın-Fernández, 2003). To prevent the imputation of negative values Palarea-Albaladejo and Martín-Fernández (2008) used a log transform model and utilised the Expectation Maximisation (EM) algorithm to estimate the undetected values.

To handle genuinely missing data a popular method of choice is the the nonlinear iterative partial least squares (NIPALS) algorithm. This algorithm was first extended by Christoffersson (1970) to find the first two principal components in PCA with missing data and the model was later generalised for arbitrary numbers of components (Grung and Manne, 1998). An example for its use is the work by Dray *et al.* (2003) whoo utilised the NIPALS algorithm when mapping data in geographic information systems. Other recent work by Dickson and Giblin (2007) used a regression approach utilising self organising maps (SOM) and a regularised EM algorithm to estimate missing trace elements in ground water data.

In petroleum geochemistry, multivariate statistical methods are used more rarely. Commonly oil and source rock correlation has been undertaken using various comparisons of bulk, molecular and isotopic properties. Examples of the approaches used by a number of laboratories on a common set of data are reported in Magoon and Claypool (1985). These approaches range from semiquantitative comparisons of visual similarity (e.g. ++, +, +/-, -, –) and correlations of molecular ratios together with matrices of correlation coefficients and derivative dendrograms. The first use of multivariate statistical methods can be dated back to the late 1980s when, for example, PCA was used by Telnaes and Dahl (1986), who anchored their statistical analysis to reality by cross-plotting the extracted PCA scores against molecular ratios dominantly controlled by a single process (i.e. $C_{20}/(C_{20+C27})$ Tri-Aromatic Sterane ratio as a measure of maturity). Further PCA has been used extensively in different biomarker studies (Fernandes *et al.*, 1999; Napitupulu *et al.*, 2000; Niggemann and Schubert, 2006) where the loadings were used to identify key biomarkers responsible for correlation or separation. In one study Kvalheim (1987b) used PCA to identify source specific parameters in a input matrix consisting of C_6 and C_7 saturates from a variety of different sources including West Texas, New Mexico, Colorado, Montana, South Dakota, the Texas Gulf Coast and the Los Angeles Basin. Other studies reflect a dichotomy of approaches based on using all available data i.e. compounds or ratios thereof (Justwan *et al.*, 2006) versus using selected data focussing on one or

a few geological processes such as organo-facies, maturation, fractionation and bacterial degradation (Zumberge, 1987; Peters *et al.*, 2007). Combined approaches include the use of different sets of variables to create multiple PCA models which helped Pasadakis *et al.* (2004) to study the Williston Basin to characterise different petroleum families.

A word of warning is needed. There are many problems like non-normality and noisy data which need special pre-processing treatments (Kvalheim *et al.*, 1994). Other problems especially with geochemical data are small sample sizes. A good paper discussing these problems has been published by Reimann *et al.* (2002). In the applied case study in chapter 7 some aspect of it will be discussed in more detail.

2.5 Summary

Looking at the methodologies employed in geochemistry for petroleum exploration and at the academic literature we can identify four major issues which will be addressed in this thesis to further the knowledge in this area:

- There is only a limited use of multivariate statistical methods in the geochemical community and the utility of other methods is still open to be explored.
- Until now no work has been done in the geochemical community with regards to probabilistic or non-linear visualisation methods.
- Until now no work has been done to assess the performance of different methods to treat missing data in petroleum geochemistry.
- There will be a problem when GC-MS data from different labs need to be analysed together. The peak heights and thus measurements are dependent on various factors like the amount of material injected into the specifications of the used machine. This is especially problematic where no measurements for a reference sample are available.

3 Toy data sets used in this report

CONTENTS

3.1 S-shaped data 47
3.2 Swiss-roll data 47
3.3 Multi phase oil flow data 47
3.4 20 and 60 dimensional toy data 48

In this thesis several data sets are used to for the demonstration or test of different algorithms. In this chapter these data sets are introduced. The S-shaped and Swiss-roll toy data are used to demonstrate the fitting of the different visualisation methods discussed in chapter 4. The multi phase oil flow data as well as the 20D and 60D toy data sets are used to benchmark different extensions for GTM in chapter 5 and to benchmark different imputation methods respectively in chapter 6. The real data were supplied by IGI and are based on samples from oils in the Barents Sea, North Sea and from a region in Africa. These data sets will not be introduced in this chapter. They will be introduced and discussed in chapter 7.

3.1 S-shaped data

This three dimensional dataset was created using two semi-circles in two dimensions and a uniformly random distribution of points along the third dimension. The shape of the data is S-like and is divided into three equally big classes over the length of the manifold. This data set is used to illustrate how the different models project higher dimensional data onto a lower dimensional manifold. Figure 3.1 shows the structure as well as a projection obtained by using the first two principal components of PCA.

3.2 Swiss-roll data

This three dimensional data set is based on the Swiss-roll. It is highly non-linear due to the inwards curving structure. It can be constructed by drawing a circle with continuously growing radius over time which gives it a spiral like structure. As with the S-shaped data the third dimension is given by a uniform random distribution. It is a data set used in the machine learning community to illustrate the advantages of local embedding techniques (Roweis and Saul, 2000; Tenenbaum *et al.*, 2000; Harmeling, 2007; Belkin and Niyogi, 2003) as discussed in chapter 4. Figure 3.2 illustrates the structure and how PCA fits this data. As expected PCA fails to pick up on the non-linear structure due to the inherent limitations of the algorithm based on the restriction to a linear formulation of the mapping function.

3.3 Multi phase oil flow data

This is a twelve dimensional data set containing data from the oil flow in a pipeline (Bishop and James, 1993). The data can be separated into three known classes corresponding to the phase of the oil, water and gas mixture as demonstrated in Figure 3.3(a). The data are collected from a non-invasive monitoring system based on gamma rays. The data set arises from a set of three horizontal and three vertical beam-lines along which gamma rays, at two different energies, are passed

through the pipeline. By measuring the degree of attenuation of the gamma rays, the fractional path through the oil, water and gas mixture can be determined. The data set was synthetically sampled by simulating the physical processes in the pipe, including the presence of noise determined by photon statistics. The data set is widely used in machine learning to demonstrate the capabilities of clustering and visualisation algorithms (Bishop *et al.*, 1996; Lawrence, 2005; Haese and Goodhill, 2001; Blanchard *et al.*, 2006). It is therefore well understood and a good benchmark for the purpose of this thesis. Further if one looks at the structure of the covariance in Figure 3.3(b) one can see a chess board pattern. This comes as no surprise since one would expect the three beams in one setting, as well as the two energies to be correlated. Therefore one might be able to reorder the covariance into a block structure, which will be explored in chapter 5. Further the structure seems to be non-linear as the first three principal components of PCA can not fully capture it (Bishop *et al.*, 1996; Lawrence, 2005) as shown in Figure 3.4.

3.4 20 and 60 dimensional toy data

To simulate a high dimensional data set which has non-linear relations between the variables and a pre-known block structure in the covariance matrix two toy data sets were created. One is 20 and the other 60 dimensional. The data were sampled from a GTM with an 8×8 grid in the latent space. This was done by randomly choosing a Gaussian for each data point and sampling it individually from this Gaussian. The grid was projected into a higher dimensional space using a radial basis function (RBF) network with 4 hidden units (2×2 grid). The weights were randomly sampled from a normal distribution with zero mean and unit variance. Since the RBF was chosen with random weights the restriction to a 2×2 RBF ensured a non-linear but smooth mapping. The settings are summarised in Table 3.1. The GTM used to generate the data had a block diagonal covariance matrix and experiments were conducted with a relative high level of variance and correlation as can be seen from Figure 3.5. In both cases several GTMs were sampled until one fulfilled the criteria of generating data where the underlying structure could not be picked up with PCA. To make the data visually interpretable the 8×8 grid was split into 4 classes with the 16 Gaussians in one corner of the grid being defined as one class. This makes it possible to assess if the visualisation or classification methods manage to capture the large scale structure of the data. As intended this is not the case with PCA as can be seen in figure 3.6 and 3.7.

Parameters	Value
Base domain	8x8 Gaussian grid
Number of samples	100
Dimension of base data	2
Projection function	2x2 RBF
Dimension of the projected data	20 and 60

Table 3.1: Summary of the specifications for the GTM respectively generating the 20D and 60D toy data.

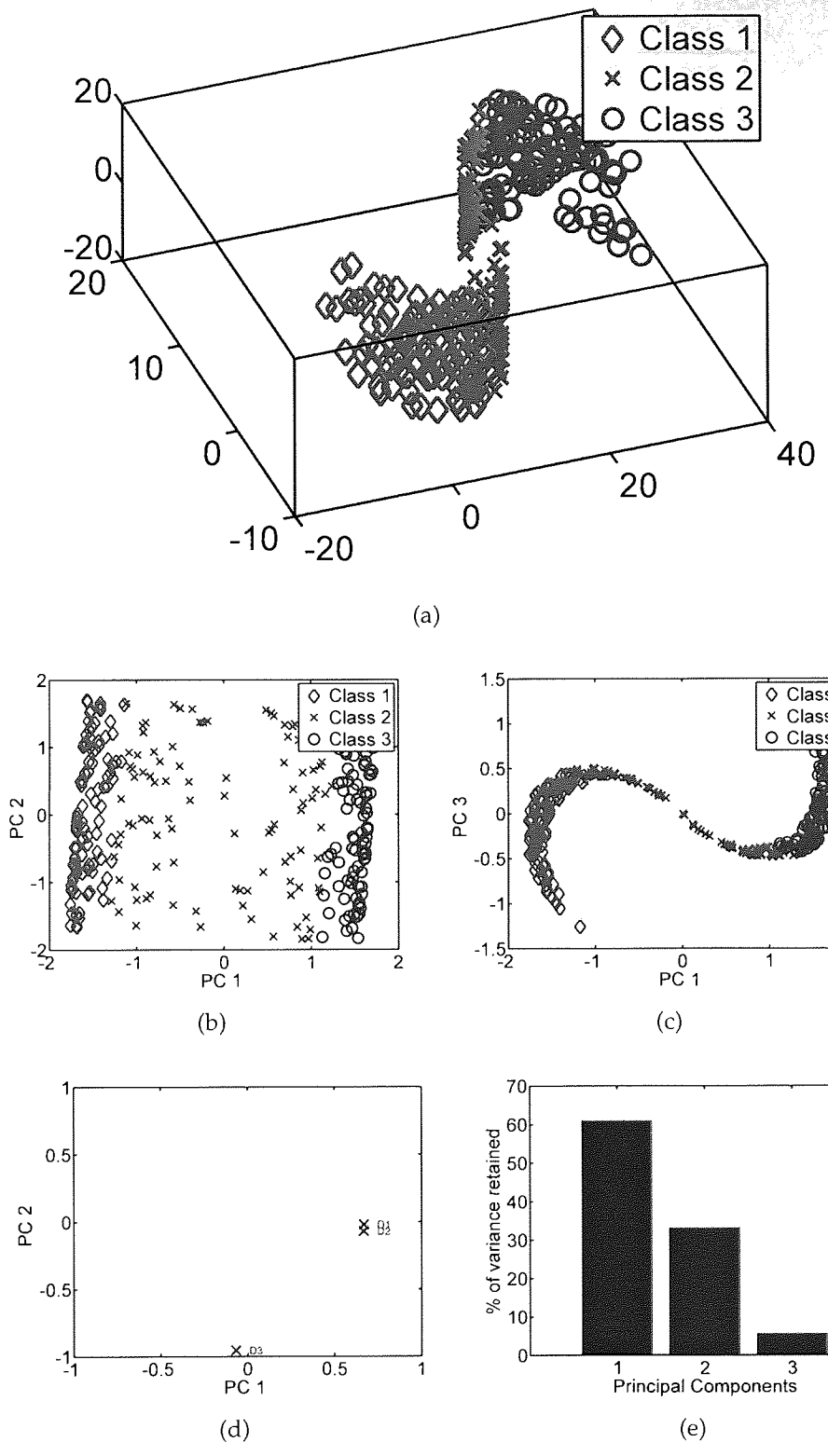
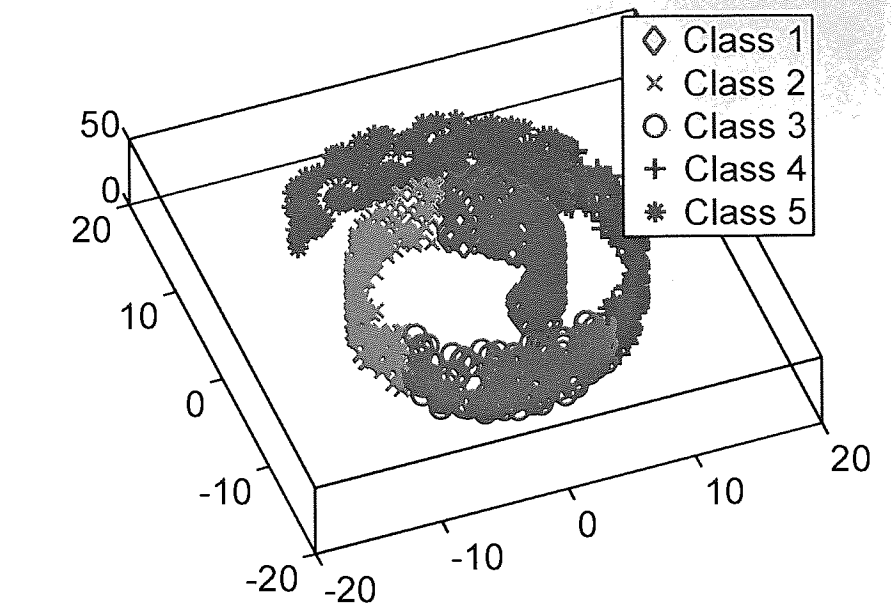
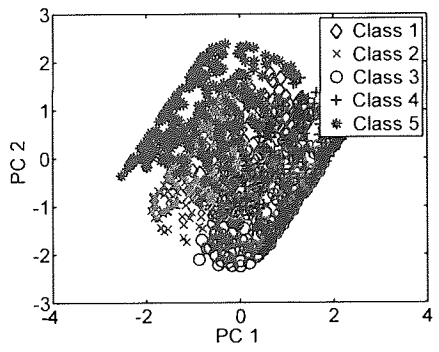


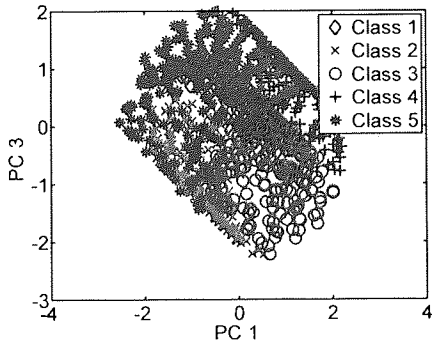
Figure 3.1: (a) The S-shaped data in 3D. (b) The PCA projection or scores plot for PC1 vs PC2. (c) The PCA projection or scores plot for PC1 vs PC3. (d) The loadings plots shows the contribution of each dimensions (labelled D1, D2, D3) to the first two principal components. A very high or low value on the axes denoted by the principal component translates to a high or low contribution. (e) Percentage of the original variance in the data explained by principal components.



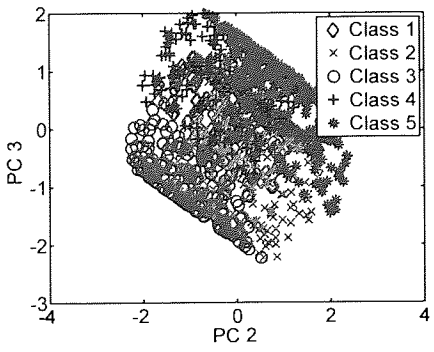
(a)



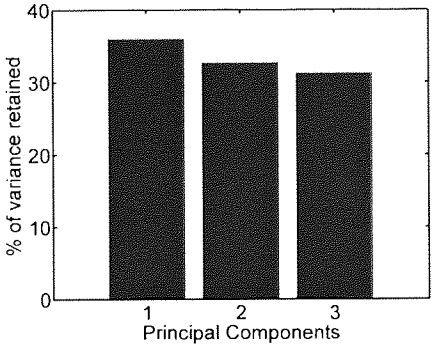
(b)



(c)

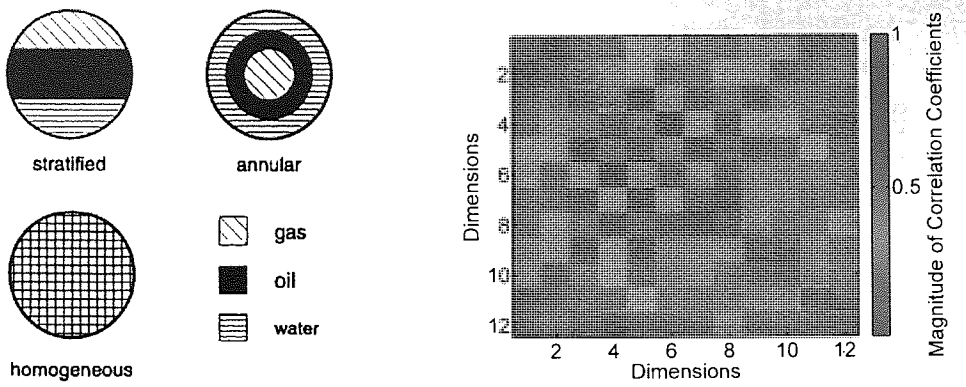


(d)



(e)

Figure 3.2: (a) The Swiss-roll data in 3D. (b) The PCA projection or scores plot for PC1 vs PC2. (c) The PCA projection or scores plot for PC1 vs PC3. (d) The PCA projection or scores plot for PC2 vs PC3. (e) Percentage of the original variance in the data explained by principal components.



(a) The possible oil phases (b) Covariance Structure of oil data

Figure 3.3: Description of the possible phases for the multi phase oil data by Bishop as well as a plot of the correlation coefficients to visualise the covariance structure in the data.

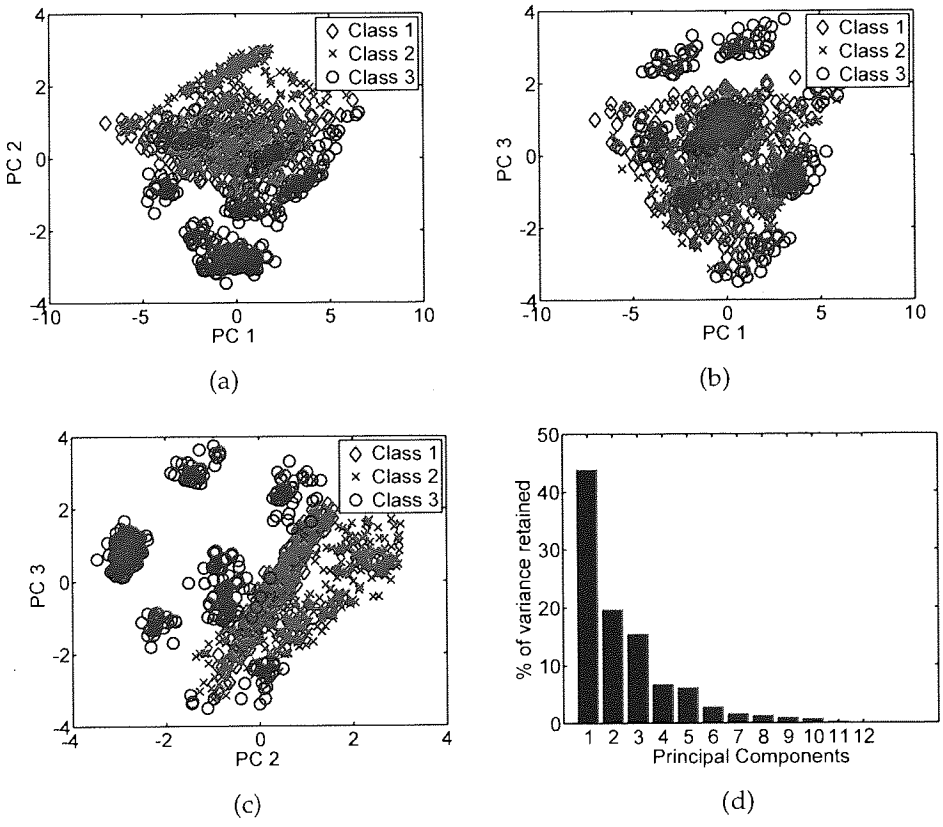
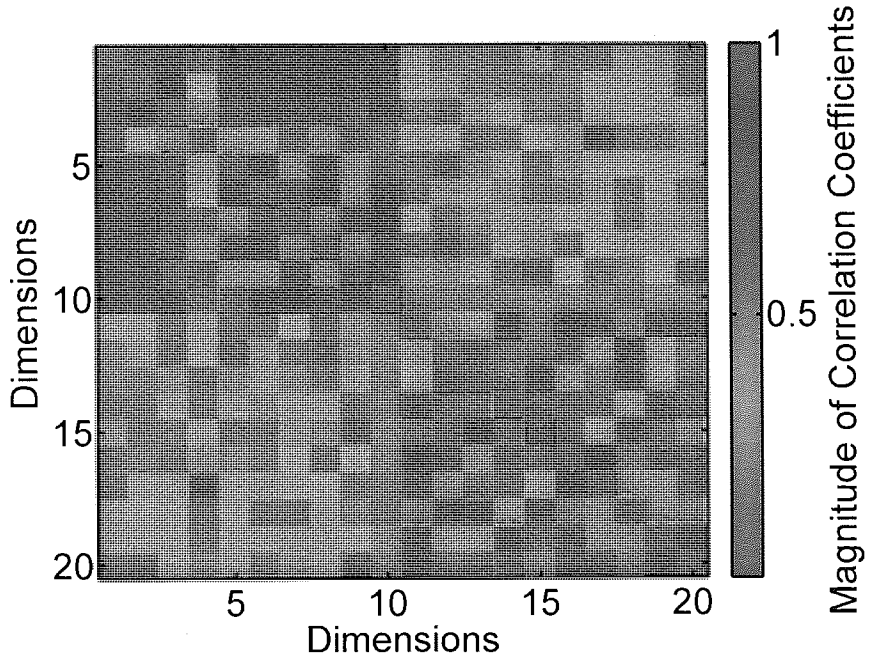
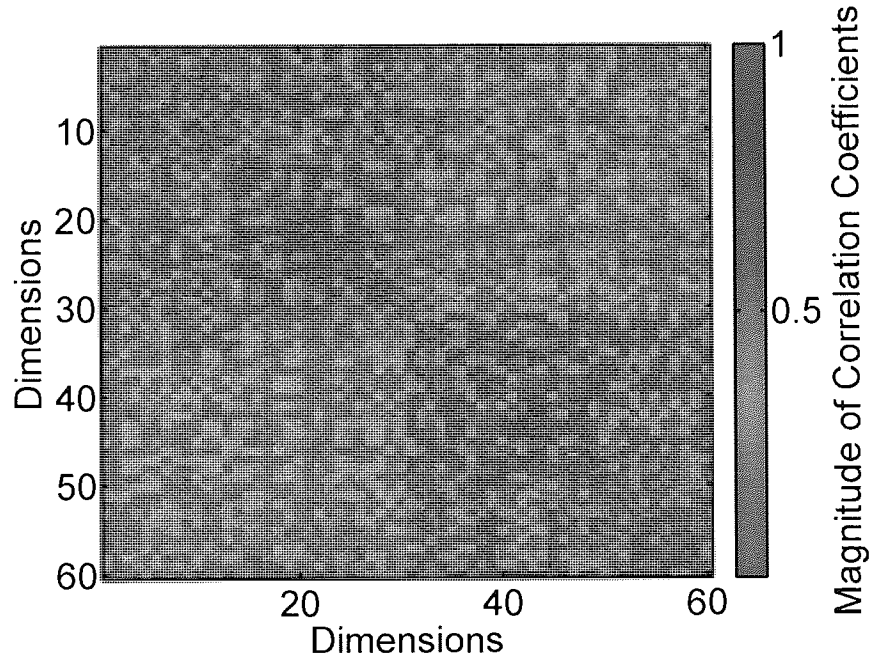


Figure 3.4: The plots show the PCA analysis of the multi phase oil flow data. (a) The PCA projection or scores plot for PC1 vs PC2. (b) The PCA projection or scores plot for PC1 vs PC3. (c) The PCA projection or scores plot for PC2 vs PC3. (d) Percentage of the original variance in the data explained by principal components.



(a)



(b)

Figure 3.5: Representation of the covariance structure by plotting the correlation coefficients. (a) The 20D and (b) the 60D data set created by GTMs according to the specifications in Table 3.1.

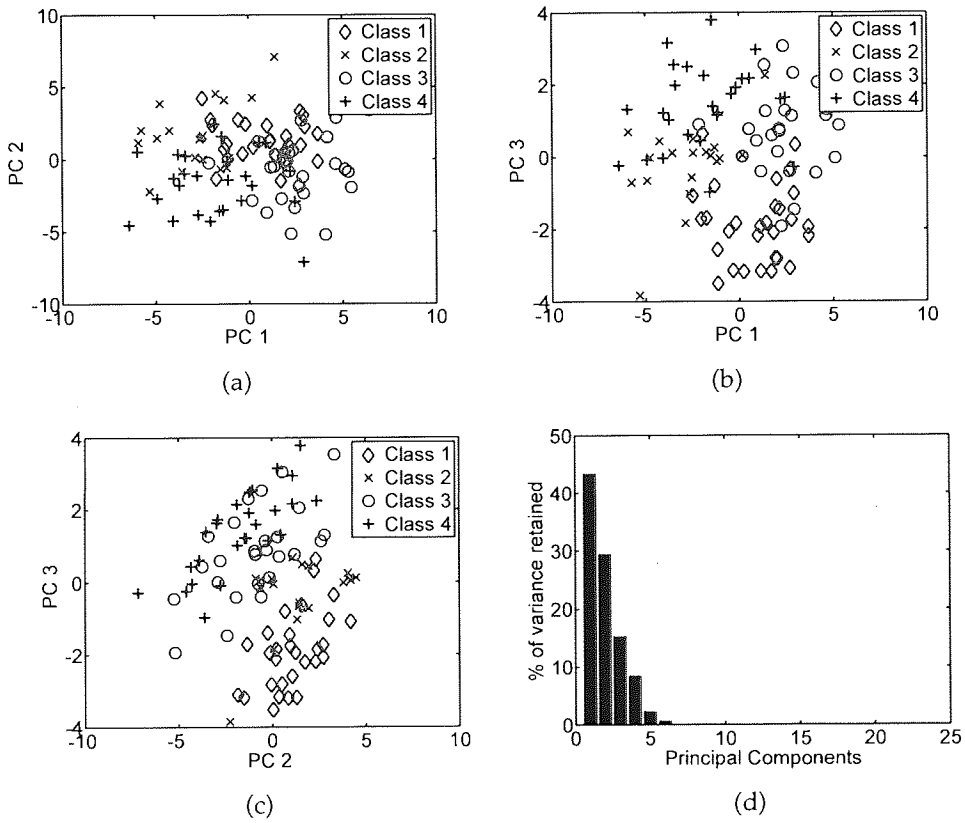


Figure 3.6: The plots show the PCA analysis of 20D toy data set. (a) The PCA projection or scores plot for the first two principal components. (b) The PCA projection or scores plot for PC1 vs PC3. (c) The PCA projection or scores plot for PC2 vs PC3. (d) Percentage of the original variance in the data explained by principal components.

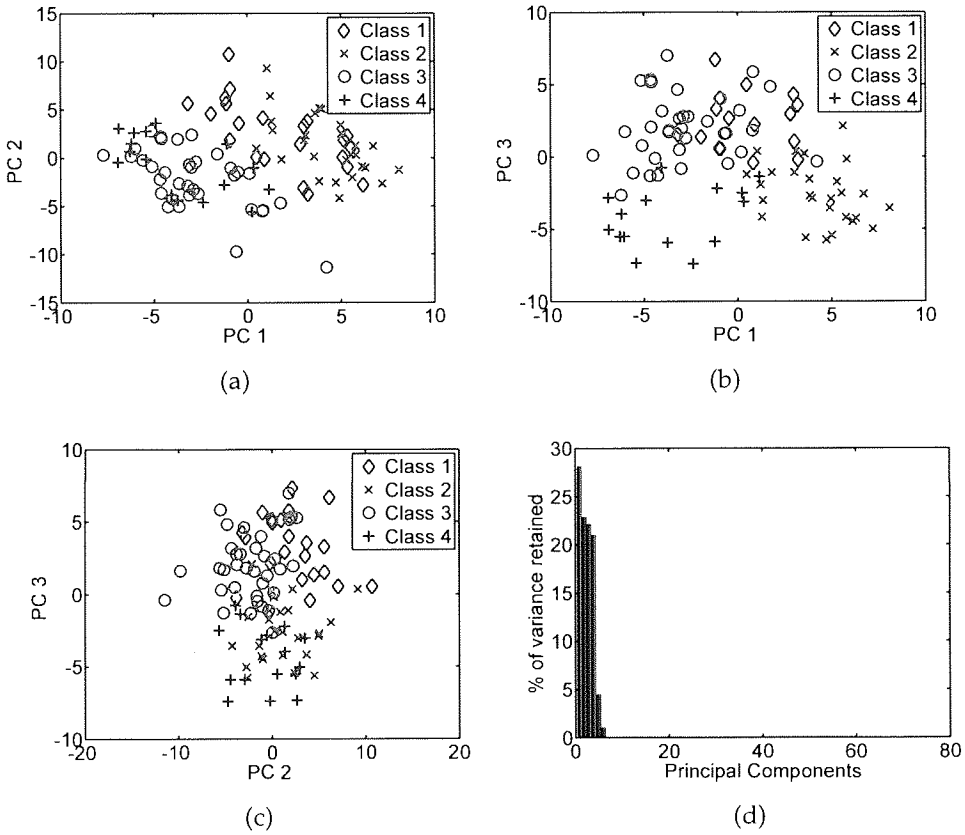


Figure 3.7: The plots show the PCA analysis of 60D toy data set. (a) The PCA projection or scores plot for the first two principal components. (b) The PCA projection or scores plot for PC1 vs PC3. (c) The PCA projection or scores plot for PC2 vs PC3. (d) Percentage of the original variance in the data explained by principal components.

4 Data Modelling and Exploration

CONTENTS

4.1 EM Algorithm 58
4.2 Mixture Models 59
4.3 Gaussian Mixture Models 60
4.4 Generative Topographic Mapping 62
4.4.1 Data Visualisation using GTM 65
4.4.2 Initialising GTM 68
4.5 Other visualisation algorithms 71
4.5.1 PCA 71
4.5.2 Probabilistic PCA 71
4.5.3 Kernel PCA 71
4.5.4 Gaussian Process Latent Variable Model (GPLVM) 71
4.5.5 MDS 72
4.5.6 Neuroscale 72
4.5.7 Isomap 72
4.6 Summary 72

Visualisation of high-dimensional data requires a method to map, or project, the high-dimensional data onto a low-dimensional representation while preserving as much information about the structure in the high dimensional space as is possible. This low-dimensional representation is usually two-dimensional to be shown on screen or paper and will be referred to as the *visualisation space.* Employing a two-dimensional visualisation space allows the human analyst to explore the data and discern structure easily and naturally. There are many possible ways to obtain such a low-dimensional representation. Context will often guide the approach, together with the manner in which the visualisation space representation will be employed. In general the methods to obtain this lower-dimensional representation are referred to as *latent variable models.* Some methods, such as Principal Component Analysis (PCA) and Factor Analysis (Chatfield and Collins, 1980), linearly transform the data space and project the data onto the visualisation space while retaining the maximum information [1]. Other methods, like Kohonen, or Self Organising, Maps (Kohonen, 1995) and the Generative Topographic Mapping (GTM) (Bishop *et al.*, 1998; Bishop *et al.*, 1996), try to capture the topology[2] of the data. Geometry-preserving methods like Multi-Dimensional Scaling (MDS) try to find a representation in visualisation space which preserves the geometric distances between the data points. With the Neuroscale algorithm the MDS approach has been extend to preserve the typological order of the data as well (Lowe and Tipping, 1996). Other modifications of the MDS algorithm like Locally Linear Embedding and the Isomap algorithm (Roweis and Saul, 2000; Tenenbaum *et al.*, 2000; Saul and Roweis, 2003) try not to preserve the euclidean geometric distance between data points but instead try to preserve the distances of a local metric which is based on a connected graph.

This chapter describes the main visualisation and modelling algorithms that underpin our work or that are relevant to enhancements. First the EM algorithm will be introduced in its general form. This is necessary since it is a key element of the GTM algorithm and the extensions proposed to it in this thesis. Afterwards there is a short introduction to mixture models and Gaussian mixture models, which will aid the understanding of GTM. The following section focuses on GTM, which can be described as a constrained Gaussian mixture model.

The next section will only give a very brief overview of additional visualisation algorithms and a more technical discussion can be found in the Appendix B. First there is an explanation of PCA, which is a widely employed method and generally provides a benchmark in this thesis. This is followed by an introduction to PPCA, the probabilistic formulation of PCA, and which is important to explain the theory behind the two further models: Kernel PCA (KPCA) and the Gaussian Process Latent Variable Model (GPLVM). To complete the review a short introduction to MDS and two methods, Neuroscale and Isomap, which are related to MDS is given. The S-data and Swiss-roll data set are used throughout the chapter

[1] Strictly the 1st principal component explains the maximum variance, which in a Gaussian setting equates to information in the Fisher entropic sense.

[2] A topological mapping is one that seeks to preserve local neighbour relations; two points that are neighbours in the data space should also be neighbours in the visualisation space.

to illustrate the practical application of the methods described.

4.1 EM Algorithm

The EM algorithm (Dempster *et al.*, 1977) is a general method to cope with incomplete data or missing values when finding the maximum likelihood estimates of the parameters of an underlying distribution. This can be exploited when one has to optimise a likelihood function that is analytically intractable but can be simplified through the assumption of additional hidden or missing parameters.

In the general case we observe the data $\mathbf{X}$ generated by some distribution. We call $\mathbf{X}$ the *incomplete data* and we assume that a complete data set $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})$ exists with a joint density function, which depends on the parameter vector θ:

$$p(\mathbf{z}|\theta) = p(\mathbf{x}, \mathbf{y}|\theta) = p(\mathbf{y}|\mathbf{x}, \theta)p(\mathbf{x}|\theta) \; .$$

With this density function we can define a likelihood function

$$l(\theta|\mathbf{Z}) \;\; = \;\; p(\mathbf{X}, \mathbf{Y}|\theta) = p(\mathbf{Y}|\mathbf{X}, \theta)p(\mathbf{X}|\theta) = p(\mathbf{Y}|\mathbf{X}, \theta) \sum_{\mathbf{Y}} p(\mathbf{X}, \mathbf{Y}|\theta)$$

called the *complete-data likelihood*. $\mathbf{Y}$ is unknown, random and generated by an underlying distribution. The EM algorithm finds the expectation for the complete-data likelihood based on the current parameter estimates θ^{i-1}, the known data $\mathbf{X}$ and the unknown data $\mathbf{Y}$. This is done through an iteration in 2 steps: the E-step, which estimates the expectation denoted as $Q(\theta^i, \theta^{i-1})$ for the posterior $p(\mathbf{Y}|\mathbf{X})$ and the M-step, which optimises the parameters of the expectation. This expectation $Q(\theta^i, \theta^{i-1})$ is defined as:

$$Q(\theta^i, \theta^{i-1}) \;\; = \;\; E[\log p(\mathbf{Y}|\theta^i)|\mathbf{X}, \theta^{i-1}] \; .$$

These parameter estimates are used to evaluate the posterior as well as θ and they are are optimised to increase Q. In the M step we determine θ^i by maximising this expectation

$$\theta^i = \arg\max_{\hat{\theta}} Q(\hat{\theta}, \theta^{i-1}) \; ,$$

which is repeated as often as necessary in combination with the E step. Each iteration will increase the expectation as well as the log likelihood until the algorithm converges. However there is no guarantee of obtaining the global maximum likelihood estimate and it is quite common to get stuck in a local maximum of the likelihood function. Hence the initialisation of this method is very important. It will be discussed in detail in the section about the GTM algorithm and one of our novel extensions in chapter 5 is also dealing with this problem.

4.2 Mixture Models

Mixture Models are generally used to model the probability density of data given the assumption that the data is an accumulation of different components, each with its own component density. Alternatively, they are used to approximate more complex densities by using a combination of simpler densities. The assumption is that the density of the data can be approximated by a linear combination of simple component densities $p(\mathbf{y}|\theta_k)$ (Bishop, 1995):

$$p(\mathbf{y}|\theta) = \sum_{k=1}^{K} \alpha_k p(\mathbf{y}|\theta_k) \,, \tag{4.1}$$

with α_k being the mixing coefficients satisfying the conditions:

$$\sum_{k=1}^{K} \alpha_k = 1, \quad 0 \leq \alpha_k \leq 1 \,,$$

which guarantee that $p(\mathbf{y}|\theta)$ is a valid density function. Assuming that all components have the same functional form, the parameters of the mixture model are $\theta = (\theta_1, ..., \theta_K)$ and α_k. Then each component density is specified by the parameter vector θ_k. A mixture model with proper and sufficiently many components is able to represent arbitrarily complex probability density functions when the parameters are selected appropriately. To fit the model to the data one first computes $l(\theta)$, the likelihood of θ for the given data:

$$\prod_{n=1}^{N} p(\mathbf{y}_n|\theta) = \prod_{n=1}^{N} \left\{ \sum_{k=1}^{K} \alpha_k p(\mathbf{y}_n|\theta_k) \right\} \equiv l(\theta) \,.$$

To determine the parameters of a mixture model from a set of data, we minimise the objective function given by the negative log likelihood for the data set:

$$\begin{aligned} -L(\theta) &= -\ln l(\theta) \\ &= -\sum_{n=1}^{N} \ln \left\{ \sum_{K}^{k=1} \alpha_k p(\mathbf{y}_n|\theta_k) \right\} \,, \end{aligned}$$

This can be done with the EM Algorithm where we assume that the observed data $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_N)$ are incomplete. A set of discrete index variables z_{kn} is introduced for each data point $\mathbf{y}_n$. The index variable will be $z_{kn} = 1$ only if the the data point $\mathbf{y}_n$ was generated by the kth component of the mixture model, otherwise $z_{kn} = 0$.

This implies a new negative log likelihood, the *complete likelihood*, which can be written as

$$-L(\theta)_{comp} = -\sum_{n=1}^{N} \ln \prod_{k=1}^{K} \left\{ \alpha_k^{z_{kn}} p(\mathbf{y}_n|\theta_k)^{z_{kn}} \right\} \,, \tag{4.2}$$

$$-L(\theta)_{comp} = -\sum_{n=1}^{N} \sum_{k=1}^{K} z_{kn} \ln \left\{ \alpha_k p(\mathbf{y}_n|\theta_k) \right\} \,, \tag{4.3}$$

because the binary form of z_{kn} implies that for any index k only one will be 1 and all others will be 0. Therefore the sequence of products $\prod$ collapses to only one element and thus we can move the logarithm inside the sum. This simplifies the whole equation because the index variable z_{kn} decouples the single component densities. Having this complete log likelihood and treating z_{kn} as the missing data we can write the EM algorithm as follows:

- E-Step:
Compute the expectation of $-L(\boldsymbol{\theta})_{comp}$ with respect to the variables z_{kn} and fixed posterior $p(\theta_k|\mathbf{y}_n)$ given by

$$\langle -L(\boldsymbol{\theta})_{comp}\rangle_z = -\sum_{n=1}^{N}\sum_{k=1}^{K} p(\theta_k|\mathbf{y}_n)\left\{\ln(\alpha_k) + \ln(p(\mathbf{y}_n|\theta_k))\right\} ,$$

and

$$p(\theta_k|\mathbf{y}_n) = \frac{\alpha_k p(\mathbf{y}_n|\theta_k)}{\sum_{k=1}^{K}\alpha_k^p(\mathbf{y}_n|\theta_k)} . \tag{4.4}$$

This term is called **posterior responsibility** and this value is an estimate on how likely it is that the density with the parameters given by θ_k generated the data point $\mathbf{y}_n$ given all the other possible K parameter vectors.
- M-Step:
Update parameters by minimising the negative log likelihood with respect to the parameters

$$\boldsymbol{\theta}^{new} = \arg_{\boldsymbol{\theta}}\min\langle -L(\boldsymbol{\theta})_{comp}\rangle_z .$$

4.3 Gaussian Mixture Models

Gaussian Mixture Models (GMM) are frequently used for density estimation as well as clustering of data. They are a special case of mixture models with K Gaussian components, where the distribution of each component is defined as

$$p(\mathbf{y}|\theta_k) = p(\mathbf{y}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}(2\pi)^{D/2}}\exp\left\{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{y}-\boldsymbol{\mu}_k))\right\} ,$$

where $\boldsymbol{\Sigma}_k$ is a $D \times D$ symmetric and positive-definite covariance matrix and $\boldsymbol{\mu}_k$ is the mean vector of component k. As in the general mixture model the parameters of the GMM can be determined via maximum likelihood estimate using the EM algorithm.

- E-Step:
The posterior probability $p(\theta_k|\mathbf{y}_n)$ or *posterior responsibilities*, given by equation (4.4), needs to be computed for every component.
- M-Step:

The parameters for the mean vector and the covariance matrix are obtained through the minimisation of the negative log likelihood $-L_{comp}$ with respect to these parameters. For the mean vector we obtain the following equation:

$$\mu_k = \frac{\sum_{n=1}^{N} p(\theta_k|\mathbf{y}_n)\mathbf{y}_n}{\sum_{n=1}^{N} p(\theta_k|\mathbf{y}_n)}, \tag{4.5}$$

i.e. the usual equation for estimating the mean but modified to weight every data point by the responsibility of the Gaussian centre.
The update equation of the covariance matrix depends on the type of covariance structure.
Spherical covariance matrix, thus all variables are independent but share the same level of noise:

$$\mathbf{\Sigma}_k = \begin{bmatrix} \sigma_k^2 & 0 & \dots & 0 \\ 0 & \sigma_k^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_k^2 \end{bmatrix}$$

$$(\sigma_k)^2 = \frac{1}{D}\frac{\sum_{n=1}^{N} p(\theta_k|\mathbf{y}_n)||\mathbf{y}_n - \mu_k||^2}{\sum_{n=1}^{N} p(\theta_k|\mathbf{y}_n)}.$$

Diagonal covariance matrix, thus all variables are independent but have different levels of noise:

$$\mathbf{\Sigma}_k = \begin{bmatrix} \sigma_{1,k}^2 & 0 & \dots & 0 \\ 0 & \sigma_{2,k}^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sigma_{D,k}^2 \end{bmatrix}$$

$$(\sigma_{d,k})^2 = \frac{1}{D}\frac{\sum_{n=1}^{N} p(\theta_k|\mathbf{y}_n)(y_{d,n} - \mu_{d,k})^2}{\sum_{n=1}^{N} p(\theta_k|\mathbf{y}_n)},$$

where $y_{d,n}$ denotes the value of the n-th data point in the d-th dimension.
Full covariance matrix, thus there are dependencies or correlations between the variables and the levels vary:

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{1,1}^2 & \sigma_{1,2} & \dots & \sigma_{1,D} \\ \sigma_{2,1} & \sigma_{2,2}^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \sigma_{D-1,D} \\ \sigma_{D,1} & \dots & \sigma_{D,D-1} & \sigma_{D,D}^2 \end{bmatrix}$$

$$(\Sigma_k) = \frac{1}{D}\frac{\sum_{n=1}^{N} p(\theta_k|\mathbf{y}_n)(\mathbf{y}_n - \boldsymbol{\mu}_k)(\mathbf{y}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^{N} p(\theta_k|\mathbf{y}_n)}. \quad (4.6)$$

4.4 Generative Topographic Mapping

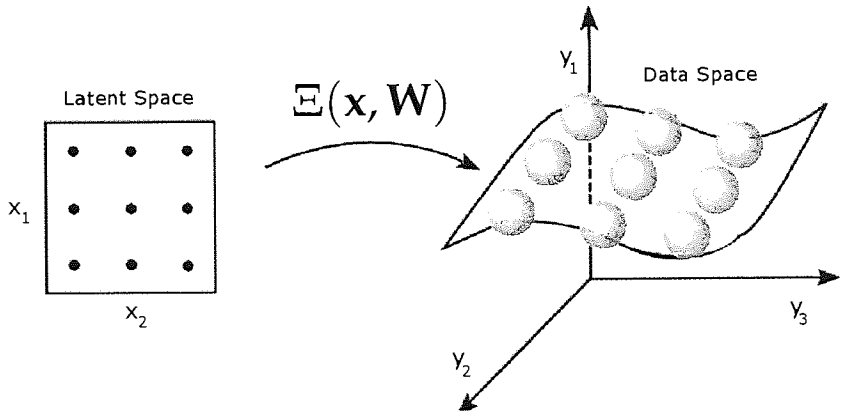


Figure 4.1: The non-linear function $\Xi(\mathbf{x}, \mathbf{W})$ defines a manifold S embedded in the data space given by the image of the latent variable space under the mapping $\mathbf{x} \rightarrow \mathbf{y}$.

The essence of GTM is to fit a *density model* to the data in the data space. The GTM is constrained to lie on a two-dimensional manifold. This can be envisaged as a flexible *"rubber sheet"*, typically two-dimensional, which is bent and stretched in the high-dimensional space to best fit the data density. This rubber sheet consists of a grid of points in the data space which are connected via a non-linear mapping function to a contorted grid of Gaussian centres in the data space. Thus the GTM may be described as a mixture of Gaussians constrained to lie on a two-dimensional manifold. To learn the intrinsic structure in the data, the rubber sheet is distorted by learning the non-linear mapping function using an EM algorithm so that the model best explains the data.

Latent variable models are usually defined as a mapping from data space to visualisation space. In contrast to this obvious mapping direction the GTM algorithm is defined as a mapping from visualisation to data space and applies Bayes' theorem to induce a posterior distribution in the visualisation space given the data.

First one considers a function $\mathbf{y} = \Xi(\mathbf{x}, \mathbf{W})$, where $\mathbf{W}$ is a weight matrix and the exact form of Ξ will be given later. This function maps points $\mathbf{x}$ in the L-dimensional visualisation space onto the points $\mathbf{y}$ which lie on an L-dimensional non-Euclidean manifold S embedded within the D-dimensional data space, shown for $L = 2$ and $D = 3$ in Figure 4.1.

Defining a probability distribution $p(\mathbf{x})$ for the data points in the visualisation space will induce a corresponding distribution $p(\mathbf{y}|\mathbf{x}, \mathbf{W})$ in the data space. Since in reality the data will not sit directly on the manifold, it is reasonable to include a noise model for the data $\mathbf{y}$. The distribution of $\mathbf{y}$ is chosen to be a radially-symmetric Gaussian centred on $\Xi(\mathbf{x}, \mathbf{W})$ with variance β^{-1}, for given $\mathbf{x}$ and $\mathbf{W}$, so that

$$p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2} \parallel \Xi(\mathbf{x}, \mathbf{W}) - \mathbf{y} \parallel^2\right\} . \tag{4.7}$$

The underlying assumption of the now spherical noise model is that all data dimensions are independent of each other and have the same amount of noise. Note that it is possible to use other probability distributions $p(\mathbf{y}|\mathbf{x})$ (e.g. Bernoulli for binary variables) or a combination of different distributions from the exponential family (i.e. see Clark and Thayer (2004) for examples and exact definition). For a given matrix $\mathbf{W}$, the distribution of y is obtained by integration over the distribution of $\mathbf{x}$,

$$p(\mathbf{y}|\mathbf{W}, \beta) = \int p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \beta) p(\mathbf{x}) dx . \tag{4.8}$$

For a given data set $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_N)$ of N data points, the parameter matrix $\mathbf{W}$ and the inverse variance β are estimated using the maximum likelihood principle. This can be done via minimising the negative log likelihood, given by

$$-L(\mathbf{W}, \beta) = -\ln \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{W}, \beta) .$$

After determining the prior distribution $p(\mathbf{x})$ and the functional form of the mapping $\Xi(\mathbf{x}, \mathbf{W})$ it is in principle possible to determine β and $\mathbf{W}$ by minimising $-L(\mathbf{W}, \beta)$. But the integral over $\mathbf{x}$ in (4.8) will, in general, be analytically intractable. Therefore a specific form of $p(\mathbf{x})$ is considered, where $p(\mathbf{x})$ is given by a sum of delta functions centred on the nodes of a regular grid in visualisation space

$$p(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^{K} \delta(\mathbf{x} - \mathbf{x}_k) ; \tag{4.9}$$

in this case the integral in (4.8) can be evaluated analytically. Now every point $\mathbf{x}_k$ is mapped to a corresponding point $\Xi(\mathbf{x}_k, \mathbf{W})$ in the data space, where it forms the centre of a Gaussian density function. Combining (4.8) and (4.9) the distribution function in the data space takes the form

$$p(\mathbf{y}|\mathbf{W}, \beta) = \frac{1}{K} \sum_{k=1}^{K} p(\mathbf{y}|\mathbf{x}_k, \mathbf{W}, \beta) , \tag{4.10}$$

and the corresponding negative log likelihood becomes

$$-L(\mathbf{W}, \beta) = -\sum_{n=1}^{N} \ln \left\{\frac{1}{K} \sum_{k=1}^{K} p(\mathbf{y}_n|\mathbf{x}_k, \mathbf{W}, \beta)\right\} .$$

Since the model consists of a mixture of distributions it is possible to find the optimal solution with an EM algorithm for β and $\mathbf{W}$, after choosing a specific functional form of $\Xi(\mathbf{x}, \mathbf{W})$. To derive the EM algorithm for the GTM model, $\Xi(\mathbf{x}, \mathbf{W})$ is chosen to be a regression model, linear in parameters, of the form

$$\Xi(\mathbf{x}, \mathbf{W}) = \mathbf{W}\Phi(\mathbf{x}) \,, \tag{4.11}$$

with the elements of $\Phi(\mathbf{x})$ consisting of B fixed radial basis functions $\Phi_j(\mathbf{x})$ (Broomhead and Lowe, 1988) and $\mathbf{W}$ being a $D \times B$ matrix.

In the case under consideration it is assumed that a hidden variable z_{kn} in (4.10) indicates which component (indexed over k) generated the data point $\mathbf{y}_n$. Therefore the EM algorithm can be formulated as follows. Assuming that $\mathbf{W}_{old}$ and β_{old} are given one can use the E-step to evaluate the *posterior responsibilities* of each Gaussian component k for every data point $\mathbf{y}_n$ using Bayes' theorem

$$R_{kn}(\mathbf{W}_{old}, \beta_{old}) = p(\mathbf{x}_k|\mathbf{y}_n, \mathbf{W}_{old}, \beta_{old}) \tag{4.12}$$

$$= \frac{p(\mathbf{y}_n|\mathbf{x}_k, \mathbf{W}_{old}, \beta_{old})}{\sum_{j=1}^{K} p(\mathbf{y}_n|\mathbf{x}_j, \mathbf{W}_{old}, \beta_{old})} \,. \tag{4.13}$$

Then the expectation of the complete data negative log likelihood has the form

$$\langle -L_{comp}(\mathbf{W}, \beta) \rangle = -\sum_{n=1}^{N}\sum_{k=1}^{K} R_{kn}(\mathbf{W}_{old}, \beta_{old}) \ln \{p(\mathbf{y}_n|\mathbf{x}_k, \mathbf{W}, \beta)\} \,. \tag{4.14}$$

Minimising (4.14) with respect to W and using (4.7) and (4.11) one obtains

$$\sum_{n=1}^{N}\sum_{k=1}^{K} R_{kn}(\mathbf{W}_{old}, \beta_{old}) \{\mathbf{W}_{new}\Phi(\mathbf{x}_k) - \mathbf{y}_n\} \Phi^T(\mathbf{x}_k) = 0 \,.$$

This can be written in matrix notation as

$$\mathbf{\Phi}\mathbf{G}_{old}\mathbf{\Phi}^T\mathbf{W}_{new}^T = \mathbf{\Phi}\mathbf{R}\mathbf{Y} \,, \tag{4.15}$$

with $\mathbf{\Phi}$ a $B \times K$ matrix with elements $\Phi_{kj} = \Phi_j(x_k)$, $\mathbf{Y}$ a $N \times D$ matrix with elements y_{nk}, $\mathbf{R}$ a $K \times N$ matrix with elements R_{kn} and $\mathbf{G}$ a $K \times K$ diagonal matrix with elements

$$G_{kk} = \sum_{n=1}^{N} R_{kn}(\mathbf{W}_{old}, \beta_{old}) \,.$$

Equation (4.15) can be solved for $\mathbf{W}_{new}$ using standard matrix inversion techniques like the Cholesky or QR decomposition (Chapra, 2004). Similarly, to minimise (4.14) with respect to β one obtains the following formula

$$\frac{1}{\beta_{new}} = \frac{1}{ND}\sum_{n=1}^{N}\sum_{k=1}^{K} R_{kn}(\mathbf{W}_{old}, \beta)||\mathbf{W}_{new}\Phi(\mathbf{x}_k) - \mathbf{y}_n||^2 \,.$$

The EM algorithm alternates between the E-step, given by evaluating (4.12), and the M-Step, evaluating $\mathbf{W}_{new}$ and β_{new}, until it converges to a (local) minimum and can be written as:

E-Step:

1: Set $\mathbf{W}_{old} = \mathbf{W}_{new}$ and $\beta_{old} = \beta_{new}$
2: Calculate $R_{kn}(\mathbf{W}_{old}, \beta_{old})$

M-Step:

1: Calculate $\mathbf{W}_{new}$ with R_{kn}
2: Calculate β_{new} with R_{kn}

An example of the result can be seen in Figure 4.2 where a GTM was fitted to the S-shaped data.

4.4.1 Data Visualisation using GTM

The visualisation of data can be achieved using Bayes' theorem to invert the transformation from visualisation space to data space. Following the choice of the prior distribution given by (4.9) one obtains a posterior distribution as a sum of delta functions with coefficients given by the responsibilities R_{kn}. These can be used to create a posterior responsibility map for single data points in the two-dimensional visualisation space. In the case where one has only a small dataset this might be a valuable way to look at the data if it is integrated in an interactive view.

In practice, however, looking at the distribution of each data point individually is impossible and unreasonable for large data sets. It is often convenient to summarise the posterior distribution by the mean, given by

$$\langle \mathbf{x}|\mathbf{y}_n, \mathbf{W}^*, \beta^* \rangle = \int p(\mathbf{x}|\mathbf{y}_n, W^*, \beta^*)\mathbf{x}\, dx \tag{4.16}$$

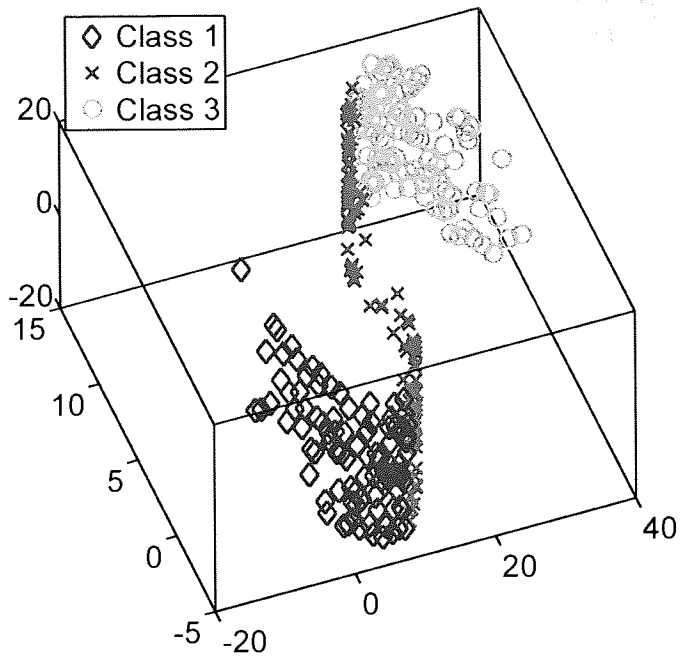
$$= \sum_{k=1}^{K} R_{kn}\mathbf{x}_k\,, \tag{4.17}$$

and thus obtain a mapping for each data point in the visualisation space. Examples for this mapping can be seen for the S-shaped and Swiss-roll data in Figure 4.3.

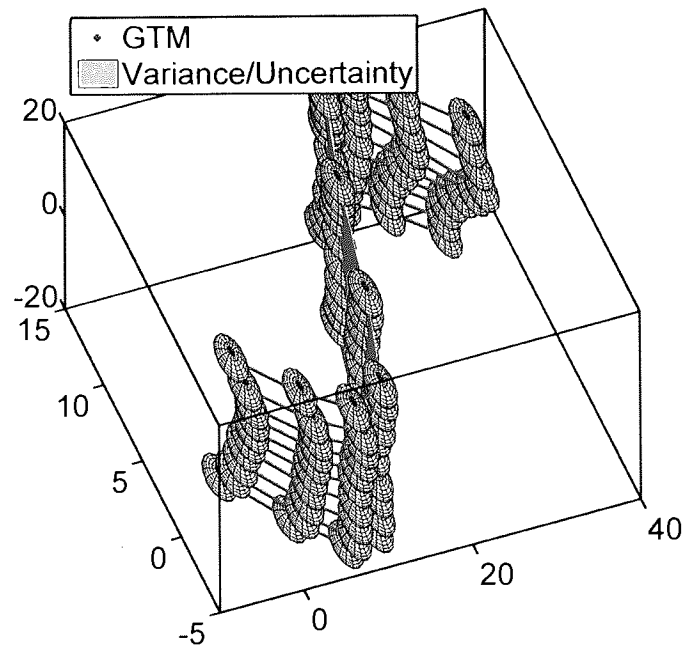
However this can be very misleading if one deals with a posterior distribution which is multi-modal. A way to check this is to evaluate the mode of the distribution

$$i^{max} = \arg\max_k R_{kn}\,, \tag{4.18}$$

In the case where the mean and mode do not match further analysis is needed. The mismatch could either indicate a bad model fit or could be a feature of the data. An indication could be obtained by looking at the single responsibility maps given by plotting $\mathbf{R}_{kn}$ on the grid, running over $k = (1, \ldots, K)$. However this is very tedious and not practicable for non-statistical experts. The recommendation would therefore be to use the modes only as a diagnostic for a bad model fit. Therefore if the modes and means are plotted and one can identify cases where the mode and mean for the same data point do not match, given an appropriate amount of tolerated variance, one should assume a bad model fit and discard the actual visualisation or only use it very carefully.

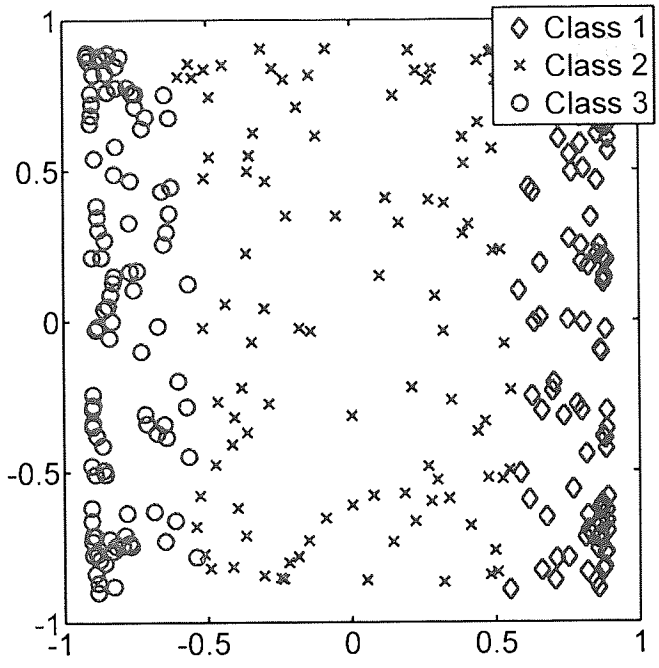


(a)

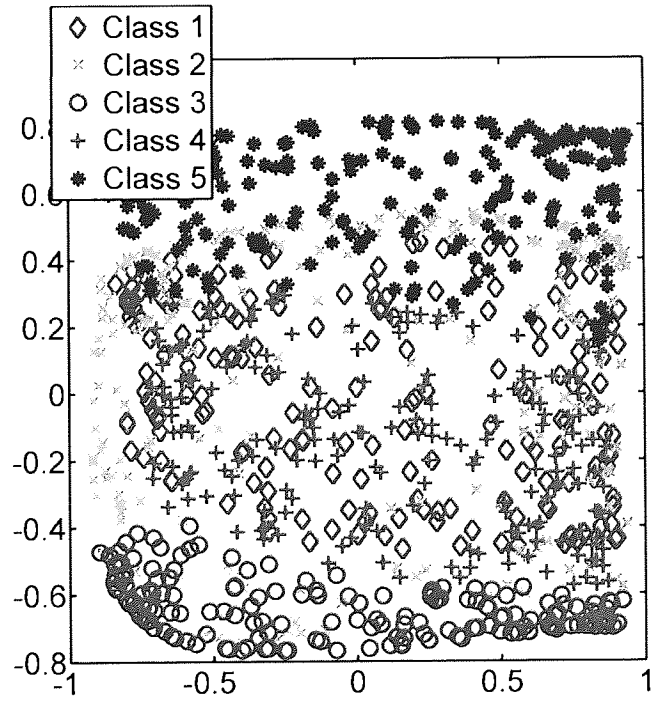


(b)

Figure 4.2: (a) The S-shaped 3D test data. (b) A 15x15 GTM with 16 RBF centres which was fit to the S-shaped test data after 50 iterations with the EM algorithm and initialisation with PCA. The GTM manifold has aligned itself to the structure of the data and fits it nearly perfectly.



(a) S-shaped data



(b) Swiss-roll data

Figure 4.3: Projection of simple data sets using the GTM algorithm which was initialised with PCA. The structure of the S-shaped data in (a) is captured and one can clearly see that the class structure is preserved. This is not the case with the Swiss-roll data in (b) where GTM fails to preserve the structure of the classes.

4.4.2 Initialising GTM

Initialisation is recognised to be a crucial issue in most non-linear data visualisation and clustering algorithms. To fit any kind of model one seeks to minimise an error function in a typically multi dimensional space, which is always a non-trivial task. The error functions are usually difficult to describe, have multiple minima and plateaus and thus finding a good solution relies heavily on a good initialisation. The GTM is no exception, especially since we use the iterative EM algorithm which is prone to run into local minima.

To initialise the GTM the Gaussian centres need to be placed in the data space. Thus the weights in equation (4.11) need to be set as well as the variance to be able to start the EM algorithm and calculate the responsibilities. The initialisation is crucial because the EM algorithm will try to minimise the complete data likelihood in (4.11) starting from that initial point. Thus the initial point should be as close as possible to the global minimum because the EM algorithm will always decrease the likelihood and thus can not escape from local minima. The initialisation could be done by just using random weights however practice has shown (Nabney, 2002; Maniyar *et al.*, 2006b) that a good initialisation for GTM is along the first one or two principal components of the data with a sufficiently large variance. One can imagine that the rubber sheet is the hyperplane spanned by the first one or two principal components. If the data structure is mainly linear this will lead to good results since most of the structure can already be captured by PCA which is used to initialise GTM. Subsequently GTM simply fine-tunes these results. Problems arise when PCA fails to capture the structure, for example when the data are generated by a underlying non-linear function like in the case of the Swiss-roll data or in the case of the toy data sets in chapter 3. This will lead to a non-optimal initialisation and GTM will in most cases not be able to fully capture the structure. In the worst case GTM will fail to capture any structure and produce misleading results, as can be seen with the Swiss-roll data in Figure 4.3. Figure 4.4 demonstrates a quite simple problem in 2 dimensions on a data set generated by a sine function. The initialisation with PCA is non-optimal and leads to a very bad fit of GTM, which cannot identify the actual structure of the data. However, an alternative initialisation in this case leads to a satisfying result as can be seen in Figure 4.5. A novel extension of GTM to utilise alternative initialisations will be presented and discussed in chapter 5.

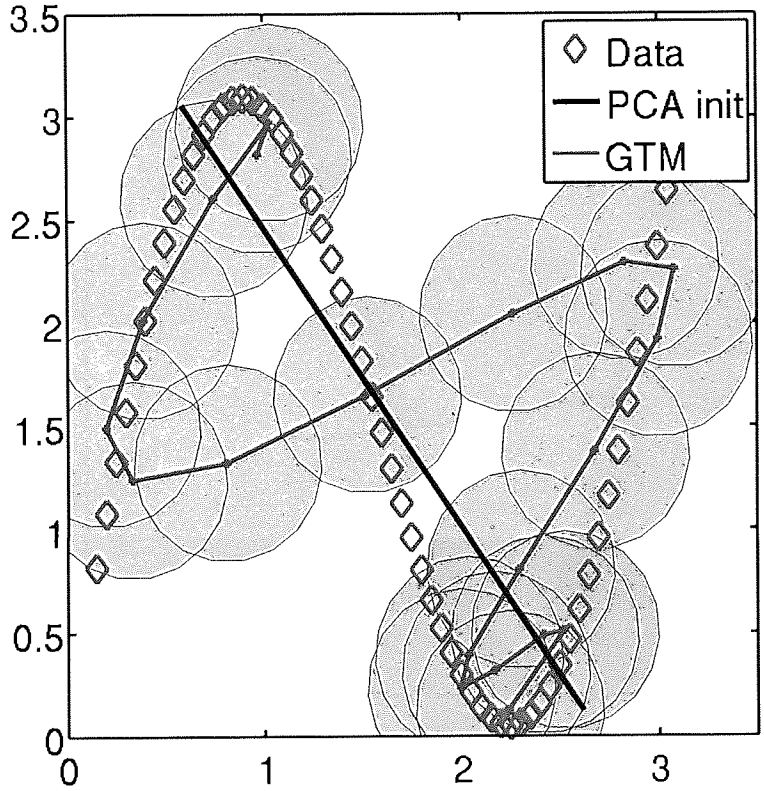


Figure 4.4: GTM is initialised with PCA and fitted to a simple sine function. The GTM fails to capture the structure of the data regardless of the number of iterations of the EM algorithm. The green circles indicate the uncertainty/variance around the GTM and their size indicates as well that the GTM has trouble to fit the structure of the data. The green line visualises the actual position and structure of the GTM manifold and it is obvious that it is not fitting the data at all. The black line indicates the initialisation which was used at the beginning of the algorithm, which is the first principal component in this case.

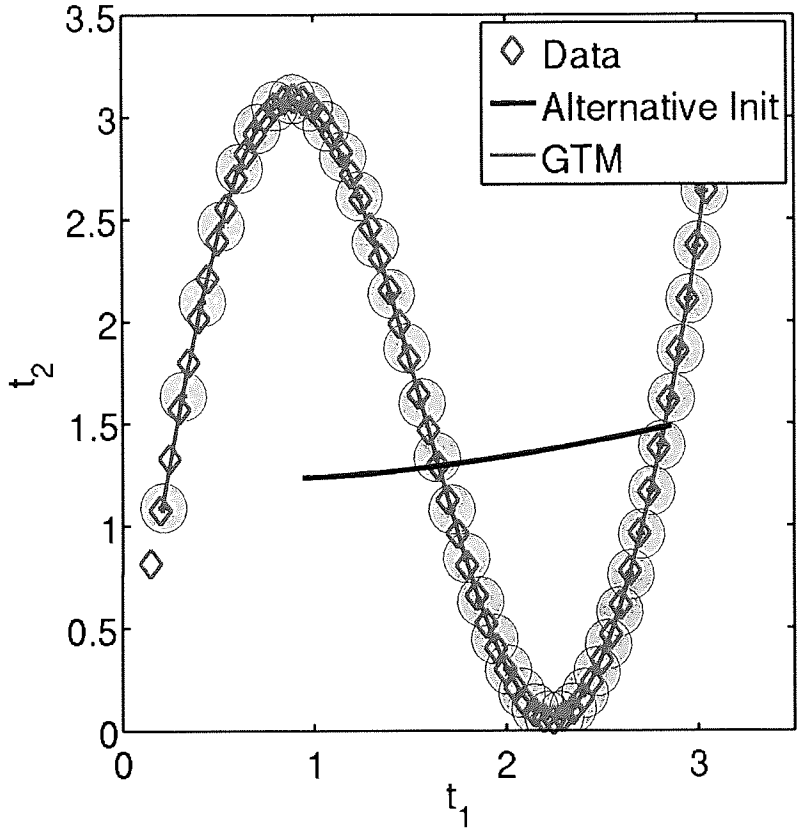


Figure 4.5: GTM is initialised with a beneficial random initialisation and manages to fit the data with convergence of the EM algorithm after 15 iterations. The green circles indicate the uncertainty/variance around the GTM and their size indicates as well that the GTM fits the data very closely. The green circles indicate the uncertainty/variance around the GTM and their size indicates that the fit is very good. The green line visualises the actual position and structure of the GTM manifold and it is perfectly aligned with the structure of the data. The black line indicates the initialisation which was used at the beginning of the algorithm, which in this case was chosen by tying random vectors until a good fit was achieved.

4.5 Other visualisation algorithms

The following visualisation algorithms are only described very briefly. A more technical description together with visual demonstrations of their projection capabilities is given in the Appendix B.

4.5.1 PCA

Principal component analysis (Jolliffe, 1986) is the most widely used method for dimension reduction, and thus visualisation. It transforms the data space into a set of orthogonal principal components which are termed scores. The principal components can be described as the directions along which the data set has the biggest variance.

4.5.2 Probabilistic PCA

The probabilistic version of PCA, called PPCA, (Tipping and Bishop, 1999) extends conventional PCA to a probabilistic framework while not changing the mapping. The maximum likelihood solution of PPCA has been shown to be the same as the one obtained through conventional PCA. The model serves as a building block for other algorithms and it can be extended to Kernel PCA, a non-linear version of PCA, or to the Gaussian Process Latent Variable Model, where one integrates over the mapping weights and optimises the positions in the visualisation space directly.

4.5.3 Kernel PCA

Kernel PCA (Schoelkopf *et al.*, 1998) is a method for using a linear algorithm to solve a non-linear problem by using a non-linear function to map the original observations into a higher-dimensional feature space, where the linear PCA algorithm is subsequently used. The downside of transforming the data into a higher dimensional feature space is that Kernel PCA loses the interpretability of the loadings for the principal components. Additionally the inverse mapping is not know most of the time and thus one loses the possibility of projecting the points in the visualisation space back to the data space.

4.5.4 Gaussian Process Latent Variable Model (GPLVM)

To extend PPCA to GPLVM (Lawrence, 2005) one uses a linear Gaussian process prior over the space of mapping functions (i.e. over the weights $\mathbf{W}$). The result is a probabilistic visualisation algorithm which generates a space of possible mapping functions with the associated probabilities for each mapping. Thus one can quantify the actual statistics for the mapping of a data point (i.e. the mean and variance). Like GTM the GPLVM is an generative model and needs to be

initialised and consequently optimised. However the GPLVM optimises the locations of the projected data points in the visualisation space directly and thus one can use an arbitrary projection to initialise the algorithm.

4.5.5 MDS

Multidimensional scaling (MDS) (Cox and Cox, 1994) describes a class of methods which provide insight into the underlying structure and relations of a data set by providing a geometry-preserving representation of this data set. The underlying idea is to use a proximity measures in the data space and the visualisation space. One then calculates the projection by preserving as much of the original proximity/geometry as possible while optimising the positions of the data points in the lower dimensional space.

4.5.6 Neuroscale

Neuroscale (Lowe and Tipping, 1996) is a dimension-reducing and topographic transformation to visualise and analyse high-dimensional data. It is an MDS style algorithm however instead of optimising the locations of the projected points directly one uses and optimises an RBF network to predict the locations of the data points in the visualisation space.

4.5.7 Isomap

The Isomap algorithm (Tenenbaum *et al.*, 2000) can be seen as a special type of MDS where the distances in the proximity measure are chosen to be of a particular form. These distances are called geodesic and are computed by using a neighbourhood graph over all the data. The idea is to only use local distances for every point and compute the global distances along the distribution of the data. Due to the geodesic distances the algorithm can fit highly non-linear data sets like the Swiss-role.

4.6 Summary

All the methods described in this chapter have their individual strengths and weaknesses and an overview of their characteristics can be seen in Table 4.1. The most commonly used method is PCA. PCA is computationally very fast and easy to use. For most practitioners it is easy to understand since it is based on undergraduate mathematics. One can easily relate PCA to regression problems where one also projects the data points onto a line or hyperplane. The interpretability of both the loadings and scores is a big asset for many practitioners since they can investigate the contribution of the variables to the final plots as well as their relation to each other. Further through the use of cross validation (Krzanowski,

	Probabilistic	Missing Data	$\mathbf{Y} \rightarrow \mathbf{X}$	$\mathbf{X} \rightarrow \mathbf{Y}$	Non - Linear	Local Structure
PCA		E	Y	Y		
BPCA	Y	Y	Y	Y		
KPCA		E	Y		Y	
GPLVM	Y	E	E	Y	Y	Y
MDS					Y	
Neuroscale			Y		Y	
Isomap					Y	Y
GTM	Y	Y	E	Y	Y	E

Table 4.1: Overview of the characteristics of the different algorithms. A 'Y' indicates the algorithms exhibits the quality, an 'E' indicates that there is an extension or heuristic to the algorithm that exhibits the quality. The different characteristics are: *Probabilistic:* Is the method based on a probabilistic framework? *Missing Data:* Can the method deal with missing data? $\mathbf{Y} \rightarrow \mathbf{X}$: Does the method provide a mapping function from data to visualisation space? $\mathbf{X} \rightarrow \mathbf{Y}$: Does the method provide a mapping from the visualisation to the data space? *Non-Linear:* Does the method allow for non-linear mappings? *Local Structure:* Does the method allow to explore local structures based for example on a connected graph?

1987) once can identify problems or uncertainties with the projection. The downside of the standard PCA algorithm is the inability to directly cope with missing data as well as the restriction to a linear model. However there are extensions to the PCA algorithm which can handle missing data and these are discussed in chapter 6. The probabilistic formulation of PCA called PPCA gives rise to the non-linear Kernel PCA, GPLVM and can handle missing data. KPCA transforms the original data space into a higher dimensional feature space using a non-linear mapping. This way one can introduce non-linearity into the algorithm. However this transformation also causes a loss of the interpretability of the loadings. The more principled probabilistic alternative to KPCA is GPLVM which utilises a Gaussian process for the mapping. The advantages of GPLVM are that it allows for non-linear mappings, can deal with missing data and exploiting the noise model can inform the user about the certain and uncertain regions in the projection. The noise model however is restricted to the visualisation space and thus one cannot use it to incorporate information one might have about noise in the data space. Another big advantage of GPLVM is that it can be initialised with any other mapping algorithm. This way GPLVM is, for example, able to utilise the advantages of local linear embedding techniques like Isomap. The downsides of the GPLVM are that there is no interpretability of loadings and the far bigger computational costs for the algorithm which scales $O(N^3)$.

An alternative principled method is the GTM (Generative Topographic Mapping) which is based on SOM (Self Organising Maps). The model is probabilistic and specifies a non-linear mapping from the visualisation space to data space,

which can be inverted using Bayes' theorem. In essence the GTM is a constrained mixture of Gaussians which is fitted to the data using an EM algorithm. The model can deal with missing data and has a noise model in data space. The noise model provides evidence of the fit in the data space. Further one can extend GTM to include expert knowledge and this novel extension will be explored in chapter 5. Another novel extension developed in the scope of this thesis will deal with alternative initialisations for GTM. It will enable GTM to be used more like GPLVM and allow the model to utilise alternative initialisations like local linear embeddings. These two extensions will also be discussed in chapter 5.

Geometric distance preserving methods based on MDS (Multi-Dimensional Scaling) are an alternative approach. These methods try to find a projection by minimising a loss function. Depending on the algorithm they can be computationally demanding. They have the advantage that one can easily include additional knowledge about class labels in the case of Neuroscale or use different distance measures in the case of local linear embedding techniques like Isomap. With the exception of Neuroscale, MDS algorithms in general do not provide a mapping function and thus one needs to recompute the mapping every time one adds a new point. They further cannot deal with missing data and are susceptible to noise in the data.

5 Extensions to GTM

CONTENTS

5.1 Block Extension to GTM (B-GTM) 79
5.1.1 Heuristics to stabilise the EM algorithm 80
5.1.2 Variable Block Determination using Optimal Leaf Ordering (OLO) 81
5.1.3 Variable Block Determination using Quick Bayesian Correlation Estimation (QuickBCE) 83
5.2 GTM Visualisation Space Reverse Mapping Initialisation (GTM-VSRMI) 85
5.3 Assessing the novel Extensions 87
5.3.1 B-GTM 89
5.3.2 QuickBCE vs. OLO 99
5.3.3 GTM-VSRMI 101
5.4 Summary 103

The GTM algorithm is a well known and often used method within the visualisation community. Multiple extensions have been proposed and developed to improve the visualisation and extent or adapt it for specific problem scenarios.

The extensions may be separated into three classes: *Analytical Extensions* e.g. the magnification factor (Svensén and Williams, 1997) to measure the stretching of the manifold, *Structural Extensions* e.g. the hierarchical extension for GTM to explore large scale data sets (Tino and Nabney, 2002) and *Fundamental Extensions* e.g. the substitution of the Gaussian nodes by hidden Markov tree models (Gianniotis and Tino, 2008). The first class consists of extensions providing additional analytics, typically focusing on gaining more knowledge from the actual projection. They provide the practitioner with additional possibilities to explore the data space by utilising the characteristics of the GTM. The second class consists of single or structural alterations to the algorithm to deal with different data types, non-Gaussian distributions or employ a full Bayesian treatment. All these alterations do not change the algorithm in a major way. This allows them to be combined with the analytical extensions and possibly amongst themselves. The structural extensions consists of major changes to the algorithm to deal with different classes of problems, use different metrics or substitute the mixture components with a different class of models. The effects of these alterations to the GTM algorithm are not easily understood. Therefore it might be quite difficult or in certain cases impossible to combine the structural extensions with any extensions from the other two classes.

The two proposed novel extensions, block GTM (B-GTM) and GTM with visualisation space reverse mapping initialisation GTM-VSRMI both fall into the second category. They can be combined with most extensions of the second category and do not impede the usage of diagnostics from the first category.

The following paragraphs present a short review of known extensions for the GTM algorithm. They are grouped by their class and presented in historical order ranked by the date of the last paper with substantial contribution to the extension.

Analytical Extensions: One of the most widely used diagnostic is the magnification factor (Svensén and Williams, 1997). It represents the extent to which an area is magnified on the projection of the data space (i.e. it shows how strongly the manifold was stretched in the specific area of the data space and reveals if points are further apart than is implied by the two dimensional projection).

Another useful diagnostic is the local directional curvature of the projection (Tino *et al.*, 2001). It provides the user with a facility for monitoring the amount of folding and neighbourhood preservation in the fitted data manifold.

Structural Extensions: Hierarchical GTM is a hierarchical visualisation system which is based on active user interaction and allows the user to explore interesting regions in more detail by manual selection (Tino and Nabney, 2002). This was extended by Nabney *et al.* (2005) in a semi-supervised learning approach and by Maniyar and Nabney (2005) to aid the development of effective local prediction

models for regression.

Two extensions of GTM which deal with unsupervised feature selection or feature relevance determination have been developed simultaneously and independently of each other (Maniyar and Nabney, 2006a; Vellido, 2006a). They both build on previous work related to modelling the background noise when fitting Gaussian mixture models (Law *et al.*, 2004). The extension allows the model to account for less important and mainly noise dominated variables. This makes it possible to quantify which variables have the most effect on the visualisation, similarly to the loadings in PCA. In contrast to PCA the selection of the features is not done after the model was fitted to the data. It is an integral part of the algorithm which assigns a lesser weight to the noisy and uninformative variables. However the weights in the GTM feature selection account for the contribution to the complete model and can not be used to distinguish between the contribution of data dimensions to certain axis in the projection.

Multiple extensions have been developed to deal with non-Gaussian distributed data. Most of them concern the special case of discrete data and substitute the Gaussian distribution of the nodes with a Bernoulli, multinomial or Poisson distribution(Girolami, 2001; Priam *et al.*, 2008). An extension of GTM utilising a mixture of Student t-distributions has been proposed (Vellido, 2006b) to make the model more robust to outliers and non-Gaussian continuous data.

Other extensions propose a variational Bayesian treatment of GTM utilising a Gaussian Process (Olier and Vellido, 2008b) instead of the normally employed EM algorithm and the standard RBF mapping. This extension solves the problem of choosing the number of RBF centres and the issue of over fitting the model, in the case of too many RBF centres.

To capture the dynamics of multivariate time series through visualisation Bishop *et al.* (1997) proposed the Generative Topographic Mapping Through Time. Subsequently the algorithm has been extensively tested, combined with feature relevance determination and put into a variational Bayesian framework (Olier and Vellido, 2006; Olier and Vellido, 2008a).

To facilitate the usage of GTM in conjunction with clustering algorithms an extension or better a special algorithm for fuzzy clustering has been proposed which utilises the Gaussian centres as candidate seeds for the initialisation of the algorithm (Bose and Chen, 2009).

Fundamental Extensions: One of the attractive features of GTM is the grid structure in the data space which is achieved through the mapping from a grid in the visualisation space. This feature is exploited by several alterations to the algorithm where the Gaussian nodes are substituted by some other kind of model. One example is the usage of hidden Markov tree models (Gianniotis and Tino, 2008). Another example is the usage of independent probabilistic principal component models termed locally linear generative topographic mapping (Verbeek *et al.*, 2002).

Using the grid structure of the algorithm but disregarding the idea of a strict

probabilistic model one can modify GTM into a heuristic algorithm. One example is the topographic neural gas algorithm which combines the harmonic mean with the neural gas algorithm to fit the centres and substitutes the Gaussian covariance by a tri-cubic kernel which is used to estimate the *responsibilities* (Pena and Fyfe, 2006).

Other possible approaches facilitate different metrics, such as the modification of GTM to penalise divergences between the Euclidean distances from the data points to the model prototypes and the corresponding geodesic distances along the manifold (Cruz-Barbosa and Vellido, 2008).

Chapter Overview: The following chapter first introduces the block extension of GTM, which improves the model by integrating prior knowledge about the covariance structure. To acquire the knowledge about this structure two possible approaches are discussed. The first approach is based on the optimal leaf ordering (OLO) algorithm (Bar-Joseph *et al.*, 2003) which sorts the variables to optimise their ordering for grouping of correlation coefficients. This requires further post processing by an expert, which may be based on a heat-map of the correlations. An alternative approach to acquire the knowledge is to us a more automated approach based on a variable grouping algorithm. One possibility is the Bayesian correlation estimation (BCE) based on an MCMC algorithm (Liechty *et al.*, 2004). The original algorithm was modified by us to improve mixing of the chains and this novel version is called QuickBCE. However no extensive experiments have been conducted with either of the algorithms (OLO and QuickBCE). They are both introduced for the proof of concept and further research in this area is needed.

In the second section a novel way to initialise the GTM called visualisation space reverse mapping initialisation (VSRMI) is introduced. Before the development of this extension GTM was either initialised randomly or along the axes of the first two principal components of GTM. VSRMI allows to initialise GTM with any 2D mapping of the data. This makes it possible to use non-linear algorithms like Isomap for the initialisation of GTM, which greatly enhances the capabilities of the model to pick up non-linear structures in the data.

In the third section the novel extensions (B-GTM and VSRMI) are assessed. This is a difficult task since one normally works with unlabelled data in geochemistry and a priori does not know what a good visualisation should look like. To help with this task a novel approach to assess visualisation models is proposed. This approach is similar to leave-one-out cross validation. In short, we propose to measure how well a model fitted to a complete data set can estimate known values that are excluded for the purpose of validation. The resulting diagnostic is similar to the likelihood. The likelihood has the disadvantage that one never knows what the best likelihood is and one needs to compare different models against each other. The new diagnostic based on missing data has the advantage that one knows that no model can do better than have an error of zero when estimating the missing values.

5.1 Block Extension to GTM (B-GTM)

In Geochemistry one often has prior information about the correlations of variables. The distribution of the variables is highly depend on the underlying geochemical processes. Thus one often can see blocks or clusters of variables when looking at the ordered heat-maps of the correlation coefficients. A novel approach to include prior information about the correlations of variables into GTM is to use a full covariance matrix in the noise model and to enforce a block structure onto it. This results in a reasonably sparse covariance matrix and keeps the number of unknown parameters low. The additional flexibility of the model, introduced through the more densely populated covariance matrix, allows the model to fit the data more closely. After ordering the variables by their known or estimated groups, the covariance matrix has the following structure:

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 & \dots & 0 \\ 0 & \Sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \Sigma_p \end{bmatrix},$$

with Σ_1 to Σ_p being the submatrices of correlated groups of variables. This implies that the correlation between variables in distinct groups is negligible. The extension of the learning algorithm is straightforward and the only changes occur in the computation of R in the E-step and of Σ in the M-step. In the E-Step the computation of the posterior probabilities of the kth Gaussian component changes because of the change from a spherical to a block covariance matrix Σ:

$$\begin{aligned} R_{kn}(\mathbf{W}_{old}, \Sigma_{old}) &= p(\mathbf{x}_k|\mathbf{y}_n, \mathbf{W}_{old}, \Sigma_{old}) \\ &= \frac{p(\mathbf{y}_n|\mathbf{x}_k, \mathbf{W}_{old}, \Sigma_{old})}{\sum_{j=1}^{K} p(\mathbf{y}_n|\mathbf{x}_j, \mathbf{W}_{old}, \Sigma_{old})} . \end{aligned} \tag{5.1}$$

To define the modified M-step one first derives the update for the full covariance matrix Σ. Taking the derivative of the negative log likelihood with respect to Σ_j we get the updates for all the sub matrices:

$$-\frac{\partial L(\mathbf{W}, \Sigma_j)}{\partial \Sigma_j} = -\sum_{n=1}^{N} \frac{D_j}{2} \Sigma_j^{-1} - \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} R_{in} \Sigma_j^{-1} \mathbf{a}_{knj} \mathbf{a}_{knj}^T \Sigma_j^{-1} ,$$

where $\mathbf{a}_{knj} = (\Theta(\mathbf{x}_k, \mathbf{W}) - \mathbf{y}_n)_j$ is only calculated on the dimensions belonging to Σ_j and D_j is the number of dimensions. Setting the derivative to zero we obtain

$$\Sigma_j = \frac{1}{N D_j} \sum_{n=1}^{N} \sum_{k=1}^{K} R_{in} \mathbf{a}_{knj} \mathbf{a}_{knj}^T ,$$

which can be described as the average empirical covariance calculated over all the Gaussians.

5.1.1 Heuristics to stabilise the EM algorithm

A major problem with the practical application of the EM algorithm with both block and full GTM, especially when working with high-dimensional data, is the collapse of the variance to very small values. This collapse is partly due to numerical problems when calculating the activation given by (5.1) and becomes especially problematic with high dimensional data sets. Another part of the problem is the probabilistic nature of the algorithm which is having problems with singularities because a variance close to zero will result in a very high likelihood. This is most serious in the case of high dimensional data sets where one might obtain a rank deficient covariance matrix due to dependencies of variables on each other.

Unfortunately this is the case in nearly any applied setting where one needs to analyse data. The problem is caused by the determinant of the covariance matrix which becomes very small. The reason is the direct relation of the magnitude of the determinant of the positive definite covariance matrix with the number of dimensions. Because of the exponential nature of the Gaussian and limited machine precision most activations get rounded to zero if the dimensionality becomes too big. This results in a smaller estimate of the covariance matrix and after a few iterations in the breakdown of the algorithm. Even if the algorithm does not directly break down, possible consequences are points where the responsibility R_{ij} is zero for all Gaussians. This occurs when the covariance matrix is too small and points which are too far away from the manifold get zero as responsibility for all Gaussians because of the limited machine precision. To prevent this from happening we use heuristics while calculating the activation as well as the covariance. These heuristics prevent the responsibilities of a data point from going to zero for all the Gaussian kernels and prevent the collapse of the variance.

The first heuristic is a simple check, where it is tested if the direct neighbouring Gaussian nodes of every Gaussian node in the grid are still in the two-sigma interval of the distribution of this node. If this is not the case the covariance matrix is multiplied with a small number > 1 until the condition is fulfilled. This provides a lower bound on the determinant of the covariance matrix which is based on the distance of the nodes to each other in the data space. The second heuristic checks if the responsibilities are all zero for one data point and substitutes these with the inverse distance to the five nearest gird points if this is the case.

However these heuristics just prevent the algorithm from producing completely meaningless results. In the case of very high-dimensional data the algorithm still does not work. What one effectively observes is that the algorithm stops after one or two iterations because the values in $\mathbf{R}$ in equation (5.1)are so small that the effect of the update is negligible. In this case the model does not move away from the initial state.

In the case of the full GTM one has an additional numerical problem. Because of the large number of parameters in the full covariance matrix and the limited sample size the EM algorithm starts to produce estimates of the covariance matrices which are not positive semi-definite. This problem is not easily solved and

the *quick fix* of adding Gaussian noise to the diagonal until the matrix is positive definite might result in the breakdown of the algorithm. This happens because if too much noise is added, the noise part in the diagonal will start to dominate the structure of the covariance matrix.

The described heuristics were successfully tested on datasets with a maximum of 71 dimensions. However there is no guarantee that they will work with higher dimensional data sets and future research in this area is needed. This should include alternative approaches to the heuristics like placing a prior over the covariance matrix which will penalise too small values and should also keep the algorithm from breaking down.

5.1.2 Variable Block Determination using Optimal Leaf Ordering (OLO)

A simple and straightforward method to obtain the block structure for the covariance matrix is to visualise the correlation coefficients as a heat-map as shown in Figure 3.3 in chapter 3. However for this method to be successful one needs to order this heat-map so that highly correlated variables are close to each other (i.e. forming blocks). The ordering of heat-maps is a typical problem faced in combinatorial data analysis (Arabie and Hubert, 1996) and the process is called ordination, sequencing or seriation (Hahsler *et al.*, 2008) a term dating back to 1899 where it was first used in archaeology (Petrie, 1899).

One approach is to generate a dendrogram using hierarchical clustering combined with heuristics to reorder the leaves to reflect their proximity (i.e. it reorders rows and columns in the heat-map with the aim of placing similar variables close together). To achieve this the tree is ordered in such a way that the distance between the neighbouring leaves is minimised. Solving this problem is akin to finding a solution for the travelling salesman problem. Following this approach the **absolute** values of the correlation matrix are used to generate a pairwise distance matrix where one wants to determine the ordering which minimises the sum of distances between consecutive elements. There are many ways to approach the problem and a good review of methodologies can be found in Hahsler *et al.* (2008): for example, simulated annealing (Morris *et al.*, 2003) or optimising the Hamilton path length by dynamical programming methods like optimal leaf ordering (OLO) (Bar-Joseph *et al.*, 2003) .

Hahsler *et al.* (2008) suggests that OLO is one of the best methods for seriation. Further the algorithm is supplied in the Matlab Bioinformatics toolbox which provides a fast and error-free implementation.

To give a short outline of the algorithm the following notation is used: A tree T has n leaves denote by $(z_1, \ldots, z_n)$ and with $n-1$ internal nodes denoted by $\nu_1, \ldots, \nu_{n-1}$. Every node has a left and a right child node labelled $\nu.l$ and $\nu.r$ respectively. The algorithm is recursive and thus works its way from the lowest level to the top. At each step the algorithm checks if it minimises the distance between the nodes if it flips the nodes over as illustrated in Figure 5.1. The al-

gorithm only depends on the distance or utility matrix S, which in our case is generated from the absolute values of the correlation coefficient matrix.

The OLO algorithm in outline is:

1: optOrdering(v,S)
2: **if** (v is a leaf) **then**
3: return v
4: **else**
5: $v.l$=optOdering($v.l$,S) // Order the left tree of v
6: $v.r$=optOdering($v.r$,S) // Order the right tree of v
7: **end if**
8: Flip node $v.l$ to $v.r$ and $v.r$ to $v.l$ if it minimises the sum of the distance of all adjacent nodes.
9: return v.

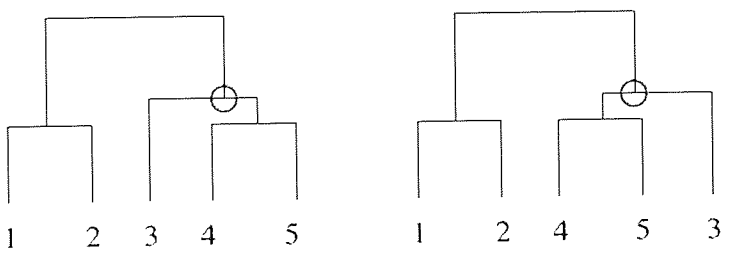


Figure 5.1: Changing the leaf order by an internal node flip. The node is marked with a red ring.

5.1.3 Variable Block Determination using Quick Bayesian Correlation Estimation (QuickBCE)

Another approach to estimate the group structure in the model is a modified version of the Bayesian Correlation Estimation based on the paper of Liechty *et al.* (2004) which relates to Bernard *et al.* (2000).

For the grouping one is only interested in the off-diagonal elements of the empirical data correlation coefficient matrix $\mathbf{C}$. Assuming the simple case where $\mathbf{C_{ij}} \sim N(\mu, \sigma^2)$ the estimation of the underlying parameters μ and σ for the model is straight forward. In the case of a full Bayesian treatment sensible priors would be $\mu \sim N(0, \tau^2)$ and $\sigma^2 \sim IG(\alpha, \beta)$ (IG= Inverse Gamma), where α is a shape parameters, and τ^2 and β are the scale parameter, which are treated as elicited (i.e. known and fixed). Thus the full conditional posterior distributions are:

$$f(\mu|\mathbf{C}, \tau^2, \sigma^2) \propto \prod_{i<j} \exp\left\{-\frac{(C_{ij}-\mu)^2}{2\sigma^2}\right\} \exp\left\{-\frac{\mu^2}{2\tau^2}\right\}, \tag{5.2}$$

$$f(\sigma^2|\mathbf{C}, \mu, \beta, \alpha) \propto \prod_{i<j} \exp\left\{-\frac{(C_{ij}-\mu)^2}{2\sigma^2}\right\} \left(\frac{1}{\sigma^2}\right)^{\alpha-1} \exp\left\{-\frac{\beta}{2\sigma^2}\right\}. \tag{5.3}$$

The model can be extended to allow for groups of variables:

$$f(\mathbf{C}|\mu, \sigma^2, \vartheta) \propto \prod_{i<j} \exp\left\{-\frac{(C_{ij}-\mu_{\vartheta_j,\vartheta_i})^2}{2\sigma^2}\right\} I\{\mathbf{R} \in \mathcal{R}^J\}$$

with ϑ_i is the index variable for the groups and $\vartheta_i \sim multinomial(p)$. With regards to B-GTM the estimates for the distribution of the groups $p(\vartheta_i|-)$ together with the estimates for $p(\mu_{k_1,k_2}|-)$ can be used to pre-define a block structure, where one groups the variables into different blocks. To explain the notation: the estimate $\mu_{1,1}$ is the mean for the first group,the estimate $\mu_{1,2} = \mu_{2,1}$ is the mean of the correlation between groups. The noise parameter σ^2 is assumed to be common for all groups. This keeps the number of free parameters to a minimum, which helps with convergence and mixing of the parameters.

Sampling the full conditional of ϑ, μ and σ^2 Evaluating the full conditional densities may be done in an MCMC approach where the posterior for ϑ_i is:

$$f(\vartheta_i = k|\mathbf{C}, \mu, \sigma) \propto \prod_{i \neq j} \exp\left\{-\frac{(C_{ij}-\mu_{k,\vartheta_j})}{2\sigma^2}\right\}. \tag{5.4}$$

The densities for μ_{k_1,k_2} are still conjugate and the required posteriors for the Metropolis algorithm is

$$f(\mu_{k_1,k_2}|\mathbf{C}, \sigma, \tau) \propto \prod_{i<j} \left(\exp\left\{-\frac{(C_{ij}-\mu_{\vartheta_j,\vartheta_i})^2}{2\sigma^2}\right\} \exp\left\{-\frac{\mu^2_{\vartheta_j,\vartheta_i}}{2\tau^2}\right\}\right)^{I(\vartheta_i=k_1,\vartheta_j=k_2)||I(\vartheta_i=k_2,\vartheta_j=k_1)}, \tag{5.5}$$

where $I(\vartheta_i = k_1, \vartheta_j = k_2)||I(\vartheta_i = k_2, \vartheta_j = k_1)$ is an indicator function. Therefore one only calculates the energy for μ_{k_1,k_2} by including those correlations where one or both of the variables belong to the actual group k. The posterior $f(\sigma^2|-)$ is given by (5.3).

Algorithm for QuickBCE While doing small scale experiments with the BCE algorithm it turned out that the mixing of the parameters was very slow. Since we were only interested in the grouping and not in the estimation of C_{ij} we designed a simpler version of the algorithm called QuickBCE. The algorithm is based on Gibbs Sampling and we initialise multiple chains and estimate all the model parameters in each chain iteratively.

The idea of the algorithm is to find the distributions $p(\vartheta_i = k)$. These distributions can then be used to quantify which variable i is in which group k. Since related variables will be in the same group we could use this information to build the variable blocks for the B-GTM.

The algorithm looks as follows:

- We sample each parameter by generating a random number h between 0 and 1. Then we generate a random number $\epsilon \sim N(0, STD)$ and update the state by $x_{new} = x_{old} + \epsilon$ and accept this state if $p(x_{old})/p(x_{new}) > h$. STD is the variable manipulating the step size and has to be chosen individually for all the parameters.

Create N chains using the following algorithm:

(a) **Initialise ϑ_i, μ_k and σ_k**

- $\mu_k \sim N(0, \tau)$ [Using $randn(0, \tau)$, where $randn$ is the command to sample from a normal distribution.]
- $\sigma \sim IG(\alpha, \beta)$ [Using $1./gamrnd(\alpha, \frac{1}{\beta})$], where α and β are chosen appropriately and $gamrnd$ is the command to sample from the gamma distribution.]
- ϑ_i chosen empirically to maximise (5.4) on the initialised C_{ij}, μ_k and σ
- Check that all initialised values, given the other values, have a probability bigger than 0 on their conditional density.

- Now we have sampled all variables for the model for the first time and in the next steps, we use Gibbs sampling, where further sampling has to satisfy the MCMC conditions to update the ϑ_i,μ_{k_1,k_2} and σ.

(b) Sample ϑ_i using (5.4).

(c) Sample μ_{k_1,k_2} and σ using (5.5) and (5.3) respectively.

(d) Check if converged, if not go to step (b).

The last step was done by visual inspection of the plots to check for mixing of the chains in our case. Some examples of the chains can be found in the appendix in Figure A.1 and A.2. This algorithm is just a prototype to demonstrate that there are semi-automated ways for obtaining the grouping of the variables. The algorithm is not fully automated since one still needs to specify the number of expected groups. Future research and experiments, which are out of the scope of this thesis, are needed to validate the algorithm and compare it against possible alternatives.

5.2 GTM Visualisation Space Reverse Mapping Initialisation (GTM-VSRMI)

This proposed novel approach to initialise GTM uses existing mappings of the data points in the visualisation space to initialise the Gaussian centres. This has the advantage of initialising the centres much closer to the data points than the conventional approach which initialises the centres along the first two principal components. In theory this should reduce the number of iterations needed to obtain an acceptable fit of the model and thus reduce computational costs. Another advantage is the possibility of using alternative mappings as demonstrated with GPLVM in chapter 4. This should broaden the possible applications of GTM and help to find better solutions. Especially in cases where PCA fails to capture the structure, for example the Swiss-roll data set, the usage of alternative initialisation for example by using Isomap should improve the results of the visualisation.

The method works as follows: first the projection which is chosen as initialisation is scaled to fit into the grid of nodes defined by (4.9), which normally lie between -1 and 1 on both axes in the visualisation space. Then for every node $\mathbf{x}_i$ in the grid, one finds the k-nearest data points whose projection is nearest $\mathbf{x}_i$. The mean in data space of these k-nearest data points is used as the centre of the Gaussian $\mathbf{y}_i$ corresponding to $\mathbf{x}_i$. An illustration on how this algorithm works is shown in Figure 5.2 and 5.3, where the Isomap projection of the S-shaped data is used together with the 5 nearest neighbours to initialise the node in the upper left corner of the grid.

The algorithm is as follows:

1: Initialise the grid of the visualisation space for GTM.
2: Scale the projection to be inside the grid of GTM.
3: Find the k-nearest data points for each node in the grid.
4: Run over all K nodes and use the mean of the position of these k data points to initialise every Gaussian $\mathbf{y}_i$ in the data space.
5: Solve the equation $\mathbf{Y} = \Phi(\mathbf{X})\mathbf{W}$ to compute $\mathbf{W}$ and fit the initialised Gaussians as closely as possible.

The fit of the manifold with respect to the initialised positions for the Gaussians depends highly on the number of RBFs. Since the number of RBFs controls the flexibility of the manifold the initialisation might still be suboptimal if the

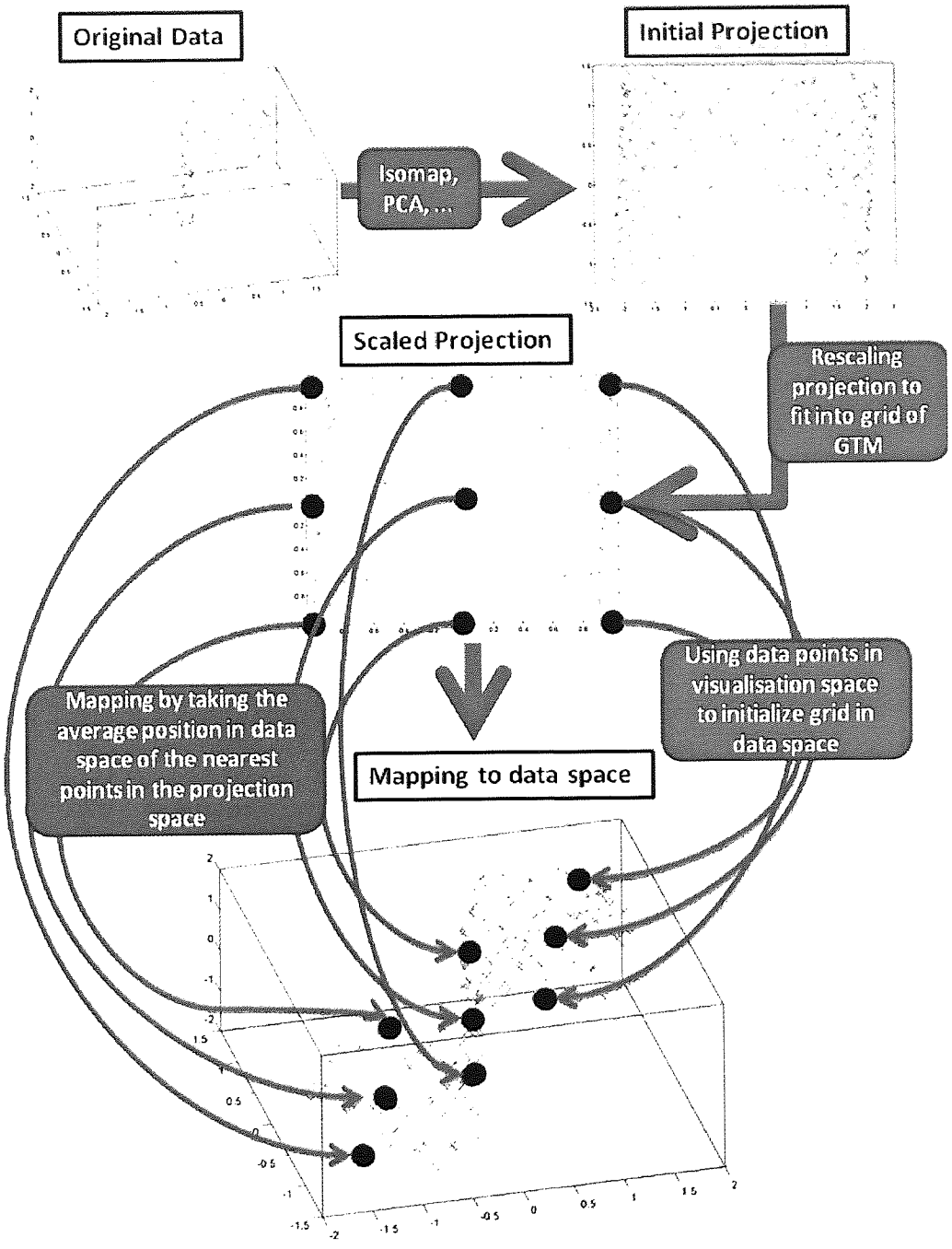


Figure 5.2: Schematic showing the high level design of the VSRMI algorithm.

number is chosen too small. If the manifold is not flexible enough it will not be possible to place all the Gaussians on their intended positions and one will end up with a bad regression estimate for the positions. However if the number is chosen too large one runs into the risk of over fitting the GTM to the data. This is not a problem when initialising the model. However during the training with the EM algorithm a too flexible GTM might overfit the data and the result will be poor generalisation.

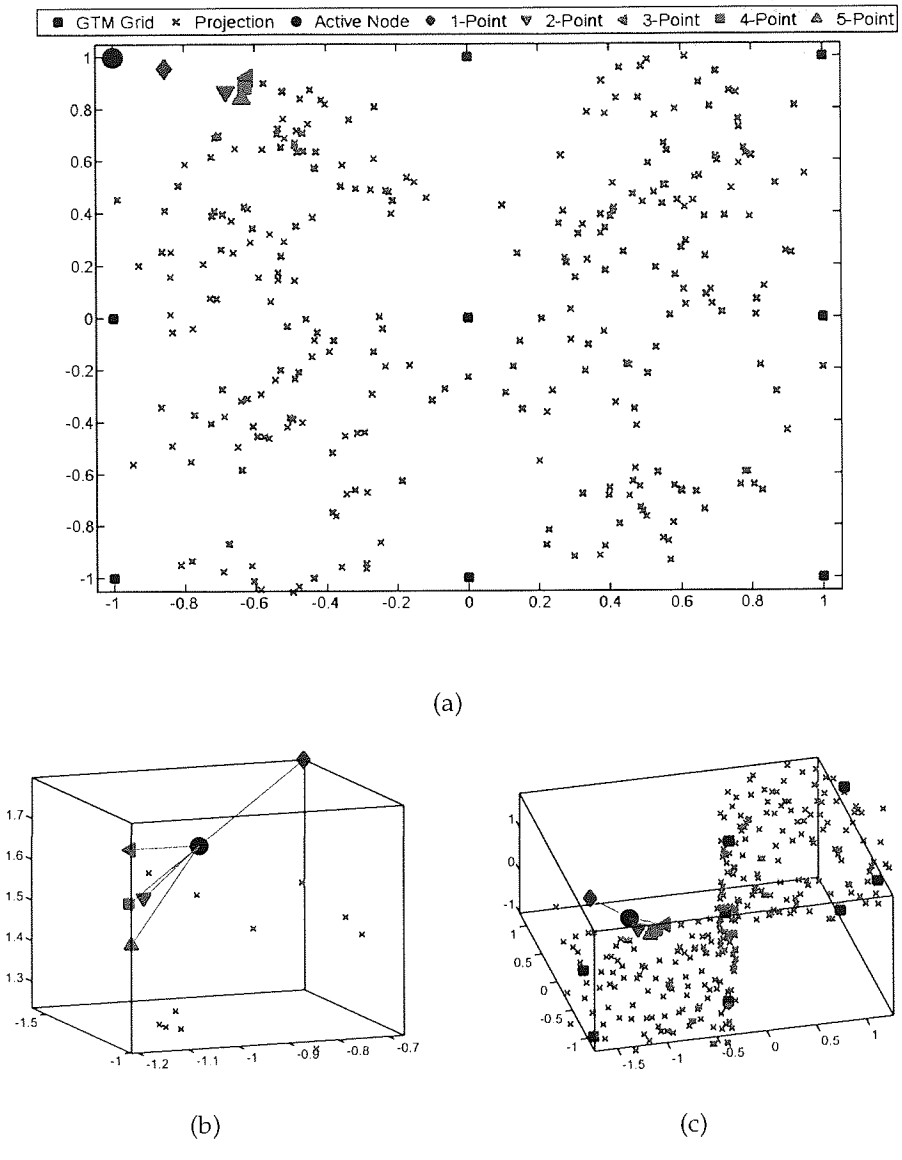


Figure 5.3: Schematic showing the low level design of the VSRMI algorithm: a) Active node and 5 closest points chosen for initialisation of the node. b) Placing of the active node in the data space through an appropriate choice of closest points. c) Overview for all nodes in the data space.

5.3 Assessing the novel Extensions

In this section the novel extensions B-GTM and VSRMI are evaluated. First the methodology for assessing the GTM is discussed in this section. Then possible advantages of B-GTM compared with a standard GTM (i.e. spherical GTM) are assessed. Then a quick demonstration of the possible uses for OLO and Quick-BCE is given. Finally the advantages of the VSRMI are demonstrated.

All algorithms based on GTM are examples of unsupervised learning and they always give a result when applied to a particular dataset. Thus we cannot tell *a priori* what is the expected or desired outcome. This makes it very difficult to

judge which method is the best (i.e. tells us the most about a certain dataset). However in the simple case of artificial data one can use prior knowledge about the structure of the data in the original space to quantify the error on the projection. This is exploited in some of the following measures for the quality of the projection:

Negative Log Likelihood (NLL): The negative log likelihood is a measure of the fit of a probabilistic model to a data set. In this thesis all quoted NLL values are computed on a test data set. The likelihood can be described as measure to assess the probability that the data were generated from the model given the actual parameters. Then one takes the logarithm of this quantity and negates it. Therefore the model which has the lowest negative likelihood fits the data best from a probabilistic point of view. However the magnitude of the measure is directly influenced by the dimensionality of the data. Thus we can not directly compare the performance of models with increasing or decreasing dimensionality of the data, but can compare different models on the same data. Another problem with the likelihood is the high dependence on the parameter for the covariance/variance in Gaussian distributions. In cases where this parameter converges to zero, which can happen because of singularities in the likelihood function or numerical errors, the likelihood will become very large (or small in the case of the negative log likelihood). This will indicate a very good model fit and thus can lead to misleading interpretations. It is therefore advisable to use the likelihood in conjunction with other measures to assess the fit of a model.

Nearest-Neighbour Label Error (NNLE): The nearest-neighbour label error can only be computed on labelled data, where we know the class of each data point. The idea is to consider the projected data and calculate for each point how many of the k nearest points are in the same class. Then we average the fraction of k-nearest neighbours in the same class over all the points. Finally we average over all the classes as well. Different values for k ranging from one to five were tried on all datasets and no big difference was apparent. Choosing bigger values on the small data sets that were used made no sense since this would result in looking at all possible neighbours which would defy the purpose of this measure. Therefore k was chosen to be $k = 3$.

Missing Data or Data Resampling (RMSE): Another evaluation method we have developed is driven by the capabilities of the models to estimate missing data. Re-estimating missing data can be seen as a re-sampling approach (Moeller and Radke, 2006; Yu, 2003) when the missing data patterns are created artificially and one retains the original value for comparison.

To benchmark the different methods against each other we are going to iteratively and component-wise delete every dimension $d = 1, \ldots, D$ from every point and see which estimates the model produce for the missing value. We then calculate the average root mean square error (RMSE) over all the points $n = 1, \ldots, N$,

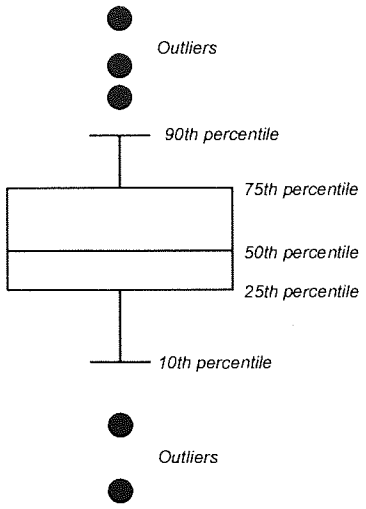


Figure 5.4: General schematic of a boxplot according to McGillan 1978.

where $\mathbf{y}$ are the original values and $\hat{\mathbf{y}}$ are the estimates:

$$RMSE = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{\sum_{d=1}^{D}(\mathbf{y}_{id} - \hat{\mathbf{y}}_{id})^2}{D}}$$

The behaviour of this measure is discussed in more detail in chapter 6. However in this chapter the model is trained and fitted on the complete data set. After the model is fitted to the data it is then used to estimate the missing values, using the theory from chapter 6. Therefore methodology is similar to leave-one-out cross-validation, but with the difference that the model is trained on the value which is later missing as well. This is done because the resulting model is only marginally different in the case where we have a complete data set and where one dimension of one data point is missing. However the GTM algorithm is coded and benchmarked in Matlab and the version of the algorithm which runs with complete data is 50-100 times faster than the version of the algorithm which needs to use indices to estimate the missing data.

To examine the results box plots are employed. A box plot (McGill *et al.*, 1978; MathWorks, 2009) (also known as a box-and-whisker diagram) is a convenient way of graphically depicting the distribution of numerical data through five-number summaries: sample minimum; lower quartile (25%); median (50%); upper quartile (75%); lower and upper outliers (Figure 5.4).

5.3.1 B-GTM

To evaluate the effectiveness of B-GTM we carried out comparative experiments with spherical (i.e. standard) GTM (S-GTM), full GTM (F-GTM) (i.e see (4.6)), and PCA. The data were sampled from a GTM with an 8×8 grid in the visualisation space. The grid was projected into a higher dimensional space using a

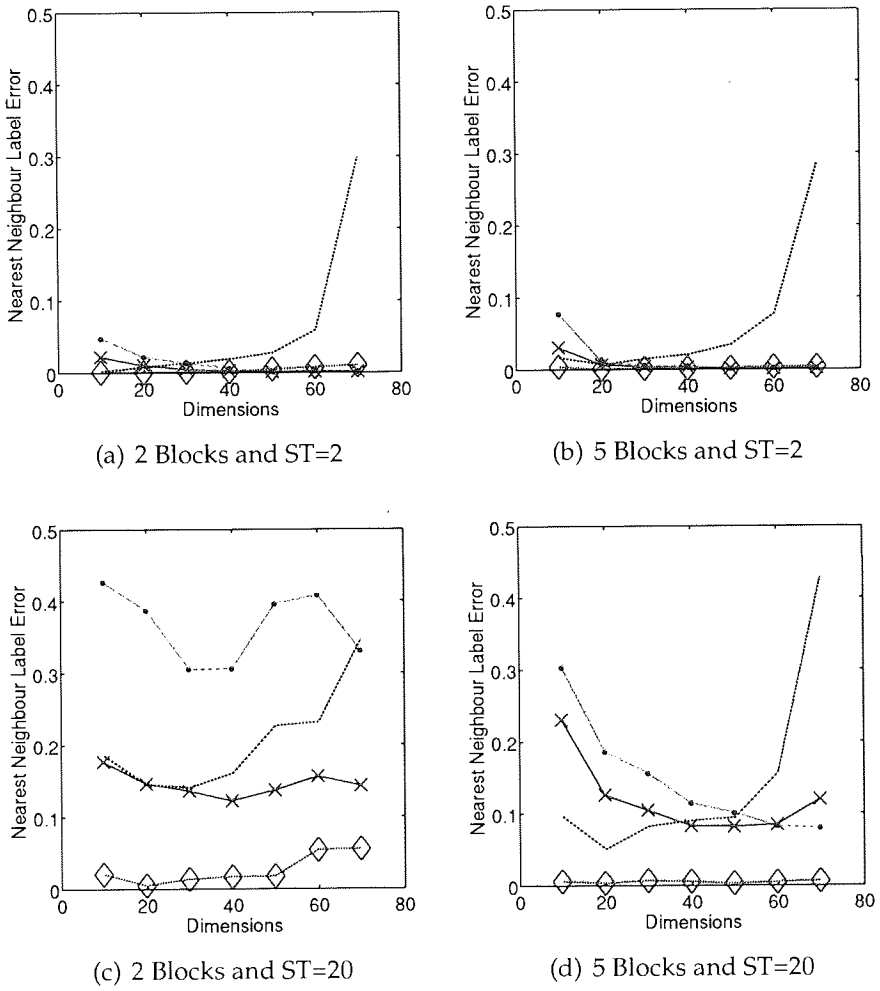
2×2 RBF network. The weights were randomly sampled from a normal distribution with zero mean and unit standard deviation. Since the RBF was chosen with random weights the restriction to a 2×2 RBF ensured a smooth and not overly erratic mapping. The GTM used to generate the data had a block diagonal covariance matrix and experiments were conducted with a range of levels of variance and correlation. The overall variance of the data varied from 6.45 to 7.55, with covariances around the single Gaussians varying from 2 to 20, denoted by ST, in Figures 5.5 to 5.11. The amount of ST controls the amount of structure in the data (i.e. the strength of the clustering in the covariance matrix). A low value for ST means no structure, while a high value means a lot of structure. In each experiment 100 data points were sampled from this GTM and each experiment was conducted 20 times, with a different randomly generated GTM each time.

Performance To calculate the NNLE the 8×8 grid was split into 4 classes with the 16 Gaussians in each corner of the grid being defined as one class. The results for this experiment, shown in Figure 5.5, indicate that in the case of little or no structure in the data the B-GTM performs as well as or only slightly worse than S-GTM or PCA, while F-GTM is clearly struggling with increasing dimensionality. This happens because the covariance matrix becomes non positive semi-definite as explained in section 5.1.1. In the case where more structure is present B-GTM clearly outperforms S-GTM and PCA, albeit once dimensionality increases the performance difference narrows. The difference in the number of blocks is significant as well since more blocks mean fewer parameters for the B-GTM model. The other models however benefit as well from more distinct blocks of variables. S-GTM profits because an increasing number of distinct blocks is closer to the initial assumption of a spherical covariance matrix (which has D blocks all of which share a common variance). The erratic behaviour of the measure seen in Figure 5.5(c) can be explained by the small sample size with different random RBF network mappings. 20 repetitions are not sufficient to obtain a smooth graph: however, with 20 repetitions the whole simulation took 2 days to run, and since the differences in model performance are quite big, the running of longer simulations was found to be unnecessary.

The RMSE was calculated on the same 20 projections to compare how B-GTM performs compared with S-GTM and F-GTM. To provide a baseline for comparison, mean imputation (MI) was also performed. This was done, not to benchmark GTM against MI, but to give an upper bound from an imputation that does not take account of any variable correlations or structure in the data. The results of this experiment, shown in Figure 5.6, indicate that both block and full GTM always outperform the S-GTM regardless of the amount of block structure in the data. However the amount by which the S-GTM is outperformed depends on the amount of block structure. Further there is no significant difference between the performance of block or the full version of GTM. This can be explained by the nature of imputation as a model validation technique which only assesses the fit of the model in the data space. For example, if the GTM is warped around itself

and thus gives poor results in the projection space, it may still give good imputation results if it properly covers the data cloud. This is also the reason why the RMSE does not show the breakdown of the F-GTM algorithm when the data dimensionality is large.

The problem of increasing dimensionality for B-GTM and F-GTM is also shown when looking at the negative log likelihood (NLL) of the models. Figures 5.7 and 5.8 show the NLL as box plots for $ST = 2$ and $ST = 20$ respectively. The box plots were carried out separately for 10, 40 and 70 dimensions because the NLL is not comparable across different numbers of dimensions. The O-GTM stands for the original GTM which generated the data and is intended as a comparison for the performance of the other models since it should, in the limit of large numbers of experiments , have the lowest possible NLL. The results for low structure (ST=2) show that F-GTM and B-GTM are always better than S-GTM in the case of lower dimensionality. However, for the 70-dimensional data set the spread of NLL over different repetitions massively widens. This indicates that the models find it harder to fit some of the generated datasets and thus indicates the breakdown of the algorithm of both B-GTM and F-GTM in the case of two blocks. In the case of five blocks B-GTM seems to be more stable while only F-GTM breaks down. The reason is likely to be the far more sparse nature of B-GTM in the case of five blocks. In the case of strong block structure the F-GTM breaks down with as few as 40 dimensions. This might happen because the stronger block structure will result in far bigger off-diagonal elements in the EM algorithm, which leads to poorly conditioned covariance matrices even in lower-dimensional spaces.



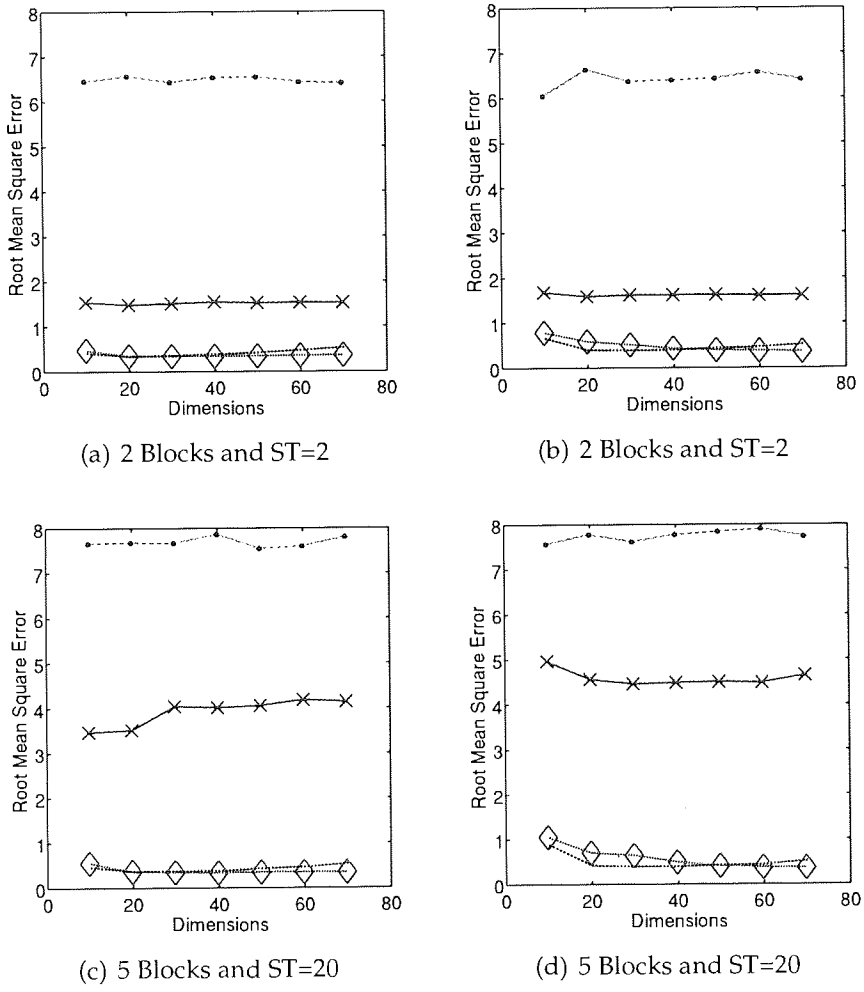
(a) 2 Blocks and ST=2

(b) 5 Blocks and ST=2

(c) 2 Blocks and ST=20

(d) 5 Blocks and ST=20

Figure 5.5: The nearest neighbour label error on the artificial test data with **high (ST=20) and low (ST=2) structure** for the GTM model with different covariance structures. S=Spherical, B=Block, F=Full GTM. PCA=(blue, dotted line with big dot). S-GTM=(green, constant line with X). B-GTM=(red, dashed line with diamond). F-GTM=(black, dashed and dotted line).



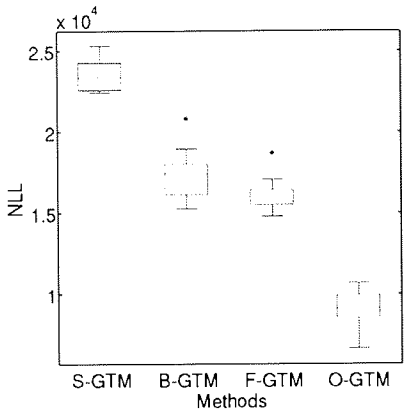
(a) 2 Blocks and ST=2

(b) 2 Blocks and ST=2

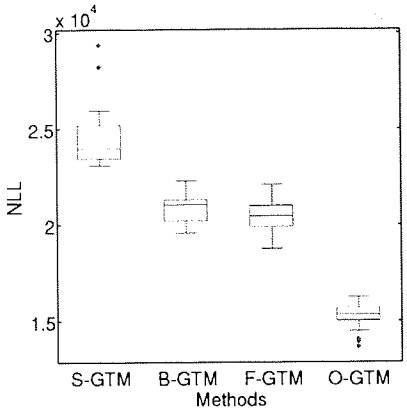
(c) 5 Blocks and ST=20

(d) 5 Blocks and ST=20

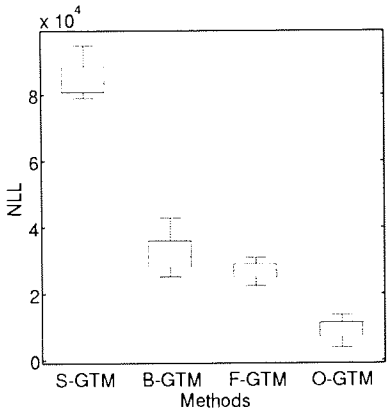
Figure 5.6: The root mean square error for imputation on the artificial test data with **high (ST=20) and low (ST=2) structure** for the GTM model with different covariance structures. S=Spherical, B=Block, F=Full GTM. MI=(blue, dotted line with big dot). S-GTM=(green, constant line with X). B-GTM=(red, dashed line with diamond). F-GTM=(black, dashed and dotted line.)



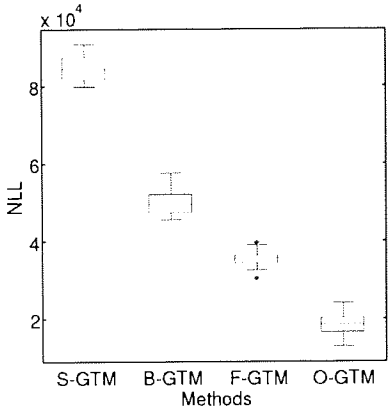
(a) 2 Blocks and 10 Dimensions



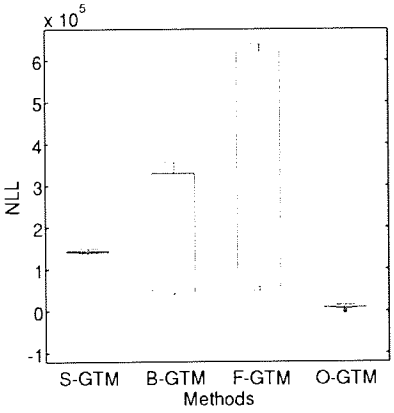
(b) 5 Blocks and 10 Dimensions



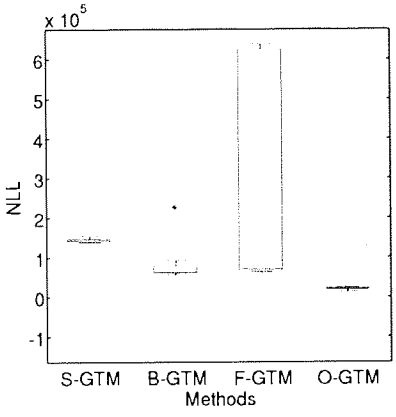
(c) 2 Blocks and 40 Dimensions



(d) 5 Blocks and 40 Dimensions

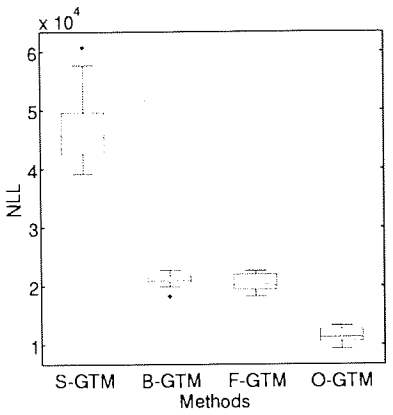


(e) 2 Blocks and 70 Dimensions

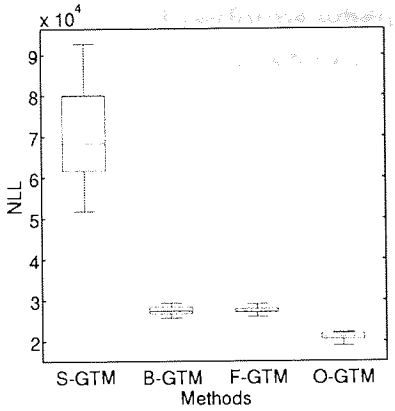


(f) 5 Blocks and 70 Dimensions

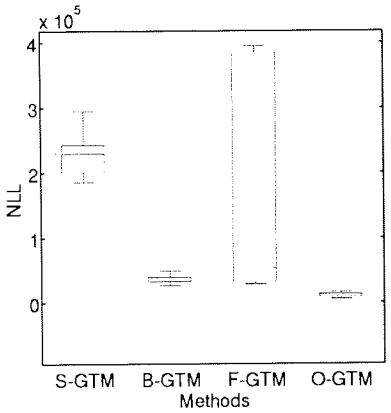
Figure 5.7: The negative log likelihood on the artificial test data with **low (ST=2) structure** for the GTM model with different covariance structures. The box plots show the variation of the negative log likelihood based on 100 different and randomly created datasets respectively for each combination of parameters (blocks and dimensions). S=Spherical, B=Block, F=Full and O=Original (thus creating) GTM. The very large box plots in the cases (e) and (f) show that the B-GTM and F-GTM, dependant on the number of blocks, are unstable and show incongruent behaviour with 70 dimensions.



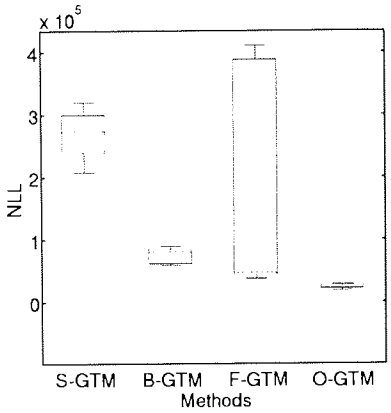
(a) 2 Blocks and 10 Dimensions



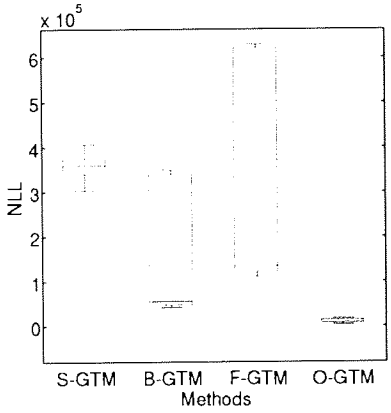
(b) 5 Blocks and 10 Dimensions



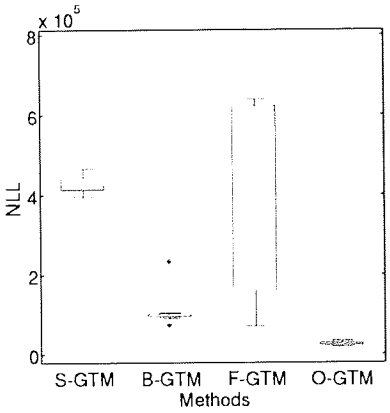
(c) 2 Blocks and 40 Dimensions



(d) 5 Blocks and 40 Dimensions



(e) 2 Blocks and 70 Dimensions



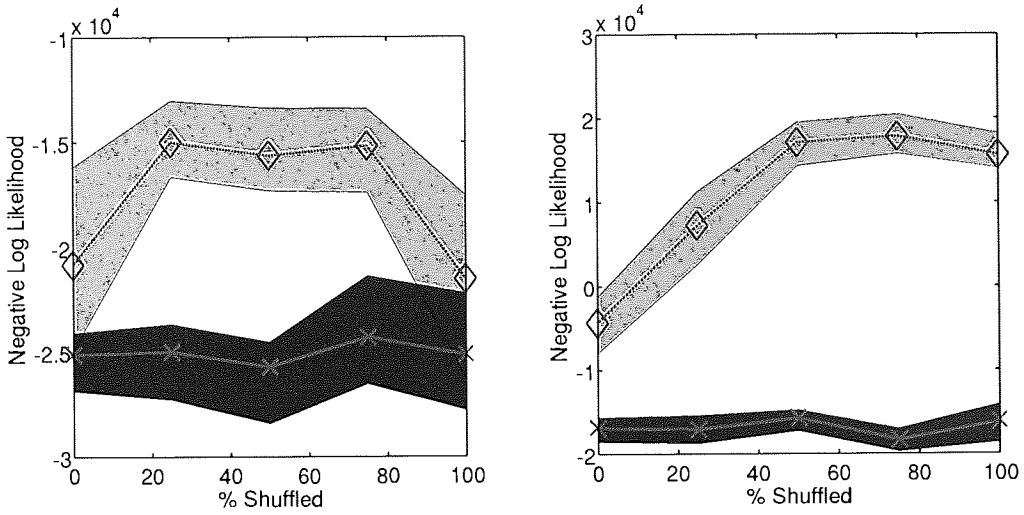
(f) 5 Blocks and 70 Dimensions

Figure 5.8: The negative log likelihood on the artificial test data with **high (ST=20) structure** for the GTM model with different covariance structures. The box plots show the variation of the negative log likelihood based on 100 different and randomly created datasets respectively for each combination of parameters (blocks and dimensions). S=Spherical, B=Block, F=Full GTM. The very large box plots in the cases (c),(d),(e) and (f) show that the B-GTM and F-GTM, dependant on the number of blocks and dimensions, are unstable and show incongruent behaviour with 40 and 70 dimensions respectively.

Block matrix dependency. To test how the B-GTM performs when the block structure is misspecified a shuffle experiment was conducted where a certain percentage of the variables were "shuffled" into another block. As in the previous experiment the data was randomly generated from a GTM. This was done 20 times for each level of shuffled block structure. The fraction of shuffled variables were selected to be 25%, 50%, 75% and 100%. For simplicity the other block was randomly selected. Therefore one has to keep in mind that variables of the same group might end up in the same group after being shuffled; in the case of just two groups this is always the case. Also in the case where 100% of two blocks are shuffled, all elements are now in the opposite group and thus this is equivalent to no shuffling, as can be seen from the results.

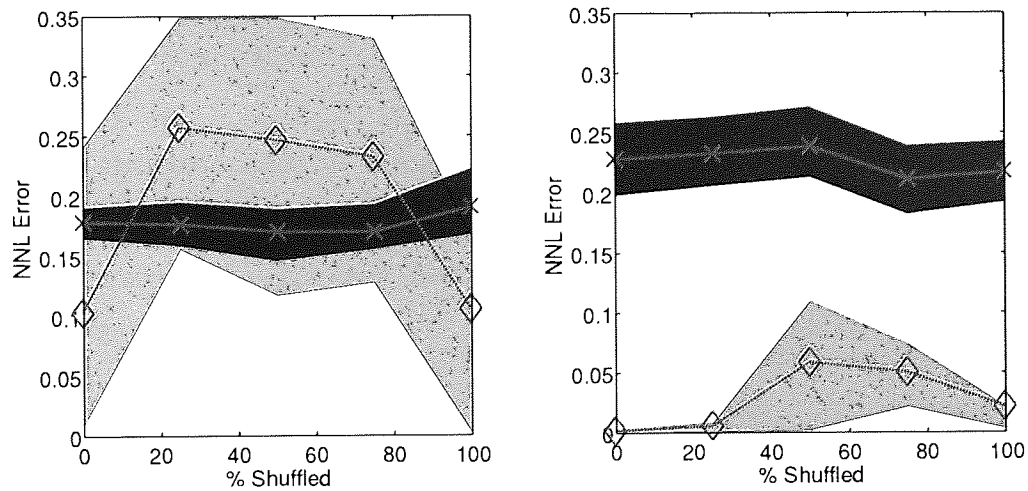
The shuffle experiment was conducted on a 30-dimensional highly structured data set with $ST = 20$ for the blocks in the single Gaussian and a standard deviation of 7.55 for the whole data set. Figures 5.9 to 5.11 show the results for two and five blocks. To compare the results the performance of B-GTM was plotted against the performance of S-GTM, which except for variations due to sampling should be stable because it is not influenced by the misspecification of the block structure. The performance of S-GTM stays constant as expected, only being altered by the random effects due to the small number of repetitions when rerunning the experiment with different random shuffle patterns.

The results on all three different measures NLL (Figure 5.9), NNL (Figure 5.10) and RMSE (Figure 5.11) show that the performance deteriorates if the block structure is misspecified. The example with two blocks shows that the consequences are quite severe if too much of the structure is misspecified. The NNL for B-GTM with wrong block structure in the case of two blocks is far worse than the NNL of the S-GTM. In the case of five blocks however the effect is not as strong. This effect presumably is related to the difference in amount of misspecified correlations. With two blocks there are more elements which can be misspecified and have an effect on the result than with five blocks, which is already a quite sparse matrix. The same effects can be seen when looking at the negative log likelihood (NLL), where the performance deteriorates in both cases. The effect is different when looking at the RMSE where the performance on only two blocks does not deteriorate a lot while the performance for five blocks strongly deteriorates. This is also corroborated when looking at the 75% confidence intervals of the performance, which increase especially when looking at the reusults for five blocks. The reason for this might be that the less sparse covariance matrix for B-GTM with two blocks still gives enough benefit to the imputation because the conditional mean is needed when calculating the missing values. A similar behaviour was observed for the RMSE where F-GTM also performed well on the RMSE while performing badly on the NLL and NNL.



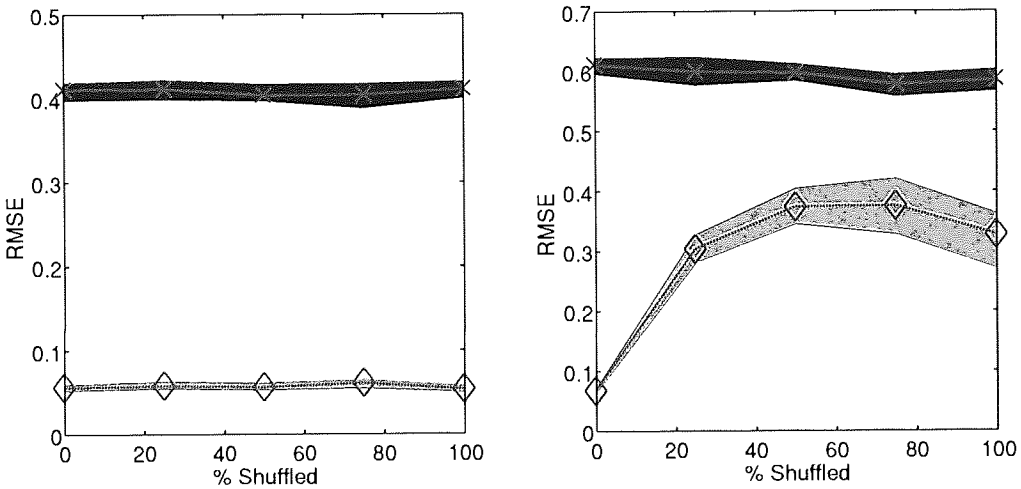
(a) 2 Blocks (b) 5 Blocks

Figure 5.9: The negative log likelihood on the artificial test data with **high (ST=20) structure** and 30 dimensions for the GTM model where different amounts of variables were *shuffled* into wrong groups. S-GTM=(green, constant line with X), B-GTM=(red, slashed line with diamond). 75% confidence intervall areas are marked by light gray (B-GTM) and dark gray (S-GTM).



(a) 2 Blocks (b) 5 Blocks

Figure 5.10: The nearest neighbour label on the artificial test data with **high (ST=20) structure** and 30 dimensions for the GTM model where different amounts of variables were *shuffled* into wrong groups. S-GTM=(green, constant line with X), B-GTM=(red, slashed line with diamond). 75% confidence intervall areas are marked by light gray (B-GTM) and dark gray (S-GTM).



(a) 2 Blocks (b) 5 Blocks

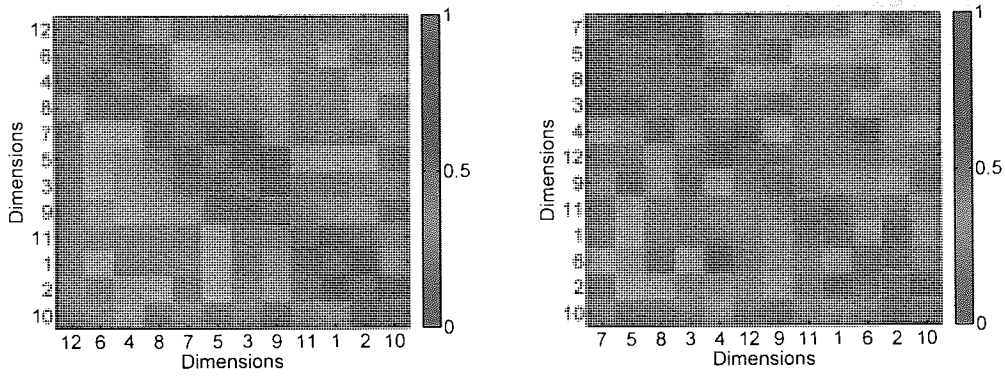
Figure 5.11: The root mean square error on the artificial test data with **high (ST=20) structure** and 30 dimensions for the GTM model where different amounts of variables were *shuffled* into wrong groups. S-GTM=(green, constant line with X), B-GTM=(red, slashed line with diamond). 75% confidence intervall areas are marked by light gray (B-GTM) and dark gray (S-GTM).

5.3.2 QuickBCE vs. OLO

Quantitative comparison of QuickBCE and OLO is complex since the algorithms are designed to perform different tasks. In essence the difference between the algorithms is that QuickBCE is trying to cluster the variables according to groups, while OLO is trying to sort the variables in a way that minimises the difference in correlations of neighbouring variables. Both algorithms can be used by the educated practitioner to identify possible groups of variables. This information can be used as means to get more insight about the data and/or as prior information for the B-GTM algorithm. It is certainly possible to enhance these algorithms to achieve both tasks however this is out of the scope of this thesis. Thus the comparison is done in a qualitative way and should be seen as an indicator for future research and applicability in geochemical applications.

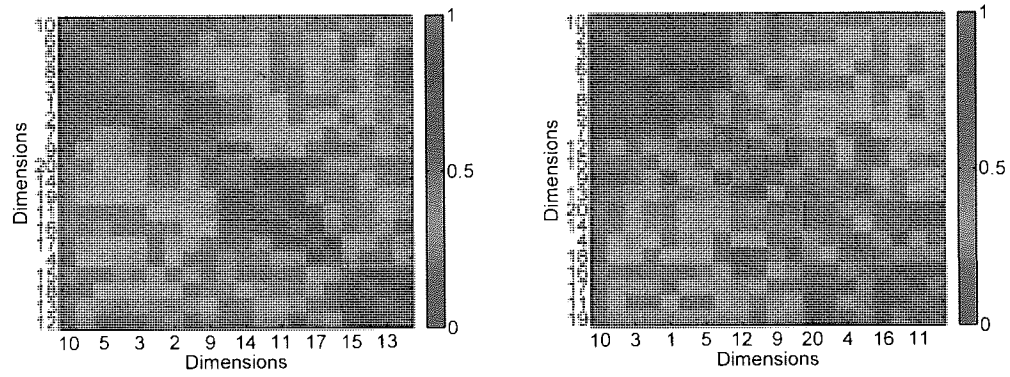
Since the QuickBCE algorithm is an MCMC sampling algorithm the results were post processed to make them comparable to OLO. The QuickBCE algorithm groups the variables but does not sort them. This requires post-processing of the results for the visual inspection in heat-maps. It was found that the post-processing of the results through PCA gave acceptable results. To do this a space was constructed where the samples were the original variables, which we want to sort. The values were the actual probabilities that the variable would fall into a certain class given by (5.4). Simply using PCA on this space and sorting the variables by their values on the first principal component lead to rough but meaningful and comparable heat-maps. However this was done only for the sake of comparing QuickBCE against OLO and further research and experiments in this area are needed. The QuickBCE algorithm was run for 31000 iterations, where the first 1000 samples were discarded allowing for burn-in. To check the mixing or convergence of the chains a visual inspection was undertaken (see Appendix A). One downside of the QuickBCE algorithm is that one needs to pre-specify the number of expected groups. In our case we only used the data sets with two clearly distinguishable groups and thus fixed this prior to two groups over all experiments.

To test the algorithms variables of the oil flow, 20D and 60D data sets from chapter three were shuffled into a random order. The results for the different data sets can be seen in Figure 5.12, 5.13 and 5.14 respectively. In all three cases the OLO algorithm performs very well at sorting the variables in a way that concentrates the high correlation coefficients towards the diagonal of the heat-map. This translates into an easy way to build groups based on obvious clusters of correlated variables. The QuickBCE algorithm is not producing such ordered heat-maps but most of the time succeeds in ordering the variables into distinguishable groups. In the first two heat-maps one can recognise two big clusters of variables which have very low correlations between each other and high correlations within each block. However in the 60D the block structure is not visible.



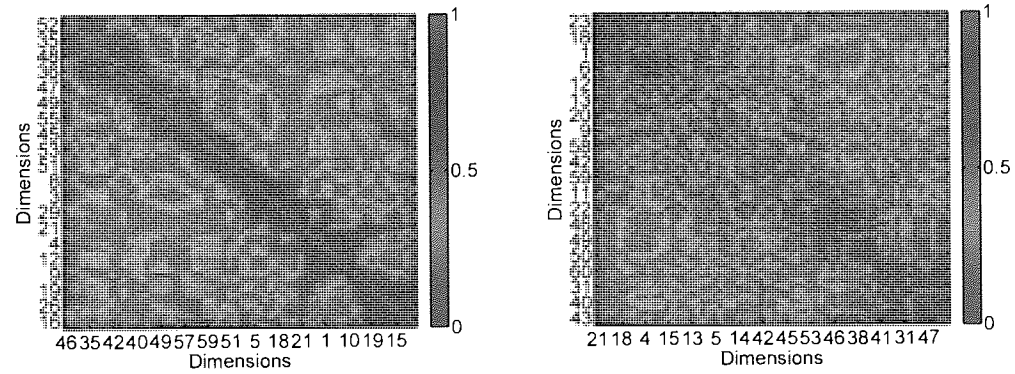
(a) (b)

Figure 5.12: The heat-maps of the correlation coefficients for the oil flow data. a) Sorting by the OLO algorithm. b) Sorting by using the grouping of the BCE algorithm.



(a) (b)

Figure 5.13: The heat-maps of the correlation coefficients for the 20D data. a) Sorting by the OLO algorithm. b) Sorting by using the grouping of the BCE algorithm.



(a) (b)

Figure 5.14: The heat-maps of the correlation coefficients for the 60D data.a) Sorting by the OLO algorithm. b) Sorting by using the grouping of the BCE algorithm.

5.3.3 GTM-VSRMI

To evaluate the performance of VSRMI we used the test data sets introduced in chapter 3. The experiment was conducted on all data sets and in all cases a spherical GTM (S-GTM) was fitted to the data. For completeness the results for B-GTM were included as well: however, this was just done to demonstrate that the results for GTM and B-GTM are similar. In all cases the EM algorithm was terminated after either 100 iterations or when the change log likelihood was less than 10^{-3}, whichever happened first. This was done to test if there are any speed gains through the usage of VSRMI. To measure the actual fit of the model the NLL, NNL and RMSE were used.

The widely used standard method to initialise the GTM was labelled *Old*. This method uses the axes of the first two principal components and initialises the grid of the GTM along these axes. Visually this can be imagined as placing the rubber sheet within the two dimensional hyperplane spanned by the first two principal components in the data space. We compared this with two variations of VSRMI in which PCA or Isomap were used to project the data to visualisation space. The results of the experiment can be seen in Table 5.1. The best results are highlighted in bold for GTM and B-GTM separately and it is apparent that in all cases the initialisation using VSRMI leads to a lower or similar NLL, NNL and RMSE. Additionally the runtime required to achieve these results is in most cases less than the conventional approach. Based on these results VSRMI can be recommended as the preferred way of initialising the GTM.

Using VSRMI and Isomap allows the user to compare a linear and non-linear initialisation. To do this one could use the initialised GTM, without training it. A possible approach would be to compare the likelihood, NNL and NLL of an untrained GTM initialised with PCA against one initialised with Isomap. If GTM performs radically differently with linear and non-linear initialisation one could use this as diagnostic to the degree of non-linearity of the data set. However more research and experiments in this area will be needed. The case of the D60 data in this experiment is special. In this case all models aborted after the first iteration, because there was no change when compared to the initial configuration. This problem is due to the EM algorithm which has problems of fitting the model in too high dimensional data. This problem will need to be investigated in further research and is not addressed in this thesis.

Data	Initialisation Method	GTM PCA Old	GTM PCA VSRMI	GTM Isomap VSRMI	B-GTM PCA VSRMI	B-GTM Isomap VSRMI
S Data	Runs	100	100	100	22	**21**
S Data	Likelihood	2264	2329	**2235**	2411	**2408**
S Data	NNL Error	0.068	0.082	**0.064**	0.019	**0.018**
S Data	RMSE	5.25	5.46	**5.22**	5.61	5.61
Swiss Data	Runs	100	100	100	22	**21**
Swiss Data	Likelihood	10207	10149	**9374**	10273	**9912**
Swiss Data	NNL Error	0.127	0.172	**0.071**	0.172	**0.031**
Swiss Data	RMSE	9.87	9.99	**9.67**	9.73	**9.72**
Oil Flow Data	Runs	100	100	100	100	**26**
Oil Flow Data	Likelihood	-4462	**-7495**	-5427	**-7365**	-5045
Oil Flow Data	NNL Error	0.026	**0.020**	0.031	**0.01**	0.013
Oil Flow Data	RMSE	0.09	**0.09**	0.094	**0.084**	0.102
BGTM D20	Runs	100	100	**21**	1	23
BGTM D20	Likelihood	3687	3800	**3301**	3604	**2627**
BGTM D20	NNL Error	0.273	0.29	**0.203**	0.62	**0.18**
BGTM D20	RMSE	1.03	1.00	**0.82**	0.38	**0.26**
BGTM D60	Runs	1	1	1	1	1
BGTM D60	Likelihood	Inf	15196	13963	3604	3604
BGTM D60	NNL Error	0.223	0.26	0.18	0.693	0.76
BGTM D60	RMSE	2.17	1.51	1.39	0.23	0.23

Table 5.1: Results after training GTM with different initialisations. The best results for each model type are marked in bold. The results for the BGTM D60 data are included just for completeness since all algorithms broke down when processing them as can be seen by the number of iterations. The results indicate that in all cases the VSRMI initialisation is superior to the old initialisation using the first two principal components of the data. The difference in the results is very relative and thus the important aspect of the table is the clear trend, which is in favour of the VSRMI.

5.4 Summary

The experiments show that B-GTM is very promising. Given the right block structure the algorithm performs equally well or better than S-GTM and F-GTM, depending on how strongly the block structure is exhibited in the data. However, if the block structure is misspecified then the B-GTM performance deteriorates quite rapidly and thus special care should be taken when specifying it.

We also noted a limitation of GTM in that it did not perform well on data sets in high-dimensional spaces (depending on the data set between 50 and 80 dimensions), a problem that has not previously been reported. This problem is inherent to the EM algorithm which apparently struggles with increasing amount of dimensions. However more research into the properties of the EM algorithm when combined with GTM is needed to draw conclusions. Using additional heuristics this limit can be extended as is demonstrated in the case of B-GTM where heuristics are needed because additionally to the problems with the EM algorithm one runs into numerical problems when using a Gaussian noise model. If one wants to analyse higher dimensional data the issue with the EM algorithm may be fixed by using alternative ways of minimising the likelihood. However in addition to numerical problems and the singularities in the likelihood there is a general problem with employing density models in very high dimensional spaces since larger sample sizes are needed to draw valid conclusions.

To specify the block structure one can either elicit the needed information from experts or analyse the data with additional tools. Possible utilities to help with the specification of the block structure are OLO and QuickBCE. OLO in general is well suited for the guided splitting of variables into groups. Ideally this would be done by an expert practitioner who could identify the groupings in the variables based on his experience and the results produced by OLO. This would be the ideal approach however if this cannot be done an algorithm similar to QuickBCE could be developed for an automatic classification of groups of variables. The QuickBCE algorithm is not fully automated, one needs to specify the number of groups, and needs more development and validation. Further it is an MCMC algorithm, which implies a long runtime before one reaches a decision. While the OLO algorithm performed the needed calculations in a matter of seconds the QuickBCE algorithm took more than 15 minutes on the tested data sets.

A very useful addition to the GTM algorithm is the VSRMI. It allows GTM to exploit the advantage of local methods like Isomap or other alternative initialisations. The results clearly show the benefits of this approach especially with more complicated data sets where PCA is not sufficient to pick up all the structure in the data. This extension will result in more flexibility for the practitioner and might even make GTM a tool to test the non-linearity of a data set. To do this one could use the initialised GTM, without training it. A possible approach would be to compare the likelihood, NNL and NLL of an untrained GTM initialised with PCA against one initialised with Isomap. The reasoning is that if the dataset is linear the difference between Isomap and PCA should be marginal, however if the dataset is non-linear and Isomap can capture the structure the results produced

by Isomap should show a considerable advantage on all measures.

6 Missing Data

CONTENTS

6.1 Single Imputation 109
6.1.1 Mean Imputation (MI) 109
6.1.2 Weighted Mean Imputation (WMI) 110
6.1.3 Sequential Multiple Regression Imputation (SRI) 110
6.1.4 Multiple Regression Imputation with Mean initialisation and Correlation Cut (MRI) 111
6.1.5 Probabilistic PCA with Missing Data (Bayesian PCA or BPCA) 112
6.1.6 EM for Missing Data in Mixture Models 113
6.1.7 EM for Missing Data in Gaussian Mixture Models 113
6.1.8 Extension of GTM for Missing Data Imputation using EM (GTMI) 115
6.1.9 Extension of B-GTM for Missing Data Imputation using EM (B-GTMI) 117
6.1.10 Performance Indicator 118
6.1.11 General behaviour of the performance indicator and the imputation methods 118
6.1.12 Benchmark 122
6.1.13 Projection Results 122
6.2 Missing data as a way to assess the model fit in unsupervised learning 124
6.3 Conclusion 127

Missing data represent a general problem in many scientific fields (Latini and Passerini, 2004) and are critical in environments like geochemistry where one has small data sets with valuable, and expensive to obtain, samples. Usually the missing data should not be ignored but most analysis tools cannot cope with them. Therefore the practitioner has either to delete the incomplete samples, which might lead to a serious bias, delete the incomplete variables which might greatly impair the analysis or use a data imputation approach to infer the missing values.

In the case of geochemistry the main causes of missing data are:

- Different analysis methods for different kind of samples (gas / rock / oil)
- Absence of complete analysis for certain samples for financial reasons or other constraints like time
- Flawed analysis of the sample (human or technical errors)
- Polluted or contaminated samples
- Missing entries while digitising or storing the data in the computer

In general we assume that the data set $\mathbf{T} = \mathbf{y}_1, ..., \mathbf{y}_N$ can be divided into an observed component $\mathbf{T}^o$ and a missing component $\mathbf{T}^m$. Every point $\mathbf{y}_n = [\mathbf{y}_n^o, \mathbf{y}_n^m]$ can be split into an observed and a missing component. Assuming a missing indicator matrix $\mathbf{M} = (M_{ij})$, the missing-data mechanism can be characterised by the conditional distribution of $\mathbf{M}$ given $\mathbf{T}$, $p(\mathbf{M}|\mathbf{T}, \theta)$, with θ being an unknown parameter vector. Given this formulation one can distinguish between three types of missing data (Latini and Passerini, 2004; Little and Rubin, 2002; Schafer, 1997; Scheffer, 2002):

- Missing completely at random (MCAR)

$$p(\mathbf{M}|\mathbf{T}, \theta) = p(\mathbf{M}|\theta) \, ,$$

  if the missing data depend only on the unknown set of parameters θ. This is a very stringent condition since the missing-data mechanism does not depend on the variable of interest or any other variable in the data set. Missing data are very rarely MCAR however this condition is required in order for case deletion to be valid (Rubin, 1976).

- Missing at random (MAR)

$$p(\mathbf{M}|\mathbf{T}, \theta) = p(\mathbf{M}|\mathbf{T}^o, \theta) \, .$$

  The term missing at random is slightly misleading. The missing-data mechanism in this case is not conditional on the values that are missing however it is conditional on other observed values in the data set.

- Not missing at random (NMAR)

$$p(\mathbf{M}|\mathbf{T},\theta) = p(\mathbf{M}|\mathbf{T},\theta)\,,$$

in this case the missing-data mechanism is conditional on the observed and missing data. This is the hardest condition to model and will not be discussed in this thesis.

Most existing imputation methods are based on statistical moments and estimation equations and give unbiased results with MCAR data, while likelihood methods also give unbiased estimates with MAR data (Little and Rubin, 2002). Without sufficient knowledge, which would allow to model the missing data mechanism, there is no possibility to engineer an unbiased approach to deal with NMAR data.

There are different standard methods to deal with missing data:

- The simplest approach is called Complete-Case analysis (Rubin, 1976) and confines attention to only those cases where all variables are available. The advantage is that one can use all the standard statistical analyses without modification. The disadvantage is that one wastes a lot of information and in addition, if the MCAR assumption does not hold, a bias will be introduced to subsequent analysis or parameter estimation.

- Another approach is the Available-Case (Little and Rubin, 2002) analysis where every variable is treated differently and one uses all the information for each variable to estimate, for example, statistical parameters. The advantage here is that one uses information from the incomplete cases but the disadvantage is that there are now different sample sizes for each variable. This makes analysis with more sophisticated methods quite complex and in addition it also has problems with bias and comparability across variables if the MCAR assumption does not hold.

There are many alternative approaches to deal with missing data. These utilise a broad range of ideas and theories. They cover a diverse field and are usually optimised to fit the needs and available information in a particular area. This makes the classification of these algorithms quite complicated but generally it is possible to differentiate between multivariate and univariate approaches. The univariate approaches are very simple and do not take into account the relationship between different variables. Algorithms in this class are:

Univariate:

- Mean Imputation (Little and Rubin, 2002).

- Hot deck imputation, which is the random drawing from observed values (Ford, 1983; Song and Shepperd, 2007).

The class of multivariate approaches can be split into 3 categories by looking at the kind of information they take into account: *Local*, *Regression* and *Structured*. Local algorithms estimate the missing data by using a local metric on the observed dimensions to determine which data points are close to the data point with the missing value. They then use the data points in close proximity to infer the missing values. Regression algorithms treat the missing data points as target values and restate the missing data problem as a regression problem. The structured class consists of a variety of more complex algorithms. These algorithms take more information about the structure of the data into account. Examples of the algorithms are:

Local

- Nearest neighbour imputation in genetics (Troyanskaya *et al.*, 2001).
- Local least squares imputation in genetics and bioinformatics (Kim *et al.*, 2005; Brás and Menezes, 2006).

Regression

- PCA or PLS utilising the EM or NIPALS algorithm in chemometrics. (Wold, 1987; Rannar *et al.*, 1995; Nelson *et al.*, 1996; Nguyen and Rocke, 2004; Kettaneh *et al.*, 2005; Andersson and Bro, 2000).
- Sequential regression imputation in survey design (Raghunathan *et al.*, 2001).
- Modification of kernel PCA algorithm to deal with missing data (Sanguinetti and Lawrence, 2006).

Structured

- Conditional mean imputation in speech recognition (Cooke *et al.*, 2001).
- Bayesian PCA in genetics (Oba *et al.*, 2003).
- Structural equation modelling in social science (Olinsky *et al.*, 2003).
- Neural networks in machine learning (Tresp *et al.*, 1994; Lakshminarayan *et al.*, 1996)

The different algorithms are all based on different assumptions and work well on different data sets. An algorithm relying on local information will do better in densely populated data sets, while algorithms relying on the regression approach will do well in very linear and highly correlated data sets. The structured class of algorithms is harder to classify because they use a variety of information and correct the estimates by structural information about the data. For example in the case of Bayesian PCA or conditional mean imputation this is done by correcting the regression or mean estimate respectively by including information about the

sample covariance matrix, i.e. the distribution of the data. Generally this class of algorithms can deal with more complex data sets. However the price is a higher computational cost, which in some cases can be substantial.

In the following chapters only a fraction of these different methods for the imputation of missing values will presented. The methods presented are chosen to represent a broad class of frequently used imputation methods using local weighting, regression and a Bayesian modelling approaches. These algorithms will be compared with GTM. The GTM algorithm itself already has a proven track record for the use with missing data and it has been benchmarked previously to test its capabilities to deal with missing data (Sun, 2002; Vicente *et al.*, 2004; Olier and Vellido, 2005; Schroeder *et al.*, 2008).

Each algorithm is introduced and motivated and the advantages and drawbacks are discussed. At the end of this chapter there is a benchmark study to compare how the block GTM extension performs against the other imputation methods as well as spherical GTM.

6.1 Single Imputation

Single imputation methods (Little and Rubin, 2002) are the most common and easy to use imputation methods. They calculate single estimates for the missing data without estimating the uncertainty that exists in these estimates (i.e. the estimated variance). The imputation methods presented are thought to be representative for most classes of imputation methods (mean, regression, likelihood) while still giving stable results for data sets with up to 60 percent missing data. Methods are defined as stable if they consistently perform better than Mean Imputation (the most simple imputation method).

6.1.1 Mean Imputation (MI)

In this very simple approach (Little and Rubin, 2002) the missing values are replaced by the mean of the known values

$$\hat{\mathbf{y}}_n(m) = \frac{1}{N^o} \sum_{j=1}^{N^o} \mathbf{y}_j^o(m) \, ,$$

with N^o the number of observed components for the variable in question and m the index for the missing dimension for the point t_n.
This method suffers from a number of drawbacks which can be illustrated with the following MAR example:
Assume we have 4 patients whose height x we measure in cm and weight y in kg if the patient is under 190 cm. This results in an incomplete data set $[\mathbf{x}, \mathbf{y}]$
$\mathbf{x} = [55, 60, 63, m]$, where $m = 90kg$
$\mathbf{y} = [170, 173, 172, 193]$
Using the mean imputation we would impute $m = 59.4$ which gives rise to the

following problems.

- A serious bias is introduced in the produced results since the MCAR assumption does not hold true: $E[\mathbf{x}] = 59.4$ *while the true value is 67, given that the true value for* $x_4 = 90kg$.
- Estimation of the covariance matrix is biased *since we are reducing the correlation between the variables.*
- An estimate of the variance is too small *since we are reducing the values which deviate from the mean.*

It can also be noted that the approach is clearly inappropriate for categorical variables, although here a median value might be the equivalent.

6.1.2 Weighted Mean Imputation (WMI)

The weighted mean imputation is motivated by HotDeck Imputation (Ford, 1983) and KNN-based imputation in bioinformatics (Troyanskaya *et al.*, 2001). It was developed to have a benchmark for the performance of algorithms using the local structure in the data space for imputation. Originally the KNN-based imputation was intended as the method of choice however it proved to be very unstable with large amounts of missing data and thus WMI was developed. The basic idea is to use the Euclidean norm as an inverse weight and build a weighted mean for every missing value based on the closest data points. The algorithm itself is therefore relatively simple:

1: Perform a mean imputation to create the complete estimated data set $\hat{\mathbf{Y}}$.
2: Compute the Euclidean distance between all the data points in $\hat{\mathbf{Y}}$.
3: Impute the missing components of $\mathbf{y}_n$ by calculating the average over all the data points in $\mathbf{Y}$ which observed these components with the inverse distance to $\mathbf{y}_n$ as weight.

The algorithm exploits the local structure of the data space and works well in densely populated areas of the data space while only doing as well as mean imputation in sparsely populated areas of the data space.

6.1.3 Sequential Multiple Regression Imputation (SRI)

Multiple Regression in general is used to approximate the linear relation between multiple variables in a data set $\mathbf{Y}$. The underlying assumptions is that the values of one variable can be obtained through a linear combination of the others:

$$\mathbf{y}_i \approx a_0^i + a_1^i \mathbf{y}_1 + .. + a_{i-1}^i \mathbf{y}_{i-1} + a_i^i \mathbf{y}_{i+1} + .. + a_{d-1}^i \mathbf{y}_d \,.$$

Sequential Multiple Regression Imputation (SRI) (Raghunathan *et al.*, 2001) was introduced for handling missing data in surveys. We tested a simplified version using only linear multiple regression since the data we focus on are usually continuous rather than discrete. The algorithm is:

Part 1:

1: Order the variables $\mathbf{y}_{1:d}$ by the number of missing values $\hat{\mathbf{y}}_{1:d}$; least first.
2: Impute any missing values in $\hat{\mathbf{y}}_1$ with mean imputation.
3: Iteratively go over all variables with missing values (starting with the one which has the least amount) and estimate the regression factors for the complete variables. The use the regression factors to estimate the missing values by treating them as target values $\mathbf{y}_i$. Once the values for one variable are estimated treat this variable as complete and include it in the regression estimation of the next variable.

Part 2:

1: Estimate the coefficients $\mathbf{a}^j = [a_0^j, ..., a_{d-1}^j]$ of the linear regression model for all variables.
2: Use these to re-estimate the missing values.
3: Assess whether the algorithm has converged; if not go to step 1 (Part 2)

This algorithm exploits the linear structure in the data but in general is vulnerable to outliers. Furthermore, the initialisation in Part 1 is relatively important since it presumes a linear relationship between all the variables. However we found that the algorithm became highly unstable once data were missing across most of the variables and ultimately started to break down with large proportions of missing data. Thus we modified the algorithm as will be explained in the next section.

6.1.4 Multiple Regression Imputation with Mean initialisation and Correlation Cut (MRI)

The SRI algorithm was designed for data sets where only a minority of the columns have missing data and all the variables have a linear relation. This assumption may be true for surveys, where missing values are mostly due to people who do not wish to answer certain questions, but in geochemistry one may experience missing data in almost all the columns and the variables might not be related in a linear sense or might have no relation at all. This led to problems on some of the *real data* sets we have used.

To have a more stable linear imputation than the SRI we created a modified multiple regression imputation. This method differers from the SRI because it is initialised with mean imputation to use the complete data matrix and make it more robust against outliers and we also use the correlation coefficient between the dimensions as measure to indicate the strength linear relationships between two dimensions. In the case of MRI this estimate was used to determine if two dimensions are highly enough linearly related to be used for calculation in the regression models.

The result is a combination of SRI and MI, which can deal even with large amounts of missing data and only performs slightly worse than the original SRI when small amounts are missing. At first one constructs a complete data set $\mathbf{C}$ using mean imputation on the incomplete data set $\mathbf{Y}$. Then one learns the regression factors $\mathbf{a}^j$ on this data set and after this one uses these regression factors

to re-estimate the missing values in **T**. Therefore the algorithm can be described as follows:

1: Perform a mean imputation to create a complete data set **C**, keeping track of the missing value locations.
2: Compute the correlation coefficient between the variables on the complete data set **C**.
3: Estimate the regression factors $\mathbf{a}^j = [a_0^j, ..., a_{d-1}^j]$ on the dimensions where the correlation coefficient is sufficiently high for stability.
4: Use the regression model to recompute the missing values and create a new complete data set **C**.
5: Check that none of the imputed values is outside the range of the known values. *(This sanity check restricts the prediction range but avoids the generation of heavy outliers. It is required because even when only including well correlated variables the linear multiple regression can become unstable when large numbers of values are missing.).*
6: Assess whether the algorithm has converged (we use a threshold on the sum of absolute change across the complete data matrix); if not go to step 2.

This algorithm exploits the linear structures in the data while still being able to cope with a large amount of missing data, though one would expect problems and poor predictions if the proportion of missing data is too high (more than 50 percent missing) or the variables exhibit non-linear relationships between each other.

6.1.5 Probabilistic PCA with Missing Data (Bayesian PCA or BPCA)

The probabilistic formulation of PCA means that it can be extended to deal with missing data. It was extended as well as termed "Bayesian PCA" by Oba *et al.* (2003) to help with the estimation of missing values in bioinformatics. The algorithm performs similarly to or better than PCA with non-linear iterative partial least squares (NIPALS) (Nelson *et al.*, 1996; Wold, 1987; Andersson and Bro, 2000), which is an alternative approach to use PCA with missing values, and only slightly worse than partial least squares imputation (Kettaneh *et al.*, 2005) and local least squares imputation (Kim *et al.*, 2005), according to a study of Brás and Menezes (2006). The algorithm is therefore a good benchmark to summarise the performance of this class of algorithms.

The model is the same as outlined in appendix B.2; however Oba employed a variational Bayes (VB) algorithm (Attias and Ar, 1999) to optimise (B.4). A schematic description of the algorithm is as follows:

1: The posterior distribution of the missing values is initialised by imputing the dimension average (Mean Imputation).
2: The parameters for the likelihood in equation (B.4) are estimated given the observed data and the posterior of the missing data (through the usage of VB there are hyperparameters which are also estimated).
3: The posterior of the missing data is estimated given the parameters and hy-

perparameters.

4: Assess whether the algorithm has converged; if not go to step 2.

In principle this algorithm should perform extremely well in data sets with strong linear dependencies between the variables; however through the usage of the conditional mean one would expect superior results to MRI since the method also takes into account the covariance structure of the data.

Oba made the Matlab code for his algorithm freely available from his website [1]. There is also a Java version and an R-version available. This was greatly appreciated by the author of this thesis and the code was used without major modifications.

6.1.6 EM for Missing Data in Mixture Models

The EM algorithm can naturally be extended to deal with missing data and be incorporated into a mixture model (Ghahramani and Jordan, 1994). The algorithm needs to be modified in the E-Step:

$$Q(\theta, \theta^{i-1}) = E[p(\mathbf{Z}|\mathbf{T}^o, \mathbf{T}^m, \theta^{i-1}) \ln p(\mathbf{T}^o|\theta)] ,$$

where the expected value is taken with respect to both sets of missing variables, the missing values $\mathbf{T}^m$ and the missing indicators $\mathbf{Z}$, where $z_{kn} = 1$ if and only if t_n is generated by component k, otherwise $z_{kn} = 0$. An example of this is given in the following paragraph where we use a mixture of K Gaussians in the case of the GMM and the GTM.

6.1.7 EM for Missing Data in Gaussian Mixture Models

As in the general case we use the formulation of (Ghahramani and Jordan, 1994) to deal with missing values by using the EM algorithm.

The expectation of the error function $\langle -L_{comp} \rangle$ given by (4.2) can be written as

$$-L(\theta)_{comp} = -\sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn} \ln P(k) - \sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn} \ln\{\alpha_k p(\mathbf{y}_n|\theta_k)\} . \quad (6.1)$$

Since we are only interested in maximising the posterior probability $p(k|\mathbf{y}_n)$ given by equation (4.4) we are going to neglect the first term in the following paragraphs.

Full Covariance Matrix

For a GMM with a full covariance matrix we can expand the second term of equation (6.1) to:

$$-L(\theta)_{comp} = -\sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn}[\ln \alpha_k + \tfrac{1}{2} \ln |\Sigma_l| + \tfrac{D}{2} \ln 2\pi , \quad (6.2)$$

[1] http://hawaii.aist-nara.ac.jp/~shige-o/tools/BPCAFill.html

$$+\tfrac{1}{2}(\mathbf{y}_n^o - \mu_k^o)^T \Sigma_k^{-1,oo}(\mathbf{y}_n^o - \mu_k^o)\ ,$$
$$+(\mathbf{y}_n^o - \mu_k^o)^T \Sigma_k^{-1,om}(\mathbf{y}_n^m - \mu_k^m\ ,)$$
$$+\tfrac{1}{2}(\mathbf{y}_n^m - \mu_k^m)^T \Sigma_k^{-1,mm}(\mathbf{y}_n^m - \mu_k^m)]\ ,$$

where μ_k and Σ_k denote the means and covariance of the kth Gaussian respectively. A more detailed derivation can be found in the work of Ghahramani and Jordan (1994). Among the superscripts, for example, $(-1, oo)$ denotes inverse followed by submatrix operations where Σ_k is divided into $\begin{pmatrix} \Sigma_k^{oo} & 0 \\ 0 & \Sigma_k^{mm} \end{pmatrix}$ corresponding to $\mathbf{y} = \begin{pmatrix} \mathbf{y}^o \\ \mathbf{y}^m \end{pmatrix}$. Taking the expectation with respect to both sets of missing variables results in the three unknown terms z_{kn}, $z_{kn}\mathbf{y_n^m}$ and $z_{kn}\mathbf{y_n^m}\mathbf{y_n^m}^T$. To calculate these terms one has to introduce the variables $\hat{\mathbf{y}}_{kn}^m$

$$\hat{\mathbf{y}}_{kn}^m \equiv \langle \mathbf{y}_n^m | z_{kn} = 1, \mathbf{y}_n^o, \boldsymbol{\theta}_k \rangle = (\mu_k^m) + \Sigma_k^{mo}\Sigma_k^{-1,oo}(\mathbf{y}_n^o - \mu_k^o)\ , \tag{6.3}$$

which is the least-squares linear regression between $\mathbf{y}_n^m$ and$\mathbf{y}_n^o$ predicted by the kth Gaussian. The expectation of z_{kn} is $\langle z_{kn} | \boldsymbol{t}_n^o, \boldsymbol{\theta}_k \rangle = R_{kn}$, which is only measured on $\mathbf{y}_n^o$.

• E-Step:

Computing these unknown expectations is done in the E-Step where we start with z_{kn} which is defined as:

$$z_{kn} = \frac{|\Sigma_k^{oo}|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{y}_n^o - \mu_k^o)^T \Sigma_k^{-1,oo}(\mathbf{y}_n^o - \mu_k^o\}}{\sum_j^k |\Sigma_k^{oo}|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{y}_n^o - \mu_j^o)^T \Sigma_j^{-1,oo}(\mathbf{y}_n^o - \mu_j^o\}}\ ,$$

For the second and third unknown terms, we obtain

$$\langle z_{kn}\mathbf{y_n^m} | \mathbf{y}_n^o, \boldsymbol{\theta}_k \rangle = \langle z_{kn} | \mathbf{y}_n^o, \boldsymbol{\theta}_k \rangle \langle \mathbf{y_n^m} | z_{kn} = 1, \mathbf{y}_n^o, \boldsymbol{\theta}_k \rangle = R_{kn}\hat{\mathbf{y}}_{kn}^m\ ,$$

and

$$\langle z_{kn}\mathbf{y}_n^m\mathbf{y}_n^{m^T} | \mathbf{y}_n^o, \boldsymbol{\theta}_k \rangle = \langle z_{kn} | \mathbf{y}_n^o, \boldsymbol{\theta}_k \rangle \langle \mathbf{y}_n^m\mathbf{y}_n^{m^T} | z_{kn} = 1, \mathbf{y}_n^o, \boldsymbol{\theta}_k \rangle \tag{6.4}$$
$$= R_{kn}(\Sigma_k^{mm} - \Sigma_k^{mo}\Sigma_k^{-1,oo}\Sigma_k^{om^T}\hat{\mathbf{y}}_{kn}^m\hat{\mathbf{y}}_{kn}^{m^T})\ . \tag{6.5}$$

• M-Step:

Now the estimates $\hat{\mathbf{y}}_{kn}^m$ are used to substitute the missing values of $\mathbf{y}_n$ and to re-estimate the mean vector as in equation (4.5)

$$\mu_k = \frac{\sum_{n=1}^N p(k|\mathbf{y}_n)\mathbf{y}_n}{\sum_{n=1}^N p(k|\mathbf{y}_n)}\ .$$

The covariance has to be estimated in the three steps. For the part of the observed and observed/unobserved data we can directly use the equation for the full covariance matrix from equation (4.6):

$$(\Sigma_k^{o,o}) = \frac{1}{D}\frac{\sum_{n=1}^N p(k|\mathbf{y}_n^o)(\mathbf{y}_n^o - \mu_k^o)(\mathbf{y}_n^o - \mu_k^o)^T}{\sum_{n=1}^N p(k|\mathbf{y}_n^o)}\ ,$$

$$(\Sigma_k^{o,m}) = \frac{1}{D}\frac{\sum_{n=1}^{N} p(k|\mathbf{y}_n^o)(\mathbf{y}_n^o - \mu_k^o)(\hat{\mathbf{y}}_n^m - \mu_k^m)^T}{\sum_{n=1}^{N} p(k|\mathbf{y}_n^o)} .$$

In the case of the unobserved/unobserved part we have to substitute the whole outer product matrix of equation (4.6) into equation (6.4):

$$(\Sigma_k^{m,m}) = \frac{1}{D}\frac{\sum_{n=1}^{N} p(k|\mathbf{y}_n^o)((\Sigma_k^{mm})^{old} - (\Sigma_k^{mo})^{old}(\Sigma_k^{-1,oo})^{old}(\Sigma_k^{om^T})^{old} + \hat{\mathbf{y}}_{kn}^m \hat{\mathbf{y}}_{kn}^{m^T})}{\sum_{n=1}^{N} p(k|\mathbf{y}_n^o)} .$$

Diagonal Covariance Matrix In the case of a diagonal matrix the estimate of the missing data simplifies to the mean of the Gaussian center

$$\hat{\mathbf{y}}_{kn}^m = (\boldsymbol{\mu}_k^m) ,$$

and the covariance matrix for the observed part is

$$(\sigma_{d,k}^o)^2 = \frac{1}{D}\frac{\sum_{n=1}^{N} p(k|\mathbf{y}_n^o)(y_{d,n}^o - \mu_{d,k}^o)^2}{\sum_{n=1}^{N} p(k|\mathbf{y}_n^o)} ,$$

and for the unobserved part is

$$(\sigma_{d,k}^m)^2 = \frac{1}{D}\frac{\sum_{n=1}^{N} p(k|\mathbf{y}_n^o)((\sigma_{d,k}^m)^{old} + \hat{y}_{d,k,n}^m)^2}{\sum_{n=1}^{N} p(k|\mathbf{y}_n^o)} .$$

Spherical Covariance Matrix In the case of a spherical matrix the estimate of the covariance has a known and an unknown part where $||\mathbf{y}_n^o - \mu_k^o||$ represents the known part and

$$\langle z_{kn}||\mathbf{y}_n^m - \mu_k^m||^2\rangle = n_m((\sigma_k)^2)^{old} + (\hat{\mathbf{y}}_{kn}^m)^T(\hat{\mathbf{y}}_{kn}^m) - 2(\hat{\mathbf{y}}_{kn}^m)^T\mu_k^m + (\mu_k^m)^T\mu_k^m ,$$

represents the unknown part of the variance, where n_m represents the number of missing values in the data point $\mathbf{y}_n$. Thus the estimate for the variance is

$$(\sigma_k)^2 = \frac{1}{D}\frac{\sum_{n=1}^{N} p(k|\mathbf{y}_n)(||\mathbf{y}_n^o - \mu_k^o||^2 + \langle z_{kn}||\mathbf{y}_n^m - \mu_k^m||^2\rangle)}{\sum_{n=1}^{N} p(k|\mathbf{y}_n)} .$$

6.1.8 Extension of GTM for Missing Data Imputation using EM (GTMI)

The algorithm using GTM to impute missing data will be called GTMI. To initialise this algorithm PCA will be used. However PCA can not deal with missing data and thus we are using mean imputation to obtain the first two principal components with PCA but not bias the result through the usage of a better alternative imputation method.

Further the EM algorithm with missing data can be extended to the GTM model (Sun, 2002) based on a simplification of (Ghahramani and Jordan, 1994). The error function given by the log-likelihood from (6.2) can be written as,

$$-L_{comp} = -\sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn} \ln p(\mathbf{y}_n|\boldsymbol{\theta}_k) \; .$$

For the GTM model with a spherical covariance matrix this term can be expanded to

$$\begin{aligned} -L_{comp} = & -\textstyle\sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn}[\frac{1}{2}\ln|\boldsymbol{\Sigma}_k| + \frac{D}{2}\ln 2\pi \\ + & \quad \tfrac{1}{2}(\mathbf{y}_n^o - \boldsymbol{\mu}_k^o)^T\boldsymbol{\Sigma}_k^{-1,oo}(\mathbf{y}_n^o - \boldsymbol{\mu}_k^o) \\ + & \quad \tfrac{1}{2}(\mathbf{y}_n^m - \boldsymbol{\mu}_k^m)^T\boldsymbol{\Sigma}_k^{-1,mm}(\mathbf{y}_n^m - \boldsymbol{\mu}_k^m)] \; . \end{aligned}$$

After taking the expectation with respect to both sets of missing variables, one ends up with 2 unknown terms $z_{kn}\boldsymbol{t}_n^m$ and $z_{kn}\boldsymbol{t}_n^m\boldsymbol{t}_n^{mT}$, so one must calculate the expectation for these terms. To compute these expectations, variables $\hat{\mathbf{y}}_{kn}^m$ are introduced,

$$\hat{\mathbf{y}}_{kn}^m \equiv \langle \mathbf{y}_n^m | z_{kn} = 1, \mathbf{y}_n^o, \boldsymbol{\theta}_k \rangle = (\mu_k^m)^{old} \; ,$$

which are the linear least-squares regression between $\mathbf{y}_n^m$ and $\mathbf{y}_n^o$ predicted by the kth Gaussian, where the superscript "old" denotes the result from the last M-step: $(\mu_k^m)^{old} = (\boldsymbol{W}_{old}\boldsymbol{\Phi}(\boldsymbol{x}_k))^m$.

- E-step: The expectation of z_{kn} is $\langle z_{kn}|\mathbf{y}_n^o, \boldsymbol{\theta}_k\rangle = R_{kn}$, with

$$R_{kn} = \frac{\frac{\beta}{2\pi}^{D/2}\exp\{-\frac{\beta}{2}||\mathbf{y}(x_k;W) - \mathbf{y}_n||^2\}}{\sum_j^k \frac{\beta}{2\pi}^{D/2}\exp\{-\frac{\beta}{2}||\mathbf{y}(x_j;W) - \mathbf{y}_n||^2\}} \; , \tag{6.6}$$

measured only on the observed dimensions y_n^o of y_n.

- M-step: The weights are updated to $\boldsymbol{W}_{new}$ as in equation (4.15) for complete training data:

$$\boldsymbol{\Phi}\mathbf{G}_{old}\boldsymbol{\Phi}^T\mathbf{W}_{new}^T = \boldsymbol{\Phi}\mathbf{R}\mathbf{Y} \; , \tag{6.7}$$

where the missing data are given by the posterior means:

$$\langle \mathbf{y}_n^m | \mathbf{y}_m^o, \boldsymbol{\theta}_{old}\rangle = \sum_{k=1}^{K} R_{kn}\hat{\mathbf{y}}_{kn}^m \; .$$

Then the inverse variance is updated as follows

$$\frac{1}{\beta} = \frac{1}{ND}\sum_{n=1}^{N}\sum_{i=1}^{K} R_{in}(||\mathbf{y}_n^o - \mathbf{y}_k^o||^2 + \langle z_{kn}||\mathbf{y}_n^m - \mathbf{y}_n^m||^2\rangle) \; ,$$

where

$$\langle z_{kn}||\mathbf{y}_n^m - \mathbf{y}_n^m||^2\rangle = n_m(\beta^{-1})^{old} + (\hat{\mathbf{y}}_{kn}^m)^T\hat{\mathbf{y}}_{kn}^m - 2(\hat{\mathbf{y}}_{kn}^m)^T\mathbf{y}_k^m + (\mathbf{y}_k^m)^T\mathbf{y}_k^m \; ,$$

and n_m is the number of missing values in data point $\boldsymbol{t}_n$. A more detailed derivation can be found in (Sun, 2002).

6.1.9 Extension of B-GTM for Missing Data Imputation using EM (B-GTMI)

The algorithm using B-GTM to impute missing data will be called B-GTMI. The changes one has to make to allow the B-GTM algorithm to deal with missing data are straight forward. As with the spherical GTM one has to modify the E and M-step.

- E-step: The expectation of R_{in} being $\langle R_{in}|\mathbf{y}_n^o, \theta_k\rangle$ is similar to (6.6),

$$\begin{aligned} R_{in}(\mathbf{W}_{old}, \Sigma_{old}) &= p(\mathbf{x}_i|\mathbf{y}_n^o, \mathbf{W}_{old}, \Sigma_{old}) \,, \\ &= \frac{p(\mathbf{y}_n|\mathbf{x}_i, \mathbf{W}_{old}, \Sigma_{old})}{\sum_{j=1}^{K} p(\mathbf{y}_n^o|\mathbf{x}_j, \mathbf{W}_{old}, \Sigma_{old})} \,, \end{aligned}$$

measured only on the observed dimensions $\mathbf{y}_n^o$ of $\mathbf{y}_n$.

- M-step: The weights are updated to $\boldsymbol{W}_{new}$ as in equation (6.7) for complete training data:

$$\mathbf{\Phi}\mathbf{G}_{old}\mathbf{\Phi}^T\mathbf{W}_{new}^T = \mathbf{\Phi}\mathbf{R}\mathbf{Y} \,,$$

where the missing data are given by the posterior means as for the GMM in (6.3)

$$\begin{aligned} \hat{\mathbf{y}}_n^m &= \textstyle\sum_{i=1}^{K} R_{in}\langle \mathbf{y}_n^m|\mathbf{y}_m^o, \theta_{old}\rangle \\ &= (\mu_k^m)^{old} + \Sigma^{mo}\Sigma^{-1,oo}(\mathbf{y}_n^o - (\mu_k^m)^{old}) \,. \end{aligned}$$

The covariance matrix has to be updated individually for every data point with

$$\Sigma^{mm} = \frac{1}{ND}\sum_{n=1}^{N}\sum_{i=1}^{K} R_{in}((\Sigma^{mm})^{old} - (\Sigma^{mo})^{old}(\Sigma^{oo,-1})^{old}(\Sigma^{om,T})^{old} + \hat{\mathbf{y}}_n^m\hat{\mathbf{y}}_n^{m,T}) \,,$$

$$\Sigma^{om} = \frac{1}{ND}\sum_{n=1}^{N}\sum_{i=1}^{K} R_{in}(\Xi(\mathbf{x}_k, \mathbf{W}) - \mathbf{y}_n^o)(\Xi(\mathbf{x}_k, \mathbf{W}) - \hat{\mathbf{y}}_n^m)^T \,,$$

$$\Sigma^{mo} = \frac{1}{ND}\sum_{n=1}^{N}\sum_{i=1}^{K} R_{in}(\Xi(\mathbf{x}_k, \mathbf{W}) - \hat{\mathbf{y}}_n^m)(\Xi(\mathbf{x}_k, \mathbf{W}) - \mathbf{y}_n^o)^T \,,$$

$$\Sigma^{oo} = \frac{1}{ND}\sum_{n=1}^{N}\sum_{i=1}^{K} R_{in}(\Xi(\mathbf{x}_k, \mathbf{W}) - \mathbf{y}_n^o)(\Xi(\mathbf{x}_k, \mathbf{W}) - \mathbf{y}_n^o)^T \,.$$

6.1.10 Performance Indicator

To compare the different imputation methods a measure of performance is needed. The following commonly used error measure (Olier and Vellido, 2005; Cooke *et al.*, 2001; Brás and Menezes, 2006) accounts for the difference between the original y_i value and the imputed $\hat{y}_i$ value and gives an idea how well the imputation has performed.

Root Mean Square Error:

$$RMSE = \left(\frac{1}{N} \sum_{i=1}^{N} [y_i - \hat{y}_i]^2 \right)^{\frac{1}{2}}$$

The RMSE is an estimate of the standard deviation of the residual errors from the predictions. It can be sensitive to outliers since it is a second moment statistic. However this measure is problematic because it will show a counter intuitive behaviour when plotted over an increasing number of missing data points N per dimension. The division by N causes the measure to show a decreasing variance or spread with an increase in the proportion of missing data. This causes the pretence of a decrease in variability of the error with an increase in missing values. However we could not find a more suitable measure which allows for the comparison of the error across different proportions of missing values.

In this thesis the RMSE is calculated for every data point and then averaged over all the data points. Further this average is calculated over a number of random missing data patterns and then averaged again. The results are therefore presented as an average over the average of the RMSE **(ARMSE)** error. This is done because different random missing data patterns will benefit or impair the various imputation methods and a sufficient sample size is needed to draw valid conclusions.

6.1.11 General behaviour of the performance indicator and the imputation methods

Since the ARMSE error is used as the measure for the benchmark it is advisable to take a closer look at its behaviour. For example it is important to know if unexpectedly high variances are occurring which render the average a less meaningful indicator in the final results. Since the work explores different proportions of missing data it is important to ensure comparability across this range.

To examine the behaviour of the RMSE and the imputation methods in combination with the random generation of missing variables box plots are employed.

The methods were tested on the oil flow data; however the results across other data sets have been tested and found to be similar. To research this behaviour 50 random missing data patterns for each level of missing values, p_i, were generated.

The results for MI, MRI and GTM can be seen in Figure 6.1. These methods are chosen since they summarise the behaviour of the other methods as well.

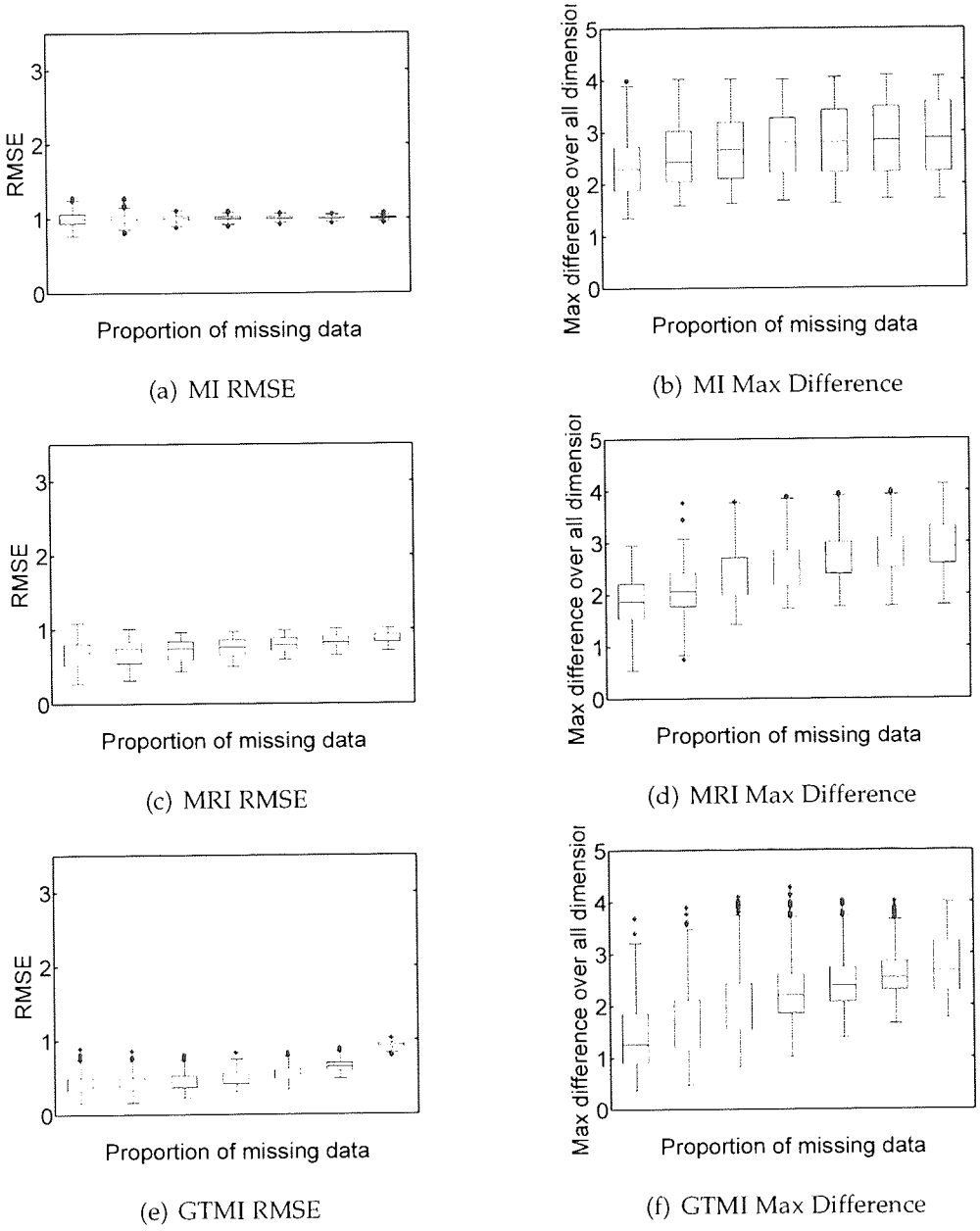
They show no unsurprising or unusual behaviour. The variability in the data is due to the different random missing data patterns. In the case of MI the spread around the error is quite small and thus the average is a good approximation of the overall performance. As already explained the RMSE is showing a decrease in variability with an increase in the proportion of missing values because of the smoothing effect when dividing by large N.

In the case of MRI and GTM the results are roughly similar. There is a bigger spread around the median RMSE for MRI. If one looks closer however one can see that the spread between the mean and the worst result stays the same, while the quantile pointing towards the better outcomes moves closer to the mean. This makes sense because with higher proportions of missing data the ability to undertake meaningful inference decreases and so does the performance of the imputation methods. If a point is not close to the density of the other points the inference of missing values will be bad no matter how many values are missing. Therefore one has to be aware that the imputation methods only perform well on average since in some cases the imputed value might still be very far away from the true value. However an increase in missing values makes it harder to infer the right values even for points close to the centre of the data density.

A special discussion is needed for the result in Figure 6.1(e), when one looks at the big jump or decrease in performance when going from $p_i = 0.5$ to $p_i = 0.6$. This decrease in performance will also show up in the next section. This anomaly is due to the way the GTM algorithm is initialised. For this experiment we initialised the GTM algorithm with PCA. The standard algorithm for PCA can not deal with missing data and therefore we used MI as an pre-processing step before using PCA to find the first two principal components on which GTM is initialised. When the proportion of missing values increased from $p_i = 0.5$ to $p_i = 0.6$ MI+PCA lost the ability to pick out the right principal components and thus GTM was initialised in a suboptimal state.

To compensate for the smoothing of the RMSE we also looked at the distribution of the maximum difference. This distribution is in essence looking at the worst outliers in terms of accuracy, where the models predict very wrong values. The results can also be seen in Figure 6.1. Here one can see that the increase in missing values boosts the variance and the number of extremely bad predictions for single values. As already mentioned this behaviour is smoothed out when looking at the RMSE.

Finally if one looks at the general distribution of the imputed values in Figure 6.2 one can see that the imputation methods in general behave well. The MI is centred around 0, which is expected since all data sets in this work have been normalised and have zero mean. Further one sees no big differences between MRI and GTM, which is reassuring and indicates that both methods do not tend to introduce a systematic bias, exhibit spontaneous breakdowns or other unexpected behaviour. The exception is again GTM when $p_i = 0.6$ however this is due to the already discussed problem when initialising GTM.



(a) MI RMSE

(b) MI Max Difference

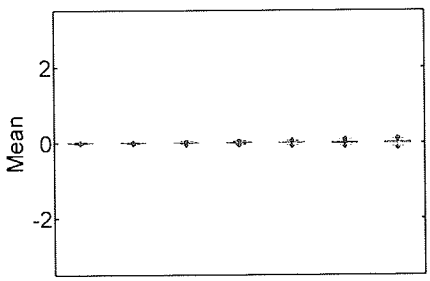
(c) MRI RMSE

(d) MRI Max Difference

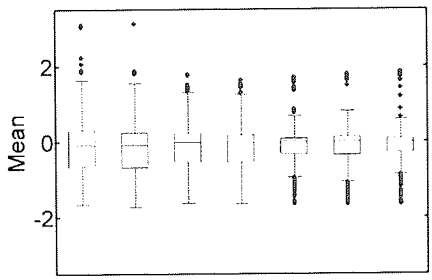
(e) GTMI RMSE

(f) GTMI Max Difference

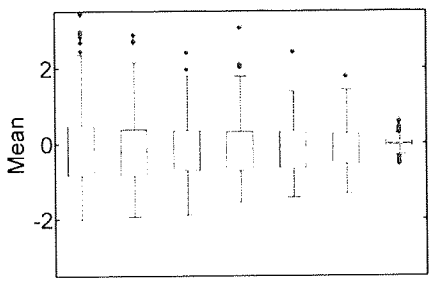
Figure 6.1: Behaviour of the RMSE and the distribution of the maximal differences between the original and imputed values. The boxplots show the distribution of results for different proportions of missing data on the oil flow data. Each boxplots describes an experiment with 50 different missing data patterns given the proportion of missing data and the used imputation method. The RMSE behaves as expected with an decrease in average performance as well as variance when the proportion of missing data increases. The distribution of the maximum difference (i.e. worst result) also behaves as expected and the errors get worse the more data are missing with a constant variance.



(a) Mean Imputation



(b) MRI



(c) GTM

Figure 6.2: Distribution of the imputed values: little bias is shown.

6.1.12 Benchmark

To compare the different imputation methods the multi-dimensional oil flow data and the toy 20D and 60D toy data sets, described in chapter 3, were used. The performance of the imputation methods was measured on a range of proportions $p_i = [0.1, .., 0.6]$. 50 random missing data patterns for each p_i were generated to average the results of the performance indicators and get a representative value.

The results of the benchmark experiment can be seen in Figure 6.3. In general the ARMSE shows a similar trend when averaged over all the dimensions. The most widely employed missing data imputation method, MI, always performs the worst. WMI performs better than MI on all data sets. The more advanced imputation methods like MRI, GTMI and B-GTMI always perform better than WMI or MI. MRI and GTMI show different performance on the different data sets. GTMI outperforms MRI on the multi-flow oil data but is worse on the toy data sets with low proportion of missing values (< 0.3). This might be explained by the linear nature of the toy data sets. The 20Dim and 60Dim toy data sets were generated by using a 2×2 RBF which does not allow for a highly non-linear structure.

The B-GTMI and BPCA algorithm outperform all other tested imputation methods on all the data sets. In the case of the multi-flow oil data B-GTMI is better with lower amounts of missing values (< 0.5). However in the case of the two test data set 20Dim and 60Dim BPCA clearly outperforms all other methods. The advantage of BPCA and B-GTMI is because both use the full covariance structure which clearly aids in the imputation. The superior performance of BPCA on the 20Dim and 60Dim toy data sets can again be explained by the near linear nature of these data sets.

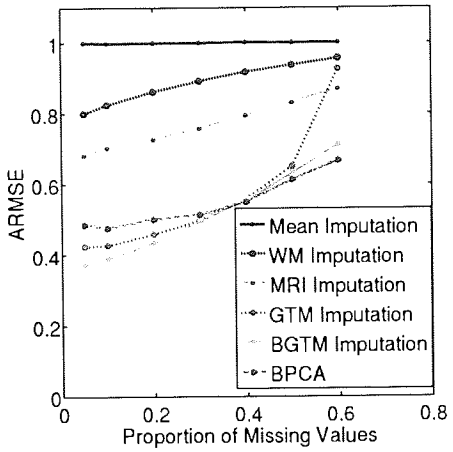
Another very interesting observation is the continuous decrease of the performance of B-GTMI without any jumps or transitions like in the case of GTMI. Especially in the case of oil flow data and $p_i = 0.6$ we know that the GTMI performs far worse because of the poor initialisation. However B-GTMI seems to be more robust and manages to compensate for it in this case.

6.1.13 Projection Results

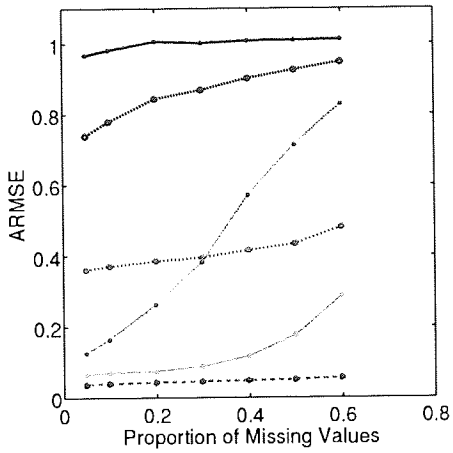
Measuring the RMSE for imputation algorithms gives an indication on how well the model is able to infer the missing data. However we are also interested in evidence about their use in data exploration. For the applied scientist it is important to know when an imputation algorithm might still help to reveal the hidden structure of the data and when it is unlikely to make much of a difference.

To get an indication about the usability of these algorithms we visually compared the projections of PCA, GTMI and B-GTMI. The PCA projection on missing data patterns was obtained by preprocessing the data either with MI or MRI. The proportion of missing values were $p_i = 0.2$ and $p_i = 0.6$.

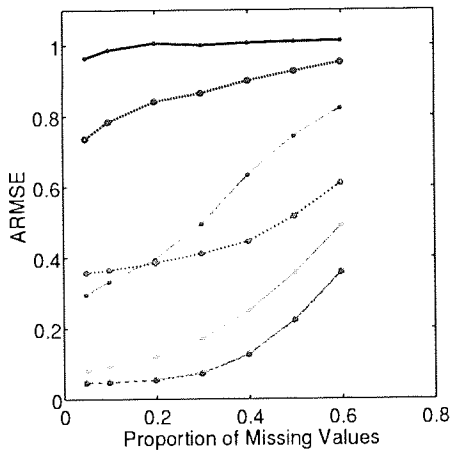
The projection results can be seen in Figures 6.4 and 6.5 for the oil flow data. Further results can be found in appendix A. These results indicate that the pro-



(a) Multi Flow Oil Data



(b) 20 Dim toy data



(c) 60 Dim toy data

Figure 6.3: Performance of the imputation methods with different proportions of missing data $p_i = [0.05, \ldots, 0.6]$ on different data sets. (a) In the case of the oil flow data B-GTM is as good or better than all other methods for little to medium amounts of missing data $p_i = [0.05, \ldots, 0.4]$. (b-c) The GTM generated toy data show a clear advantage for BPCA regardless of the proportion of missing data with B-GTM being the second best method.

jections obtained through the use of GTM and B-GTM are still meaningful when one has a low proportion of missing data. However, as one would expect, the results are less informative once the amount of missing data grows beyond a certain threshold. The usage of PCA combined with MI or MRI provides only projections of very limited significance, where it is still possible to distinguish class 3 from class 1 and 2 in the case of a small amount of missing data. However the methods produce very uninformative projections with large amounts of missing data where no distinction between classes can be made. B-GTMI and GTMI produce meaningful results with low amounts of missing data and one can still distinguish between classes in Figure 6.4. However both algorithms fail when confronted with high proportions of missing data as can be seen in Figure 6.5, where no distinction between different classes is possible. The worst result is given by GTM which due to the bad initialisation breaks down and projects all data points onto a one-dimensional structure.

From the results it is apparent that the deterioration in performance is directly related to the increasing proportions of missing data: the threshold, which renders a visualisation uninformative will depend on the data set, the missing data pattern and the require accuracy from the practitioner.

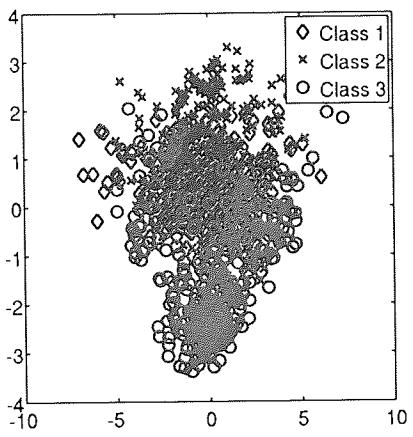
6.2 Missing data as a way to assess the model fit in unsupervised learning

In data exploration and visualisation most tasks are performed on unlabelled data. This poses a big problem when one wants to evaluate the performance of the model or to assess whether results are meaningful. Since the data are unlabelled there is no possibility to assess directly whether the visual results or structures are meaningful.

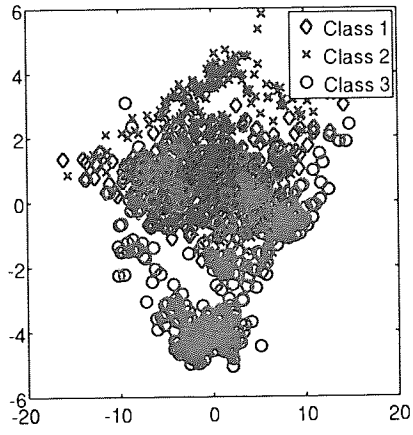
The only way to assess the actual quality of the exploration or visualisation is through the use of indirect measures. However common measures like the likelihood might be misleading since they are relative and one has no a priori information about what value of the likelihood is good. The usage of the likelihood permits the comparison of models against each other but cannot tell whether a particular result is at all meaningful.

The use of multiple measures to help with the validation of model results makes for a more robust assessment. An understanding of the restrictions of the measures used is essential. Otherwise faulty conclusions might be reached if the employed visualisation and exploration algorithms fail to capture the structure of the data. For example, if one only relies on the likelihood and all the compared models do not capture the structure of the data, even the model with the best likelihood might not provide a good or meaningful visualisation.

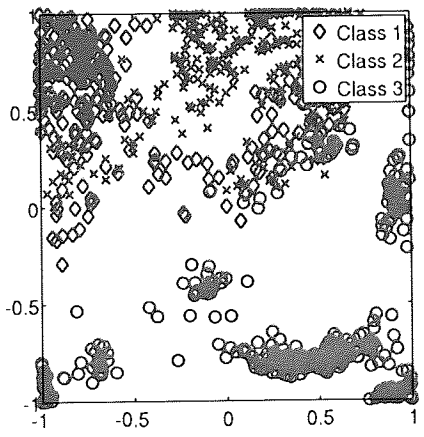
One approach to help assess each method is based on re-sampling (Moeller and Radke, 2006; Yu, 2003) and comparison of the moments of the data. The general idea is that if one samples new data from the model these newly sampled



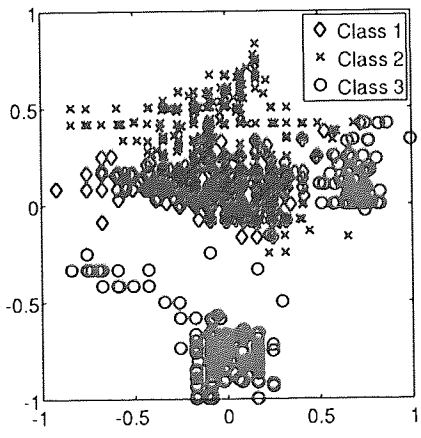
(a) MI then PCA



(b) BPCA

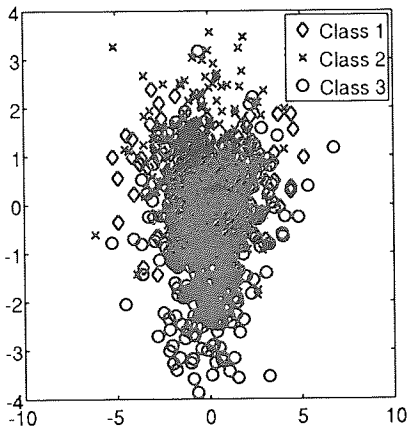


(c) S-GTM

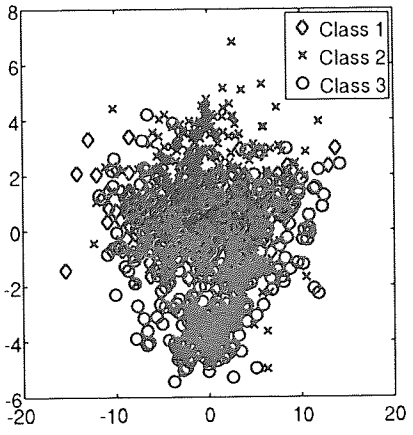


(d) B-GTM

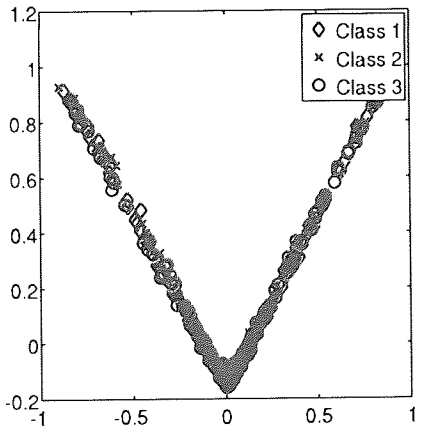
Figure 6.4: Example projection on the multi-flow oil data, with $p = 0.2$. In the PCA projections (a) and (b) it is not possible to distinguish between the classes. In the GTM projections (c) and (d) the performance has deteriorated and the class boundaries are not clear.



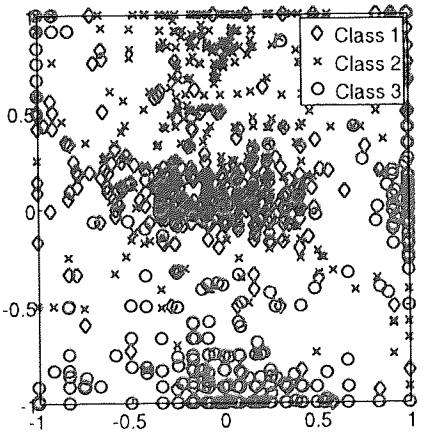
(a) MI then PCA



(b) BPCA



(c) S-GTM



(d) B-GTM

Figure 6.5: Example projection on the multi-flow oil data, with $p = 0.6$. It is impossible to distinguish between the classes in all classes (a)-(d) and in the case of of spherical GTM (c) the algorithm broke down completely.

data should have similar statistics to the original data. However this is a very general measure and no guarantee of a meaningful visualisation.

A similar approach would be to create missing data artificially and retain the original value for comparison. Then the estimation of missing data can be seen as a re-sampling approach. However in contrast to the overall statistics which are a global measure the estimation of missing data is a local measure. It tells the user how well the model is doing locally at a certain point in the data space. The projections in Figure 6.4 and 6.5 support this idea since they show a relation between the RMSE and the separability of the classes in the projection. For example the collapse of the projection of GTM in Figure 6.5(c) happens simultaneously with a large increase of the RMSE for GTMI at $p = 0.6$ in Figure 6.3. However a better RMSE for one method is not imperative for a better visualisation as can be seen for BPCA, GTM and B-GTM when looking at the results of the projections in the appendix in Figures A.3, A.4, A.5, A.6. Even though BPCA has the lowest RMSE the visualisation of GTM and B-GTM are as good or better if one assesses them on the basis of class separability in the projection. Hence this measure has to be seen as providing a partial assessment of performance and should be used in combination with other measures.

To use this measure, visualisation methods need to be able to deal with missing data. Probabilistic based methods can in most cases be modified to cope with missing data. The estimates of the model can then be used as imputation estimates for the missing data.

This fact is exploited in chapters 5 and 7 where the ARMSE is one of the key measures to assess whether the extension of GTM to B-GTM enhances the visualisation and model fit.

6.3 Conclusion

The GTM algorithm has in previous work proven to successfully cope with missing values in data sets (Vicente *et al.*, 2004; Olier and Vellido, 2005; Sun, 2002). However these studies were concerned with small test data sets and the general ability of the algorithm to handle missing data. To the knowledge of the author no extensive comparative and systematic study has been performed until Schroeder *et al.* (2008). The probabilistic formulation of the GTM algorithm allows for the inclusion of partially missing data which maximises the amount of available information when fitting the model to a given data set. In this chapter it has been shown that GTM remains relatively robust on toy data even when a high proportion of data is missing. The extension from GTM to B-GTM enhances the capabilities of the algorithm when dealing with missing data if the information provided about the block structure is correct.

To benchmark the imputation abilities of GTMI and B-GTMI two well known imputation algorithms (WMI and MRI) were used. These algorithms have proven to be stable and give reliable results on the toy data sets, which in the case of MRI are as good or better than GTMI. WMI and MRI have far lower computational

costs than GTMI, B-GTMI or BPCA. They might be good alternatives in certain cases. In densely populated data sets WMI performs well. In data sets with high linear relations between the variables MRI performs well. However the results suggests that GTMI, B-GTMI and BPCA have a very useful role in the replacement of missing values since they outperform WMI and MRI especially in the case of high proportions of missing data.

The visualisation capabilities of GTMI and B-GTMI are subjectively superior when compared against a two stage process of PCA and MI and depending on the case even over BPCA. The results indicate that GTMI, B-GTMI and BPCA will still deliver meaningful results even when faced with moderate quantities of missing data. However there is no guarantee that the visualisation will be meaningful and the algorithms will ultimately fail and produce inappropriate projections if the proportion of missing data is too great.

The use of artificial missing data itself might have application to the assessment of the fit of the model: in data exploration and visualisation, where most tasks are performed on unlabelled data, this measure could prove to be very helpful. Common measures like the likelihood might be misleading since they are relative and one has no a priori information about what level of likelihood will qualify as good. In the case where one is able to modify the model to deal with missing data and get a viable estimate for the missing data the ARMSE might be a good additional measure to quantify the performance of the model where a performance close to zero will indicate a perfect fit of the model. A small ARMSE alone will not ensure a good visualisation and thus ARMSE should be used in conjunction with other measures to assess the quality of the visualisation on unlabelled data.

Another application for the RMSE error and artificial data might be the assessment of the non-linearity of a data set. If GTM or B-GTM is performing better than BPCA like in the case of the oil flow data, it could be argued that this data set must be of a non-linear nature which can not be picked up by BPCA. Possible approaches to this idea will be discussed in the future work chapter at the end of this thesis.

7 Working with real data

CONTENTS

7.1 Data pre-treatment 130
7.2 Exploring the non-linear mapping of GTM 131
7.3 Integration of GTM and PCA with pIGI for data exploration . . 138
7.4 Case Study: Barents Sea Data 138
7.5 Benchmark Study 144
7.5.1 African Data 147
7.5.2 North Sea Data 151
7.6 Performance Study: Speed of the methods 154
7.7 Summary 156

The aim of this Ph.D. is to explore the possibility of using non-linear and probabilistic models in geochemistry. New methodologies were developed to deal with particular properties of geochemical data, such as blocks of correlated variables. The theoretical foundation and application of these methods were discussed and benchmarked on toy data sets in earlier chapters. In this chapter their practical use will be highlighted in a small case study on real geochemical data.

First some general aspects of data pre-treatment and the difficulties this presents are discussed. Then different diagnostics for the visualisation of GTM will be illustrated on a simple three-dimensional toy data set. This is followed up by a short introduction to the data visualisation tools we used and developed.

Finally different studies to demonstrate the performance of the algorithms on real data are discussed. First case study on the Barents Sea data demonstrates the use of non-linear techniques. Second, a benchmark study on two additional geochemical data sets was carried out to benchmark the visualisation and imputation algorithms. Finally a performance study is presented to show how long it takes to run the different algorithms.

7.1 Data pre-treatment

Mixing major, minor and trace elements and overcoming differences in the amount of variation. In multi-element analysis of geological materials one usually deals with elements occurring in very different concentrations. In rock geochemistry the chemical elements are divided into "major","minor" and "trace" elements. Major element concentrations are 3 % upward, minor element concentrations are about 1% and trace elements are measured in ppm (parts per milion) or even ppb (parts per bilion). These can then be combined with data from more complex analysing and screening techniques like GC-MS which might not give concentrations at all but relative peak heights or areas. This becomes a problem if one uses statistical methods and considers multiple variables simultaneously since naturally the variables with the greatest magnitude will dominate the results. As a consequence one should not mix variables of different units in one and the same multivariate analysis without prior treatment (Rock, 1988). Possible pre-treatment includes transformation and or standardisation techniques. A good discussion of the topic can be found in Reimann *et al.* (2002) which discusses data pre-treatment and the application of factor analysis on geochemical data. Another good reference is Kvalheim *et al.* (1994) which deals with the implicit assumption in many models (e.g. spherical GTM and PCA) that all variables are independent and have similar levels of noise.

In the following sections we pre-processed all real data sets in the same way. First all GC-MS data are block normalised to a value of 100 for each sample. Then the values are autoscaled (standardised) over the variables; subtraction of the mean and division by the standard deviation. This was done to eliminate the influences or dominance of individual compound concentrations (Brereton, 2003).

Different Labs and Sources. One major problem when analysing geochemical data is the constrained set of samples. Many geochemical data sets feature less than 100 samples while having 100 or more variables. If one wants to augment these data sets by similar data from the same region (or with similar history and constitution) one might run into the problem that not all samples are analysed by the same laboratory. It is known that different laboratories will report different results even if they all receive an identical sample (Blankenhorn *et al.*, 1992; Isaacs, 2001; Kucklick *et al.*, 2002). This laboratory difference is due to the state and configuration of the GC-MS, the amount of sample matter injected into the GC-MS and in unfortunate cases to the mal-handling of samples and bad practice in some labs (i.e. samples were left open for too long and the lighter molecules evaporated).

This difference introduces a bias into the analysis of the samples and is not easily treated. In theory the use of ratios or autoscaling should reduce the bias which is caused by different amounts of sample matter injected. Further one would hope that for good labs the configurations of the GC-MS are standardised and highly comparable. To the knowledge of the author no research has been done on how to reduce or treat this laboratory bias. It is therefore unknown what to do, except for autoscaling, if one needs to mix the results of different laboratories for multivariate analysis. This problem has not been addressed in this work and is in dire need of future research. The data set we used all came from multiple labs where a difference between the laboratories could clearly be observed. Thus a possible bias had to be accepted when benchmarking the different methods.

7.2 Exploring the non-linear mapping of GTM

The non-linear mapping and probabilistic formulation of GTM has many advantages but since the mapping is non-linear a straight forward interpretation of the mapping like that given by the loadings in PCA is not possible. However using techniques for data exploration, namely parallel coordinate plotting (Inselberg *et al.*, 1990; Edsall, 2003), one can acquire a similar level of information about the projection. The technique of parallel coordinates is very powerful and has been applied in outlier detection and visualisation in earlier applications in geochemistry (Grunfeld, 2007). The technique, illustrated in Figure 7.1, itself is relatively simple: The D dimensional data space is plotted in X/Y plot where one partitions the X axis into D equidistant parallel parts and plots the samples as discrete values for each part and connects them with a line. Instead of displaying all samples in parallel coordinates, which is impractical for larger data sets, we only plot the samples which are selected in a certain part of the manifold. This technique is called local parallel coordinates (Maniyar and Nabney, 2006b) and it allows the user to study the properties of the points in the high-dimensional data space while working with the lower dimensional representation.

However, the non-linear mapping of the GTM brings certain dangers. If the GTM model is chosen to be too flexible it can happen that the manifold folds over

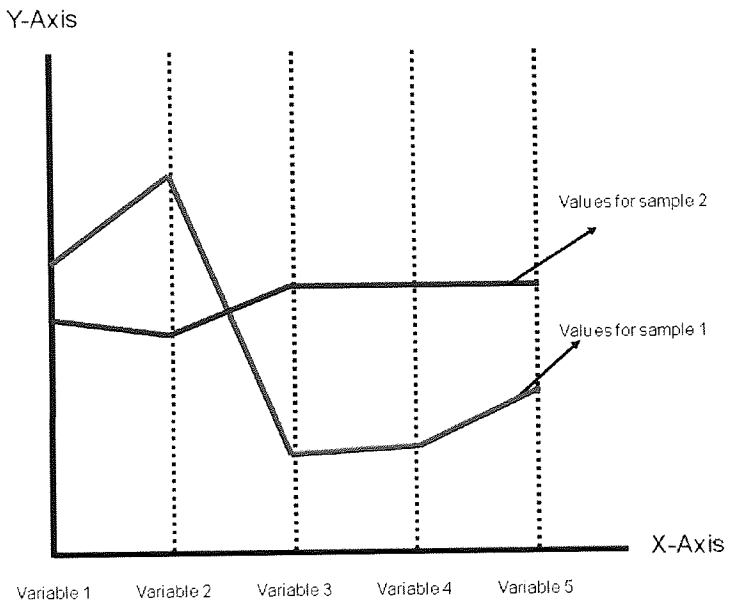


Figure 7.1: Schematic illustrating the set-up of a parallel coordinate plot with 5 variables and 2 samples. The values of the samples for each variable are plotted on the dotted line and joined by lines in their respective colours.

or twists. As result this might cause points to get projected to different corners of the manifold while in reality they are very close together in data space. This can be checked by looking at the *mode* of the posterior distribution of each point. The normal visualisation or projection is given by the mean, which is the average of all nodes weighted by the inverse distance to the data point. If the manifold is smooth the mode and the mean should be very close together. However if the mode deviates considerably from the mean this is an indication that there are problems with the model fit and one should consider reducing the flexibility of the GTM and repeating the modelling.

Another characteristic of the GTM is that the distances in the visualisation space can be misleading since the GTM behaves like a rubber sheet and will stretch itself in the data space. Thus points close in the visualisation space may not be close in the data space. This however can be checked by looking at the magnification factors (Svensén and Williams, 1997) which give an indication of how strongly the GTM is stretched in a certain area.

To show the process of data exploration we will give a demonstration on a modified version of the Swiss-roll data set in chapter 3. To demonstrate the diagnostics of GTM we include an additional outlier group in the data. The 2D and 3D structure of the data set and the projection obtained through PCA can be seen in Figure 7.2. In this example the first two principal components obtained by PCA fail to capture correctly the non-linear structure of the data. This is shown by the fact that the main classes overlap strongly in visualisation space but not in data space. If this simple approach to PCA-based visualisation was the only method employed in this case an incorrect judgement about the separability of the classes would be made. Albeit if one looks at all possible cross plots between the obtained principal components (in this case PC2 and PC3) one can distinguish between the classes and identify the underlying structure. However in practise with far higher dimensional data it can not be guaranteed that PCA will capture the structure even when looking at more than the first two principal components. The simple reason is that it quickly becomes unfeasible to look at all possible cross plots if one goes beyond 4-5 principal components.

To demonstrate the advantages of non-linear data exploration we fitted a GTM to the data with the following specifications: $[8 \times 8]$ RBF network, $[25 \times 25]$ grid of latent space nodes and initialisation by the Isomap algorithm. After testing multiple parameter combinations the likelihood of the model indicated that these parameters yielded the best results. The results can be seen in Figure 7.3 and 7.4. The visualisation of the GTM shows a perfect separation of the classes with the outlier group on the left border of the manifold. Using the diagnostics for the GTM one can now obtain additional information about the manifold. Looking at the magnification factors in 7.4(a), one can see that these outliers are actually in a highly stretched area. This indicates that they are much further away from the rest of the data than is immediately apparent in the projection and one should explore these groups in more detail. This can be done by using the parallel coordinate plots. To demonstrate how parallel coordinate plots can be used we selected

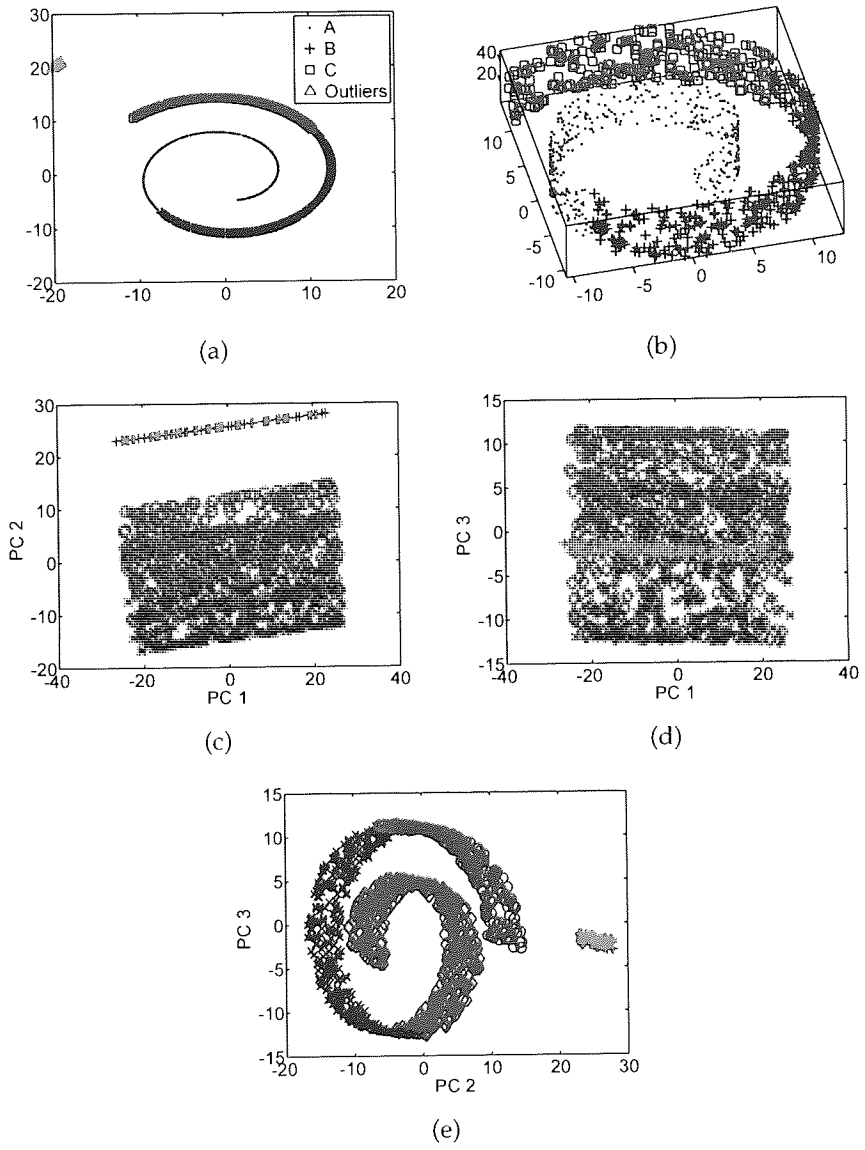
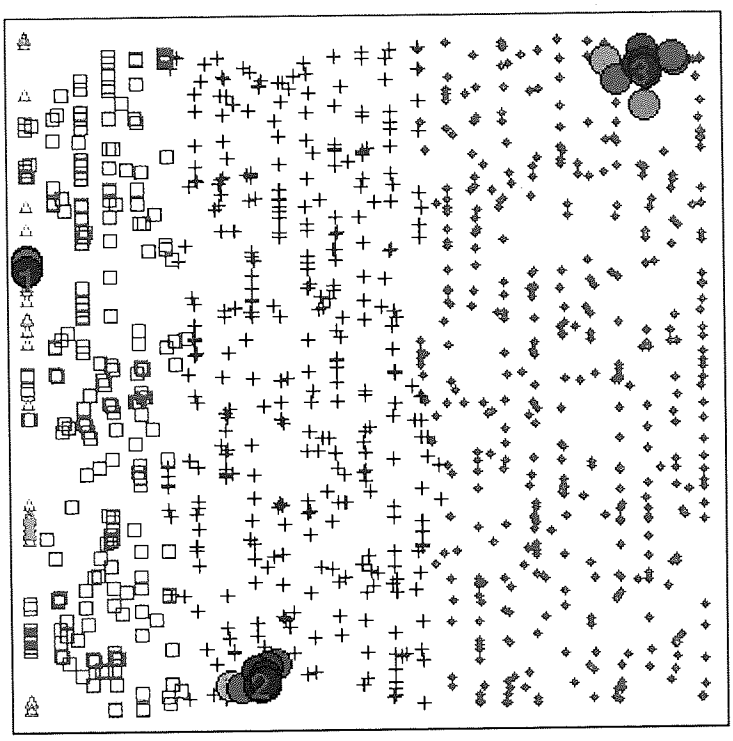


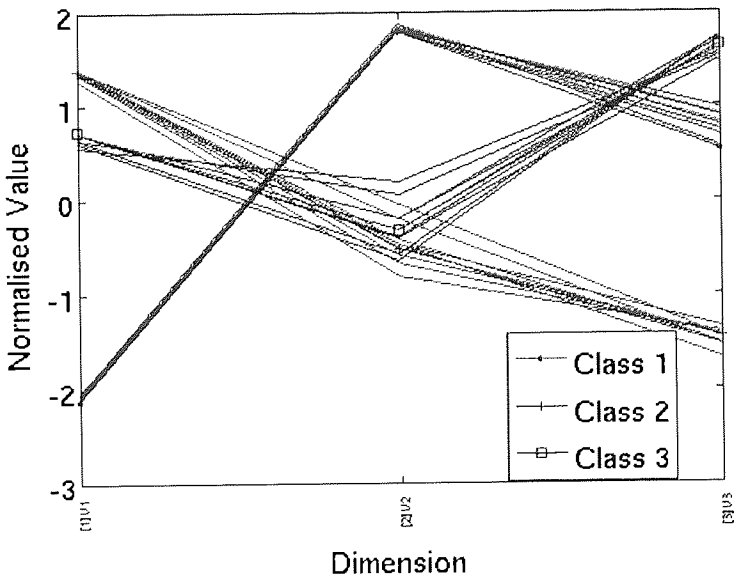
Figure 7.2: a) 2D Structure of the Swiss-roll with outlier group. b) 3D Structure of the Swiss-roll. c-e) Visualisation using PCA, where the structure is visible after looking at cross plots of all possible combinations of available principal components.

3 groups of 10 points in different regions of the manifold. The parallel coordinate plot in 7.3(b) shows clearly the different characteristics of data in the different areas of the manifold. The outlier group, marked as class 1, is very different to the other groups in all three variables, while the 2nd and 3rd group are mainly separated because of differences in the first and second variable. Finally the plot of the modes in 7.4(b) shows that there are no big differences between the means and the modes, which is a safeguard to check that nothing unusual has happened to the manifold.

In a similar fashion one can explore the non-linear visualisation of geochemical data which will be demonstrated on Barents Sea data in section 7.4.

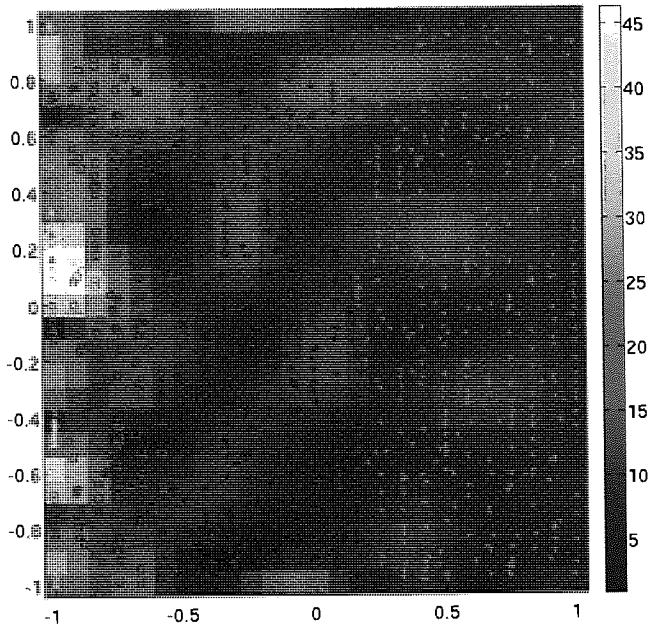


(a)

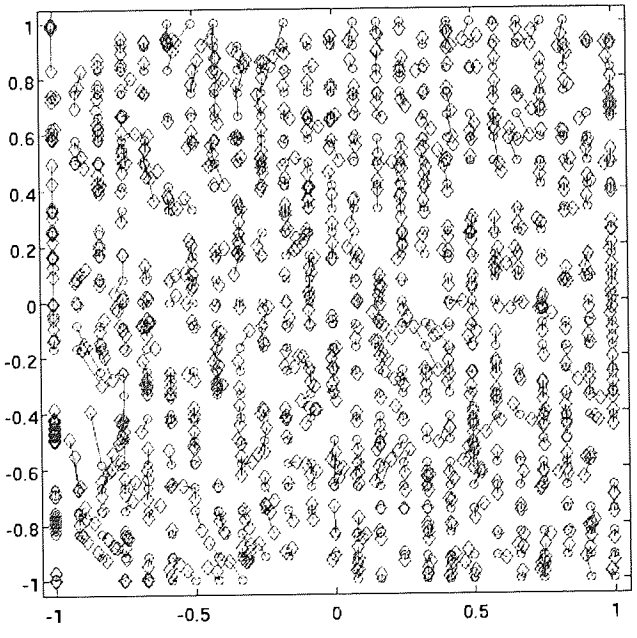


(b)

Figure 7.3: Visualisation of Swiss-roll using GTM: a) The projection where 3 sub classes have been marked (identifiable by bigger and red/brown/yellow coloured dots) . b) Parallel coordinates plot of the 3 sub classes.



(a)



(b)

Figure 7.4: Visualisation of Swiss-roll using GTM: a) Magnification factors map with the projection plotted into the map. b) Mode projection with distance of points to mean projection. Mean and and corresponding mode joined by a line.

7.3 Integration of GTM and PCA with pIGI for data exploration

To enable us to validate our methodologies on geochemical data it was necessary to integrate software with a geochemical analysis toolkit. For the later we used pIGI (IGI-Ltd., 2009), shown in Figure 7.5. pIGI is a powerful data analysis tool which is developed and maintained by IGI, who were co-funders and collaborators on this project. It is used by geochemists to use screening, molecular and isotopic geochemistry for well, acreage, prospect and basinal evaluation.

The integration into pIGI was done as an additional module under the .NET framework to ensure usability and easy maintenance in future pIGI developments. This was done by implementing the algorithms in managed C++ using public Lapack/Blas[1] routines and wrapping this code in a .NET wrapper.

The tool is very basic at the moment featuring simple data manipulation techniques (like autoscaling, block normalisation and mean imputation for missing values). Further futures include simple plotting functions for PCA (scores and loadings). In the current version, the only visualisation model implemented are PCA and spherical GTM. This tool is still a prototype. The layout can be seen in Figure 7.6. In the future it will be fully integrated into the next version of pIGI to interface with the advanced plotting and sample selection methods of pIGI. This should make the program a very powerful tool for data exploration and analysis.

For academic use and as a prototype another tool , named DVMS (Data Visualisation and Modeling System), was used and modified in Matlab. This tool is based on modified code from the Netlab toolbox (Nabney, 2002), GPLVM toolbox (Lawrence, 2005), BPCA toolbox (Oba *et al.*, 2003) and earlier work by Maniyar and Nabney (2006b). This tool does not feature any data pre-processing features. The implemented algorithms are PCA, S-GTM, B-GTM, non-linear initialisation for GTM with Isomap and BPCA.

7.4 Case Study: Barents Sea Data

In order to test the capability of GTM for visualising non-linear variation in geochemical data, a suite of 33 oils and condensates from the Barents Sea was used as input. This initial study was carried out to validate the method (i.e. show the application of the model and the use of the diagnostics) rather than seeking to fully interpret the data. A fundamental requirement of such a basic test of GTM was to limit the range of artefacts that would introduce artificial variation in the dataset e.g. missing data, wide discrepancies in analytical procedure,and oils with significant variation in bulk physico-chemical characteristics. Due to the selection of a number of condensates as samples, the primary data (i.e. integrated peak

[1]BLAS (Basic Linear Algebra Subprograms) and LAPACK (Linear Algebra PACKage) and are software libraries for numerical linear algebra. They provides routines for solving systems of linear equations and linear least squares, eigenvalue problems, and singular value decomposition.

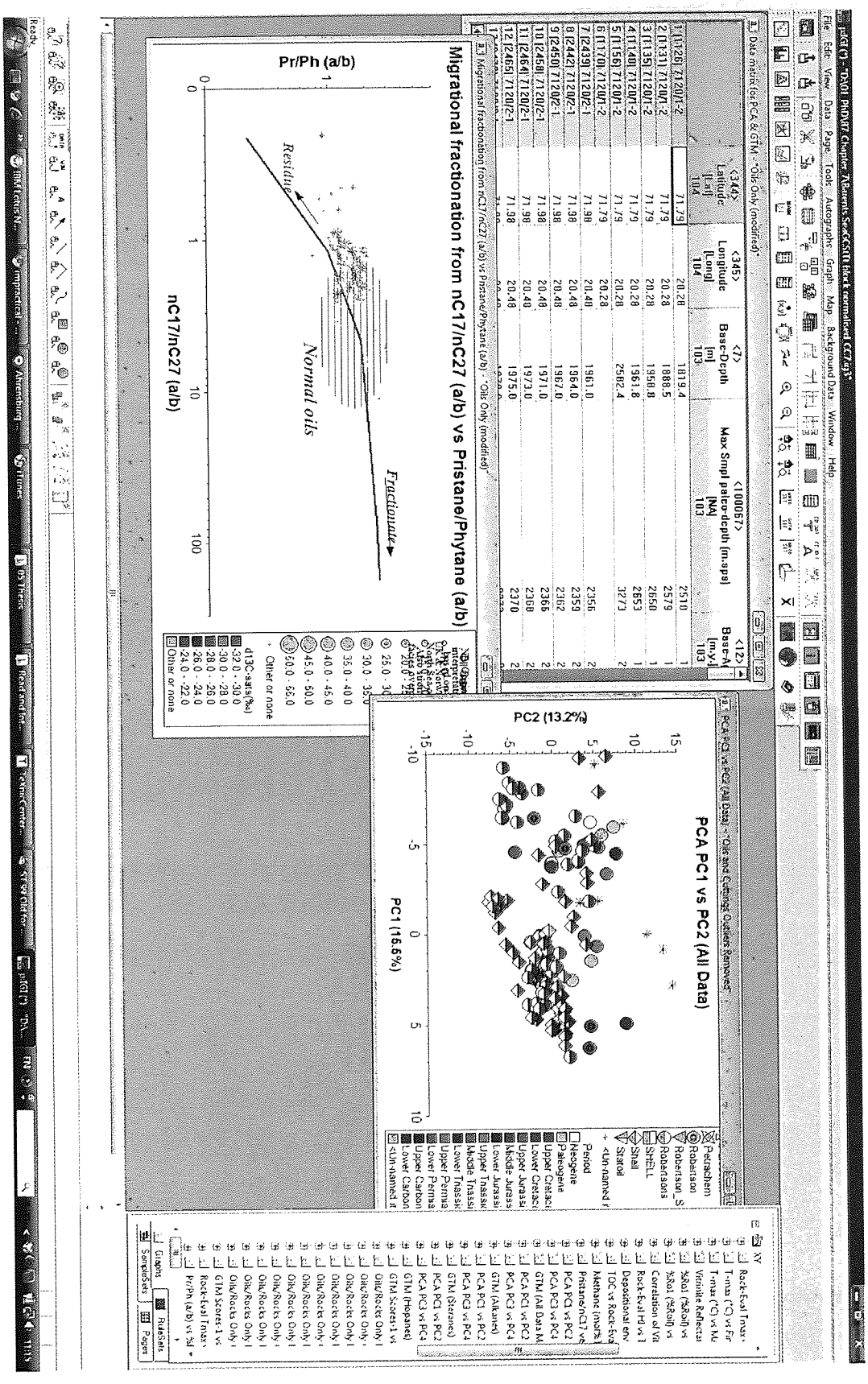


Figure 7.5: pIGI Utility

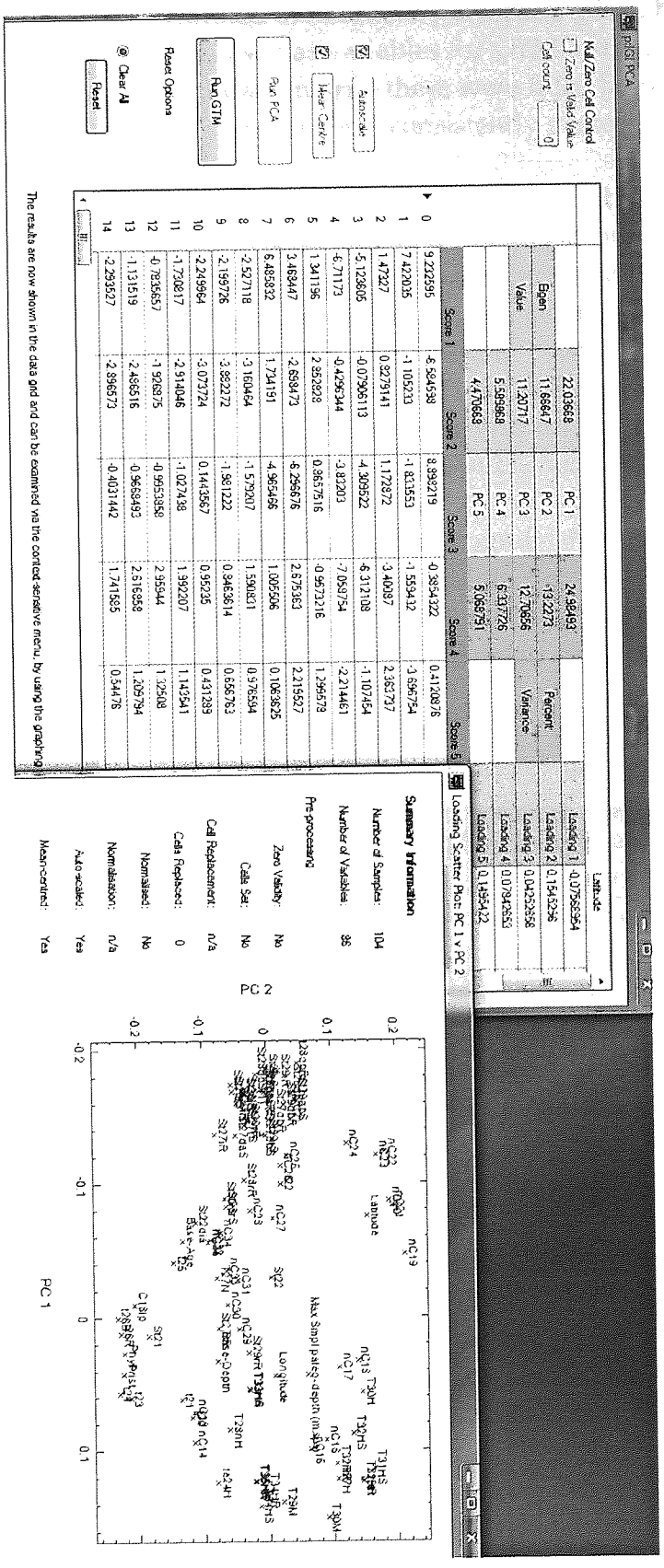


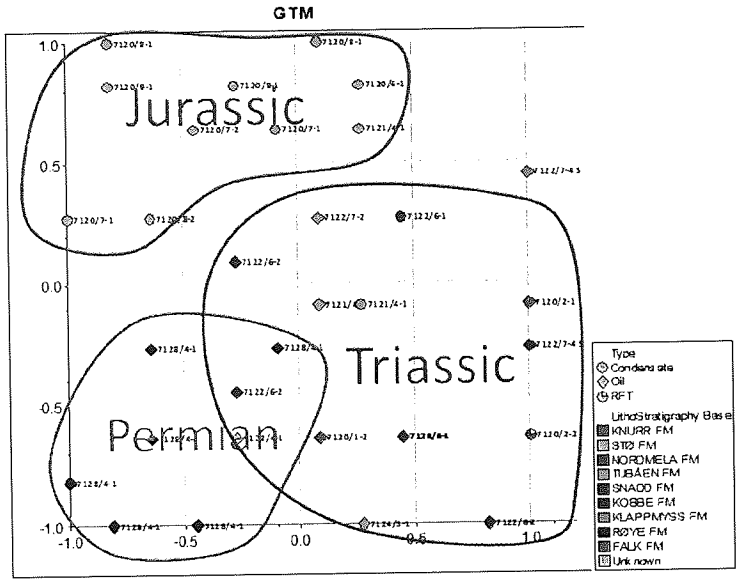
Figure 7.6: pIGI PCA Utility.

heights and areas) for the saturated biomarkers were incomplete, hence a selection of 11 molecular ratios were used as variables for GTM testing. Several gaps nonetheless remain in the input data matrix: these were filled by using Bayesian PCA (Oba *et al.*, 2003), because this method consistently performed as good or better than all other tested methods.

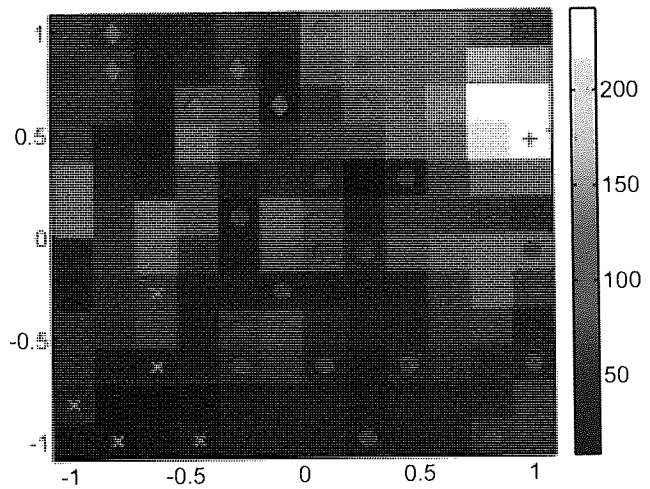
The apparently random scatter of oils and condensates on the GTM plot (Figure 7.7(a)) belies the demonstrable coherence of sample distribution highlighted by the use of colour and symbols. Condensates from the Snøhvit and Askeladd fields reservoired in Jurassic Stø Formation sands are clustered in the top left quadrant of the plot. By comparison, samples found in Paleozoic intervals (Permian Røye Formation) are clustered in the bottom left hand quadrant of the plot. Triassic oils are grouped in the intermediate region and further to the right side of the plot. These clusters are in agreement with published data describing oil families identified in the Barents Sea petroleum province (Ohm *et al.*, 2008). An interesting anomaly was identified by using the magnification factors in Figure 7.7(b), which indicate that one oil in the upper middle right is a clear outlier. A close inspection showed that the oil does not match the signature of the other oils. As explanation for the mismatch we suspect some kind of contamination or error in the lab analysing the oil.

The GTM plot illustrates generic clustering and bulk physical character of the fluids, but the roots of this 2-D visualisation of variation in the data lie in the input variables which can be visualised using parallel coordinates. Figure 7.8 shows the parallel coordinates for the 3 different groups. Within the scatter clear trends can be identified. The three clusters of oils show large overlap for the majority of analysed ratios with the exception of a few key variables. For example the ratio of isoprenoids/n-alkanes (i.e. Pristane/nC17 and Phytane/nC18) clearly distinguishes the Permian-reservoired oils from the other two groups. The ratio of C24 tetracyclic terpane/C30 Hopane separates Triassic from Jurassic oils, probably on the basis of lithofacies, since this ratio is source-sensitive (Peters *et al.*, 2005). The isoprenoid/n-alkane ratios are not particularly diagnostic but GTM does allow distinction of oils groupings on the basis of non-diagnostic biomarkers and therefore provides a basis for more detailed oil-source correlation studies.

Comparison to PCA: To highlight the supplementary character of GTM and allow for a comparison the same data were analysed with PCA. Before using PCA the missing data were treated using BPCA because this method performed best next to BGTM. The results shown in Figure 7.9a indicate that the visualisation of PCA and GTM are quite similar. The score plot of the first two principal components allows for a similar separation of classes as GTM. However the outlier identified in the through the magnification factors in GTM and here labeled as "'Uncategorised"' is not as apparent as in GTM. When looking at the third principal component in Figure 7.9b it is apparent that this component does not describe any effects useful for separating the classes. This is surprising since the third principal component still accounts for roughly 12% of the overall variance as can be



(a)



(b)

Figure 7.7: a) GTM visualisation for the Barents Sea oils. b) Corresponding magnification factors for the GTM visualisation. In the GTM visualisation one can clearly distinguish between the three classes of oils depending on their origin (Jurassic, Permian, Triassic). When looking at the magnification factor plot it is further apparent that the sample in the upper right of visualisation is a clear outlier.

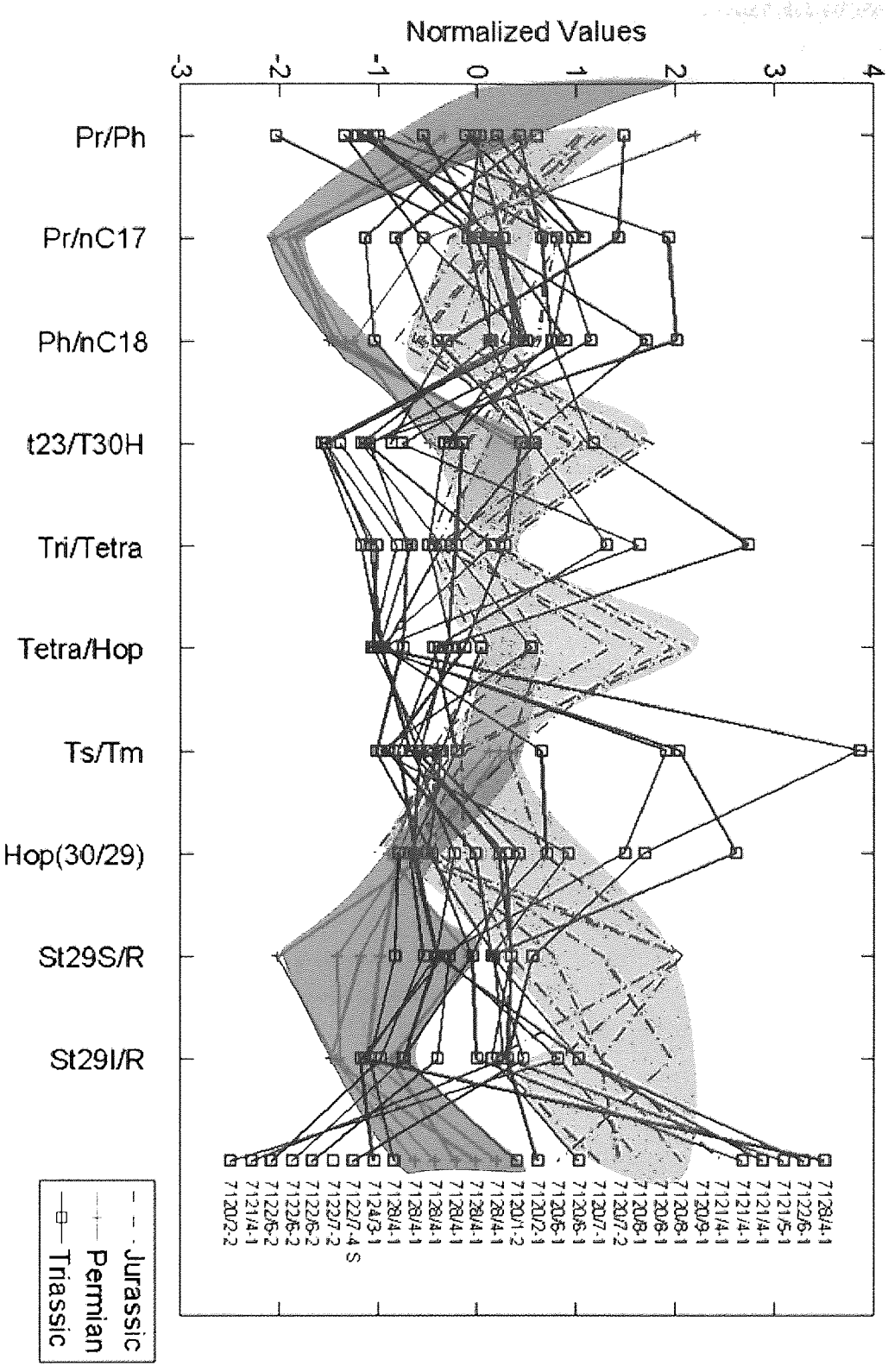


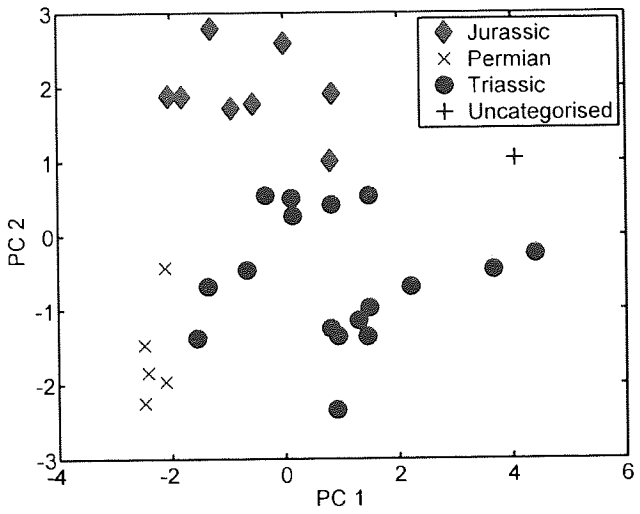
Figure 7.8: Parallel coordinates plot for the different clusters identified in the GTM visualisation. In the plot one can identify that the Pristane (Pr) and Phitane (Ph) ratios can be used to distinguish the Permian from the other two classes. Similarly the sterane ratios St29S/R and St29I/R can be used to discriminate between the Jurassic and the other two classes.

seen in 7.10b. It could be speculated that this component is mainly describing intra class noise/variance. The loadings plot in Figure 7.10a shows similar results when compared to the prallel coordinate plot of GTM. The first principal component is mainly distinguishing between the Permian and the Triassic and in the parallel corrdinate plots one could identify the Pristane and Phitane ratios as highly discriminative. This is mirrowed by the very high values of Pr/nC17 and Ph/nC18 in the loadings plot. Similarly the sterane ratios St29S/R and St29I/R have the highest absolute value for the second principal component in the loadings plot. The second principal component discrimates between the Jurassic and the other two classes. This is also mirrowed in the parallel coordiante plots where the sterane ratios also showed that they can be used to distinguish between the Jurassic samples from the others. In conclusion the results for PCA and GTM are quite similar, with the exception that GTM allows for the better identification of the outlier. The argument would therefore be to use GTM and PCA in conjunction to cross check the obeservations and get a better understanding of the data.

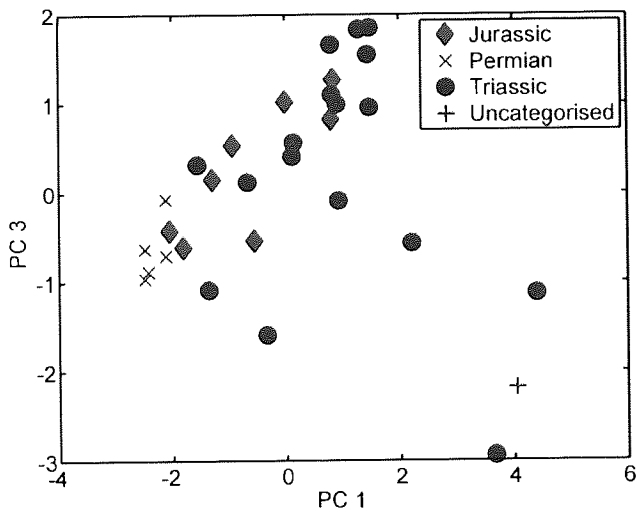
7.5 Benchmark Study

To measure the performance of the visualisation and imputation algorithms on real data sets we took two confidential but complete data sets from IGI Ltd. The first data set is based on oils from the North Sea and the second data set is based on oils from a basin in Africa. On both data sets we ran two experiments. The first experiment was to calculate the RMSE following the cross-validation leave-one-out framework described in chapter 5. In this experiment we tested the different visualisation algorithms which can deal with missing data.

In the second experiment we deleted a certain proportion of the data to generate different missing data patterns like in chapter 6. To provide a robust result with respect to the missing data pattern the results were averaged over the 20 random missing data patterns. The proportions of missing data p_i are between $p_i = 0.1$ and $p_i = 0.5$.

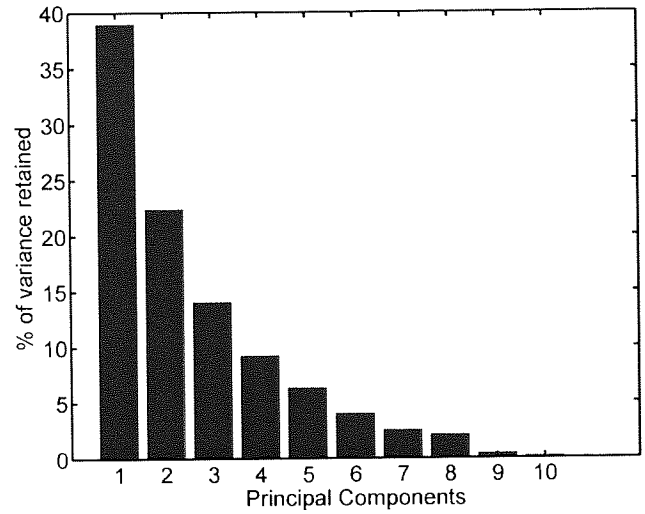


(a)

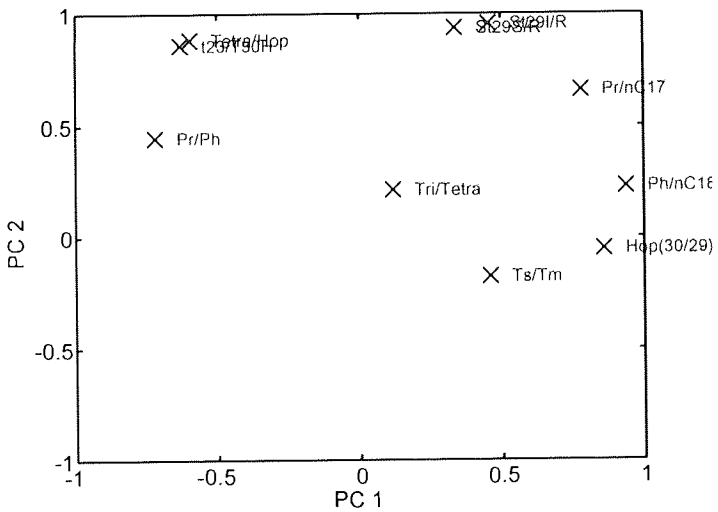


(b)

Figure 7.9: a) The scores plot for the first two principal components showing a similar distinction of classes as the GTM. b) The scores plot for the first and third principal component.



(a)



(b)

Figure 7.10: a) Histogram showing the contribution of each principal component towards the variance. b) The loadings plot showing how much each variable contributed towards the first two principal components relative to the other variables.

7.5.1 African Data

This data set is based on samples of oils from a reservoir in Africa. The data samples were analysed at two different laboratories: one analysed 36 samples and the other 40 samples. The variables are peak height measurements from a GC-MS including alkanes, steranes and hopanes. In total the data set comprises of 72 variables and 76 samples. The block structure of the data was defined by looking at the heat-map of the absolute values of the correlation coefficients which were ordered by the OLO algorithm.

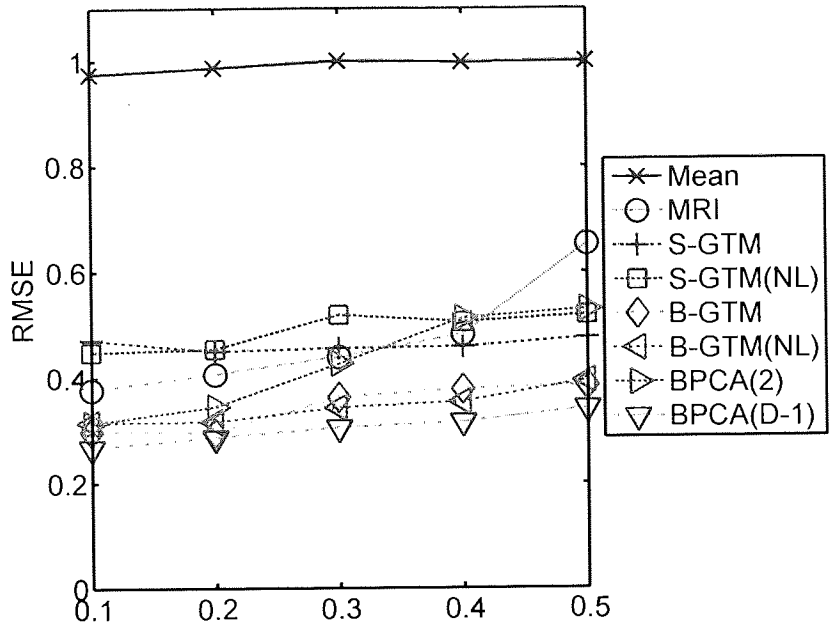
The results of the first experiment can be seen in table 7.1 where S-GTM, B-GTM and BPCA are compared with each other given different initialisations and different numbers of retained principal components respectively. S-GTM performs better with a non-linear initialisation while in the case of B-GTM there is no difference between the initialisation with PCA and Isomap. Both S-GTM and B-GTM perform better than BPCA in the case of two or three retained principal components. It is therefore likely that they give a more accurate representation of the data than PCA with two or even three principal components. In the case where 71 of the 72 dimensions are retained for principal components analysis only B-GTM performs better than BPCA but retaining so many components does not support visualisation.

The mean results of the imputation experiment can be seen in Figure 7.11. The results show that BPCA where 71 components are retained always outperforms all other methods. Retaining 71 variables is done to maximise the imputation performance of the model. BPCA, due to the limitation of the algorithm, can only retain a maximum of $D - 1$ factors in the model, thus 71 in this case. In the case where only two principal components are retained B-GTM which was initialised with PCA outperforms BPCA (the case with three retained principal components is not shown because it is very similar to the case with two principal components). Following with a gap in performance are MRI and S-GTM. When looking more closely at the results it can be seen that the non-linear initialisation is actually not beneficial to the imputation with S-GTM or B-GTM. However this could be due to the fact that the Isomap algorithm cannot deal with missing data naturally. As with the PCA initialisation a mean-imputation was performed to use the data with Isomap but not bias the results.

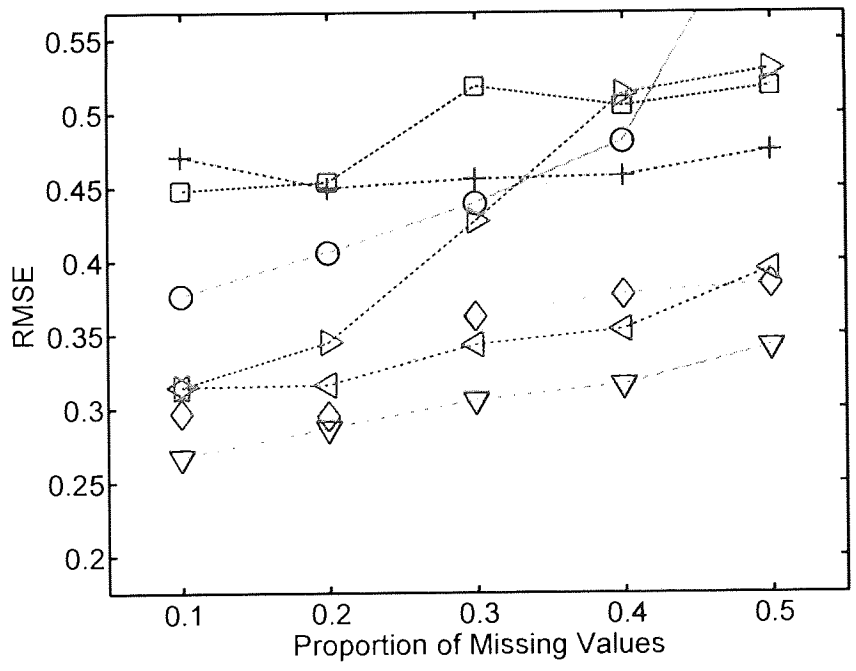
The results also show that in the case of only two retained principal components BPCA deteriorates very rapidly, with more than 20% of missing data. However BPCA, retaining $D - 1$ principal components, and B-GTM stay relatively stable. The boxplots for the results of the different imputation methods for different levels of missing data can be seen in Figure 7.12. The boxplots show in essence the same results as summarised by the mean plots in Figure 7.11.

Model	RMSE
S-GTM (PCA)	0.2
S-GTM (Non-Linear)	0.15
B-GTM (PCA)	0.09
B-GTM (Non-Linear)	0.09
BPCA (2)	0.33
BPCA (3)	0.24
BPCA (D-1)	0.12

Table 7.1: RMSE for African data leave-one-out-cross-validation. The information in brackets relates to the initialisation in case of GTM and to the number of retained principal components in case of BPCA.

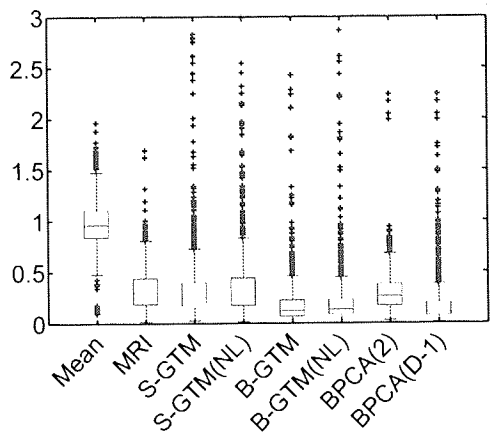


(a)

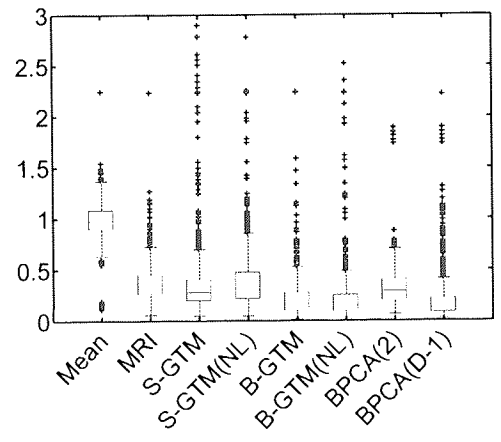


(b)

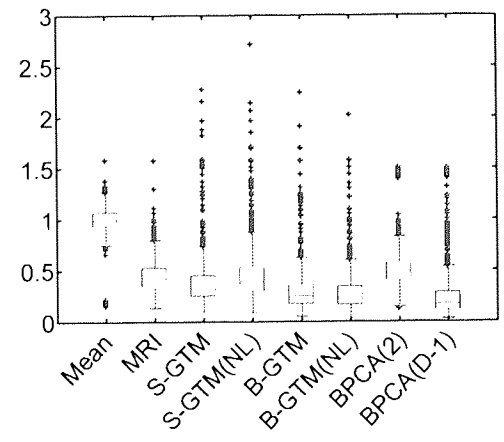
Figure 7.11: African Data: (a)Average imputation results for different amounts of missing data. (NL) stands for the initialisation with Isomap in case of the GTM. BPCA is shown for the cases where two and 71 principal components are retained. (b) Area of interest in plot (a). It is apparent that BPCA retaining all principal components outperforms all other methods with B-GTM becoming second. BPCA retaining two principal components becoming third at least in the cases where less than 30% of the data are missing.



(a) p_i=0.1



(b) p_i=0.2



(c) p_i=0.4

Figure 7.12: African data: boxplots for different proportions of missing data p_i showing the spread of the RMSE for the different imputation methods. They verify that the results given by Figure 7.11 are not skewed due to unnatural outliers in the average performance.

7.5.2 North Sea Data

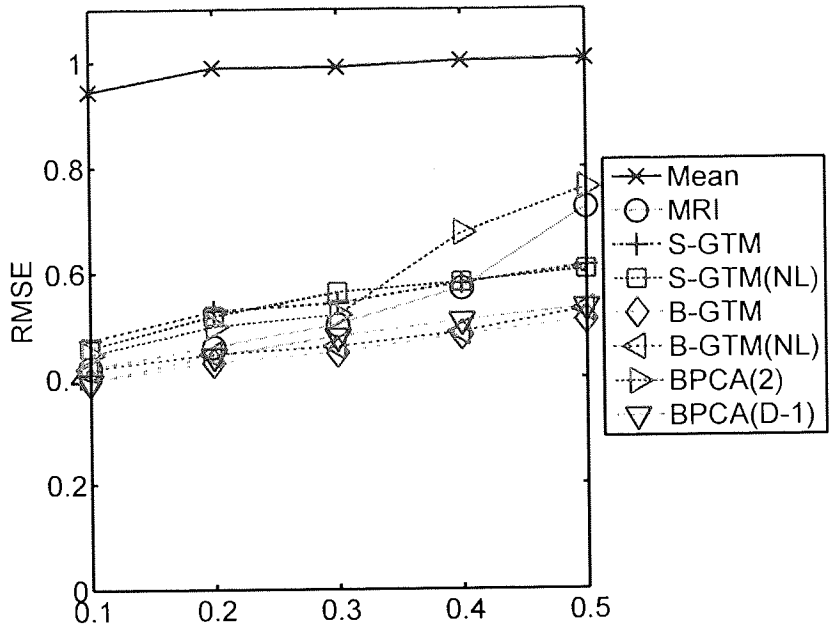
This data set is based on samples of oils from the North Sea. The data samples were all analysed at a single laboratory. The variables are peak height measurements from a GC-MS including alkanes, steranes and hopanes. In total the data set comprises of 67 variables and 132 samples. The block structure of the data was defined by looking at the heat-map of the absolute values of the correlation coefficients which were ordered by the OLO algorithm.

The results of the first experiment can be seen in table 7.2 where S-GTM, B-GTM and BPCA are compared with each other for different initialisations and different numbers of retained principal components respectively. In this data set S-GTM and B-GTM perform better with a non-linear initialisation using Isomap. Again both S-GTM and B-GTM perform better than BPCA in the case of two or three retained principal components. Also in the case where 66 principal components are retained only B-GTM performs better than BPCA.

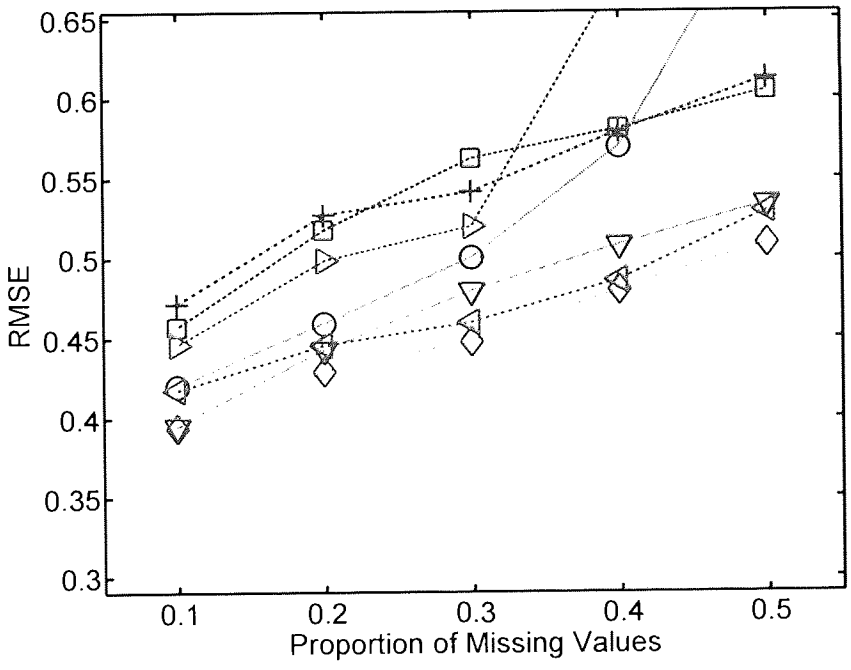
The mean results of the imputation experiment can be seen in Figure 7.13. The results show that this time B-GTM, initialised with PCA, nearly always outperforms all other methods or in the case of only 10% of missing data performs as well as BPCA, retaining 66 principal components. In this data set BPCA, retaining only two principal components, even performs worse than MRI. However it still performs better than S-GTM. When looking more closely at the results the non-linear initialisation seems to have a negative effect. We suspect that this is due to the mean imputation we perform to initialise Isomap. However this needs to be confirmed by doing a series of experiments with different levels of missing data and different initialisation methods. Again the results show that BPCA in the case of only two retained principal components deteriorates very rapidly. The boxplots for the results of the different imputation methods for different levels of missing data can be seen in Figure 7.14. Again the boxplots show in essence the same results as summarised by the mean plots in Figure 7.13.

Model	RMSE
S-GTM (PCA)	0.51
S-GTM (Non-Linear)	0.31
B-GTM (PCA)	0.21
B-GTM (Non-Linear)	0.18
BPCA (2)	0.45
BPCA (3)	0.40
BPCA (D-1)	0.22

Table 7.2: RMSE for North Sea data leave-one-out-cross-validation. The information in brackets relates to the initialisation in case of GTM and to the number of retained principal components in case of BPCA.

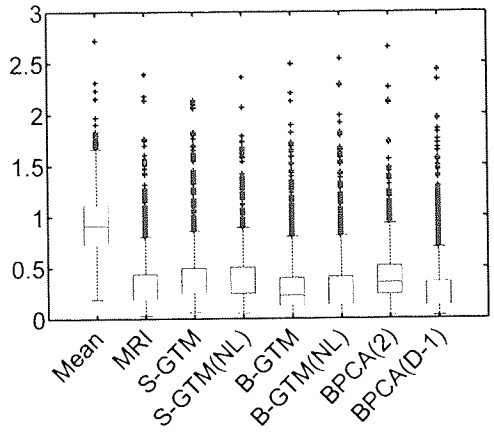


(a)

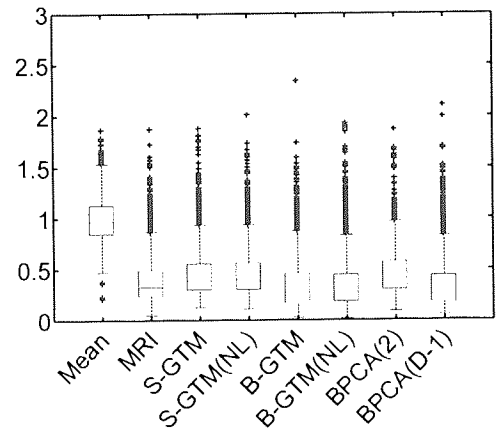


(b)

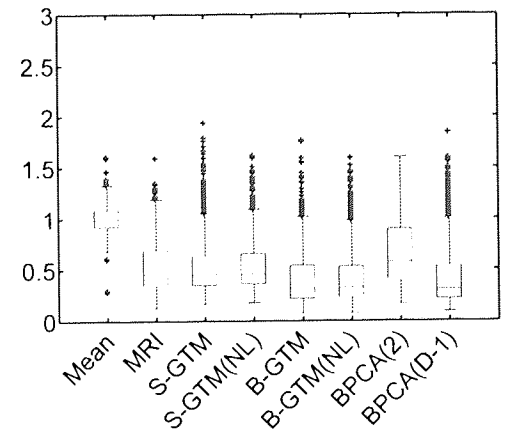
Figure 7.13: North Sea Data: (a)Average imputation results for different amounts of missind data. (NL) stands for the initialisation with Isomap in case of the GTM. BPCA is shown for the cases where 2 and 66 principal components are retained. (b) Area of interest in plot (a). It is apparent that BPCA retaining all principal components and B-GTM perform equally well on this data set, with a slight advantage for B-GTM once higher amounts of data are missing. BPCA retaining only two principal components performs even worse than MRI and deterioates quickly once the amount of missing data goes beyond 30%.



(a) p_i=0.1



(b) p_i=0.2



(c) p_i=0.4

Figure 7.14: North Sea data: boxplots for different proportions of missing data p_i showing the spread of the RMSE for the different imputation methods. They verify that the results given by Figure 7.13 are not skewed due to unnatural outliers in the average performance.

7.6 Performance Study: Speed of the methods

To evaluate the difference in speed of the different methods a small performance experiment was done. The benchmark environment was Windows Vista and Matlab 7.9.0 on a Pentium Core 2 Duo P8600@2.40 GHz and 4GB. The different methods had a limit of 100 iterations but would stop earlier if the respective termination criterion was reached (commonly less change than 10^{-3} in their respective utility functions). The results of the experiment can be seen in Table 7.3. The results are not surprising and show that the visualisation S-GTM and especially B-GTM always take the longest to run, except in the case of the toy data sets with many samples where GPLVM takes the longest (because the inversion of the matrix scales with the cube of number of samples). In the case of S-GTM and B-GTM the algorithm scales with the number of dimensions and the number of internal Gaussian nodes in the grid (which was chosen to be 25×25). The B-GTM algorithm takes much longer than the S-GTM because due to the block structure one can not write the code as matrix and vector operations, which have a significant speed gain in Matlab.

In the case of the imputation algorithms the results are similar. Here the S-GTM and B-GTM algorithm both take far longer because due to the missing data one needs to use indicators and even more loops, which is penalised in Matlab with a longer runtime. However the iterative nature of the algorithm makes GTM generally slow because the EM algorithms takes numerous (normally between 25 to 100) iterations to converge to a good solution. This process is drastically prolonged by the implemented heuristics, which make the algorithm more stable but also slower since the heuristics take considerable amounts of computer time. Table 7.4 shows the average results for single steps in the algorithm on every data set. The results show that depending on the dimensionality of the data set the heuristics take 37% to 99% of the computer time. There is certainly much scope for improving the speed of the heuristics. Either by tuning the current heuristics or by using other ones, which run faster.

There is an anomaly with the African data set and to some extent the North Sea data set, which have an enormous runtime. This was caused by a problem with the Cholesky decomposition in one of the heuristics. In certain cases Matlab runs out of memory and catching this error prolonged the process considerably.

The considerable difference in speed is an important point. One needs to keep these results in mind when planning to implement the algorithms in any kind of application since most user don't want to wait many seconds or even minutes to see a result after they have clicked a button. The very long runtime is also an issue if one wants to do benchmark studies against other algorithms. Benchmarking with many iterations and with different parameters is only possible with a cluster computer and by distributing the experiment across different nodes.

Finally it is important to note that these experiments were conducted by using Matlab, which is highly optimised for mathematical calculations involving matrix operations. If one simply implements the algorithms in C++ without using a highly optimised matrix library like BLAS in conjunction with an optimised li-

Data set:	Swiss Data	Oil Data	BGTM D20	Africa	North Sea
Samples:	1000	1000	100	76	132
Variables:	3	12	20	71	67
Method	Time in Seconds				
PCA	0	0	0	0	0
Isomap	31	33	0	0	0
GPLVM	69	237	3	7	16
GTM	71	158	767	4800	100
B-GTM	97	617	683	1393	960
Imp. Mean	0	0	0	0	0
Imp. WMI	0	1	0	0	0
Imp. MRI	0	0	0	2	1
Imp. BPCA	3	10	1	68	81
Imp. S-GTM	64	27	30	909	119
Imp. B-GTM	844	278	115	974	564

Table 7.3: Runtime until termination of the algorithm for the different methods in seconds on the different data sets.

brary providing advanced matrix operations like LAPACK one might experience even longer runtimes than presented in this benchmark.

Data set:	Swiss Data	Oil Data	BGTM D20	Africa	North Sea
Samples:	1000	1000	100	76	132
Variables:	3	12	20	71	67
S-GTM	Time in Seconds				
EM-Step	0.25	0.27	0.03	0.04	0.08
Heuristic(H.)	0.15	0.68	5.53	36.4	15.8
% of step for H.	37	71	99	99	99
B-GTM	Time in Seconds				
EM-Step	0.25	.56	0.12	1.55	1.04
Heuristic(H.)	0.28	3.1	4.74	39.8	40.0
% of step for H.	45	75	97	96	97

Table 7.4: Table showing the average runtime in a single step for two different parts of the GTM algorithm. The analysed parts are the two most computationally intense parts, namely the EM-Step and the added heuristic to stabilise the algorithm. It is apparent that the heuristics take up most of the time when the dimensionality of the data increases.

7.7 Summary

In this chapter we successfully applied the developed non-linear methods to real data from geochemistry. It was demonstrated how one can explore and understand the non-linear mappings by using plotting mechanism like the local parallel coordinates and diagnostics like magnification factors. Integrated into tools like DVMS these utilities can give users an interactive experience when exploring the manifold and their data. The result is a better understanding of the data structure and the underlying variables/processes. The local parallel coordinates technique is a particularly useful feature because it will give the users a diagnostic akin to the loadings plots in PCA.

The successful use of GTM to distinguish clusters in a data set based on oils from the Barents Sea suggests that the method has great potential for the rapid and accurate distinction of trends in large, complex datasets frequently encountered in petroleum geochemistry and further afield.

The novel methods B-GTM and VSRMI were further successfully benchmarked on real geochemical data sets. The benchmark results are very similar to the results obtained on the toy data sets. They show that in the case of a strong block structure B-GTM always outperforms S-GTM. This is the case for visualisation and imputation. The second extension VSRMI improves the results of GTM on the real data sets as well. On both real data sets, the initialisation with Isomap gave better or similar results for the visualisation. However when one uses GTM for imputation VSRMI does not seem to make a big difference. The reason for this is that one needs an initial mapping to utilise VSRMI. However to obtain this mapping one needs to use either PCA, Isomap or another projection method. Since we are dealing with missing data one now has to employ data imputation

as a pre-processing step. In our experiments we used the mean imputation so as not to degrade the results and this initial mapping.

In general the increase in performance of B-GTM and VSRMI comes at a higher computational cost. The VSRMI algorithm needs to be initialised by a projection which adds the cost of this algorithm to the overall computational cost. Where implemented in Matlab the B-GTM algorithm itself is far slower than the S-GTM algorithm because we have to utilise loops to evaluate the block structure. In Matlab, loops are far slower than the equivalent operations written as matrix and vector operations. Additionally we use heuristics to make the B-GTM more stable and robust, which also add to the computational overall cost. Depending on the data set this makes the algorithm 5 to 10 times slower than S-GTM.

8 Discussion and Future Work

CONTENTS

8.1 Discussion 159
8.2 Future Work 161
8.3 Summary 163

8.1 Discussion

In this thesis we extended the well-known GTM algorithm with a novel method to deal with the particular characteristics of geochemical data sets, namely the block structure of highly correlated variables in the covariance matrix. Further we proposed a new method for initialising GTM which makes it possible to use any available projection of the data. Both methods were extensively tested and validated in many experiments.

Novel extensions (chapter 5 and 7): In the first set of experiments we used toy data sets where we could control the amount of block covariance structure and the dimensionality of the data. The results of the experiments show that B-GTM improves both the model fit and the visualisation. Given the right block structure the algorithm performs as well as or better than spherical GTM or GTM with a full covariance matrix. However if the block structure is misspecified the performance of B-GTM deteriorates quite rapidly. The method of choice for finding the block structure is the OLO algorithm in combination with an expert practitioner who can identify sensible groupings in the variables.

Further the experiments showed a not broadly documented problem; the GTM algorithm with a more complex covariance structure is limited to only a moderate number of dimensions (less than 40). The problem is partly due to numerical errors and partly due to singularities in the likelihood function. However this limit could be extended by using additional heuristics to circumvent these problems with the EM algorithm and it was successfully tested for data sets with up to 72 dimensions.

The experiments further show that VSRMI is a very useful addition to the GTM algorithm. It allows GTM to exploit local methods like Isomap or other alternative projections to act as initialisation. The results clearly show that this approach is highly beneficial in cases where PCA is too restrictive to pick up the structure in the data. Using alternative mappings like Isomap it was possible to improve the model fit and in some cases decrease the time to fit the model as well. The VSRMI might even open up the possibility to use GTM to assess how non-linear a data set is. For this purpose one would need to initialise the GTM with a linear and a non-linear mapping and directly calculate how well the GTM is fitting the data without using the EM algorithm to increase the fit to the data.

Further both extensions of GTM were validated and successfully applied to real data sets from geochemistry. It was demonstrated how this non-linear method can be used to explore geochemical data and how one can draw inference about the data. This was done by introducing advanced diagnostics like the local parallel coordinate plotting, magnification factors and plotting of the modes. It was shown that the integration of these diagnostics into an interactive tool like DVMS is a powerful utility that geochemists could use to explore the structure of their data.

Missing data imputation (chapter 6): The GTM algorithm also has the advantage that it can deal with missing data. An extensive experiment with missing data was conducted which showed that especially the extension to B-GTM has very good imputation characteristics. Previous work has shown that GTM can successfully cope with missing values (Vicente *et al.*, 2004; Olier and Vellido, 2005; Sun, 2002) but no extensive comparison to other methods has been conducted. Motivated by incomplete data sets which are common in geochemistry, in this thesis B-GTM was benchmarked against mean imputation, multiple regression imputation and BPCA, which were chosen as representatives from different categories of imputation algorithms. The only method which performed as well as B-GTM was BPCA. The experiments with missing data showed that GTM is not only a good model for data visualisation but can also be used for imputing missing values. However from the results it is evident that missing values impair the visualisation considerably. In the toy data sets GTM was able to pick up most of the structure until 20% of the data was missing. This implies that the method can be used on data sets with real levels of missing values and still identify interesting and useful structure.

Further, the experiments with missing data helped to motivate a new methodology for assessing unsupervised learning algorithms for data visualisation which can deal with missing data. Using a combination of the RMSE and leave-one-out-cross-validation we created a measure which has similar characteristics to the likelihood of a probabilistic model but which can be used with non-probabilistic models (e.g. NIPLAS for PCA or missing data algorithm for kernel PCA). Further the measure has the advantage of having a clearly defined optimal value, which is zero, unlike the likelihood where the optimal value is often not known.

In conclusion we can claim to have achieved the main goals as defined in chapter 1. However during the course of this thesis it become apparent that there are many more questions which need to be answered in order to establish non-linear methods like GTM or any of its extensions as standard tools for data visualisation in geochemistry. A comprehensive overview over these possible areas of research will be given in the next and final section.

8.2 Future Work

To make non-linear methods easily usable and accessible for non-mathematical application experts ,who want to explore their complex data, more work is still needed. At the current stage of development the non-linear methods should be used with guidance from an expert in the field of chemometrics or multivariate analysis if they are applied to a geochemical data set. One still has to make many critical choices about parameters like the number of RBF functions, the size of the grid, the block structure of the covariance matrix or the method of choice for the initialisation. Also the algorithms need to be extensively tested to determine their limits, especially in regards to the maximum number of variables and the minimum number of samples. Additionally there are many more small research ideas, which could not be followed up because of the time constraint inherent in a Ph.D. thesis.

In the opinion of the author these are the most interesting areas where some research could be done:

- Future work can be done to integrate the prior knowledge about the covariance matrix in a more principled way into the algorithm. At the moment the enforcement of the block structure is very rigid and ad-hoc. One could imagine an approach where priors are placed on the covariance matrix and the procedure is integrated in a Bayesian way into the GTM model. A downside of such an approach might be the significant time it would take to fit such a model.

- To determine the block structure and the relation between the variables one could try to develop an automated approach by using Bayesian graphical models as an alternative to OLO.

- Another area of interest is the definition and assessment of metrics or measures to assign a value to the quality of the visualisation in general or the goodness of the fit in case of the GTM and other probabilistic models. We have done some work on this introducing the RMSE in combination with a leave-one-out-cross-validation approach, however there are other possibilities like using clustering algorithms and Kullback-Leibler divergence.

- As pointed out in the summary of chapter 5 the RMSE in combination with leave-one-out-cross-validation might have an application as a measure of how non-linear a data set is. However to validate this one needs to design sensible experiments and define non-linearity in a more principled and measurable way.

- If the development for a measure of non-linearity is successful one could look at the the possibly non-linear relations in various geochemical data sets to further the understanding of geochemical processes and help in the development of new classification boundaries considering oil-oil and source rock-oil correlation studies.

- Olier and Vellido (2008b) did some work on placing a Gaussian process prior over the mapping function in GTM, which replaces the RBF network and thus makes it unnecessary to specify the number of RBF functions. It would be interesting to test this algorithm on his applicability on geochemical data sets since this development potentially makes GTM more automated and easier to use.

- Vellido (2006a) and Maniyar and Nabney (2006a) did work on variable selection or feature saliency. This diagnostics identifies significant features and thus provide similar information to the loadings in PCA. It would be interesting to research the applicability of these methods to geochemical data sets and compare them against each other.

- During the research it has become apparent that the GTM algorithm is not coping well with very high-dimensional data because of numerical problems and singularities in the likelihood function. An alternative approach would be to use alternative heuristic models which are based on GTM. By using or modifying these heuristics it might be possible to circumvent these problems. However, an extensive benchmark study and more research would be needed to compare GTM against heuristics like the topographic neural gas algorithm (Pena and Fyfe, 2006) to evaluate how these perform in high dimensional data.

- An important question is: how stable is the GTM algorithm on high dimensional data? Research in this area should have the aim to determine and specify the reasons for the breakdown of the algorithm. It is unlikely but the optimal result would be a theoretical limit for the maximum dimensionality of the GTM algorithm. However, as first step an empirical study could help to generate rough guidelines that assist the practitioner in choosing the right parameters and in validating the model when using the algorithm.

- With respect to missing data an interesting area of research is the 'multiple imputation' approach. The multiple imputation framework is used to assign a value for the uncertainty (variance) of the estimates for the missing values. In the case of GTM it might be possible to use the Gaussian process prior over the mapping function to directly calculate this estimate.

- To use missing data as a measure for benchmarking visualisation methods one would have to asses how other visualisation methods like Isomap and Neuroscale could be extended to deal with missing data. In the case of Neuroscale one possible approach is to ignore the dimensions which have missing data on a case by case basis when calculating the distances. Similarly one could modify the algorithm constructing the graph for Isomap to just ignore the dimensions with missing data on a case by case basis. More sophisticated approaches might include the integration of local imputation techniques or imputation heuristics similar to local conditional densities.

- Especially when looking at the imputation in geochemical data one should consider looking at the log transform (Reimann *et al.*, 2002). Currently the imputation methods have no prior knowledge about the data structure, but it is known that there are no negative values in geochemical data. However, the imputation methods will regularly produce negative values when estimating missing data. Using the log transform one could circumvent this problem. It would be interesting to see if this could improve the overall imputation results.

- Another interesting area related to missing data is the treatment of zeros in geochemical data sets. Normally a zero value is not really zero but was just too small to be picked up by the measurement device. Modifying imputation methods to treat those missing values (implying a prior on the range of the values will be crucial) and comparing the results against already published approaches to treat zeros in compositional data (Farnham *et al.*, 2002; Thió-Henestrosa and Martın-Fernández, 2003; Palarea-Albaladejo and Martín-Fernández, 2008) could be an area of possible research.

- An important area of research is the intra-laboratory difference when analysing samples. The existence of the difference is well known and published (Blankenhorn *et al.*, 1992; Isaacs, 2001; Kucklick *et al.*, 2002), however no research has been done in regards to removing the bias computationally and on how this bias is effecting multivariate analysis. This is especially important in geochemistry where one has only small data sets and might want to augment their own data set with publicly available data from the same or similar regions.

- To address the problem of non positive definite covariance estimates in the block and full GTM algorithm one could consider the parametrisation of the matrix. Instead of estimating the covariance matrix directly one could try to estimate directly the cholesky decomposition matrix C given that $\Sigma = CC^T$.

8.3 Summary

In this thesis we have laid the foundation for the use of non-linear and probabilistic methods in geochemistry. We have discussed the advantages of using probabilistic models; namely the possibility to deal with missing data, the use of artificial missing data as means to assess the model fit, the availability of a noise model to deal with uncertainty and the possibility to tune this noise model to improve the model fit. We have discussed the advantages of of using a non-linear model; i.e. one can capture more complex data structures, which might stay hidden when only relying on linear models like PCA. Further we have demonstrated how one can improve the initialisation of GTM by using other non-linear methods like Isomap.

We have shown the need for more research to assess the the restrictions and limitations of these models. It is unsatisfactory to not know when probabilistic models like GTM become unstable when used with high-dimensional data. More benchmarks on real data sets are needed to assess how well GTM is capturing the structure and how much improvement can be achieved by using alternative initialisations like Isomap.

For practitioners who want to integrate these models into industrial work environments one has to work on the automatisation of the models. More work is required on the performance of the heuristics, which is insufficient at the moment because they take far too long.

However we are confident that non-linear and probabilistic models are a great addition to the toolbox of the geochemical practitioner. We hope that in the future they will be a regularly used tool to understand and explore geochemical data.

9 References

References

Andersson, C. and R. Bro 2000. The N-way toolbox for MATLAB. *Chemometrics and Intelligent Laboratory Systems* **52** (1), 1–4.

Andrade, A., S. Nasuto, P. Kyberd, and C. Sweeney-Reed 2005. Generative topographic mapping applied to clustering and visualization of motor unit action potentials. *Biosystems* **82** (3), 273–284.

Arabie, P. and L. Hubert 1996. *Clustering and classification*, Chapter An overview of combinatorial data analysis, pp. 5–63. World Scientific.

Ash, R. 1990. *Information theory*. Dover Publications, Inc., New York.

Attias, H. and L. Ar 1999. Inferring parameters and structure of latent variable models by variational Bayes. In *Proc 15th Annu Conf Uncertainty in Artificial Intelligence*, Volume 30. San Francisco, CA: Morgan Kaufmann Publishers.

Bar-Joseph, Z., E. Demaine, D. Gifford, N. Srebro, A. Hamel, and T. Jaakkola 2003. K-ary clustering with optimal leaf ordering for gene expression data. *Bioinformatics* **19** (9), 1070–1078.

Belkin, M. and P. Niyogi 2003. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation* **15**, 1373–1396.

Bernard, J., R. McCulloch, and X.-L. Meng 2000. Modeling Covariance Matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica* **10**, 1281–1311.

Bishop, C., G. Hinton, and I. Strachan 1997. GTM through time. In *Artificial Neural Networks, Fifth International Conference on (Conf. Publ. No. 440)*, pp. 111–116.

Bishop, C. and G. D. James 1993. Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. *Nuclear Instruments and Methods in Physics Research* **A327**, 580–593.

Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*. New York,N.Y.: Oxford University Press.

Bishop, C. M., M. Svensen, and C. K. I. Williams 1996. GTM: a principled alternative to the self-organizing map. *Artificial Neural Networks ICANN 96* **9**, 165–170.

Bishop, C. M., M. Svensen, and C. K. I. Williams 1998. Developments of the Generative Topographic Mapping. *Neurocomputing* **21**, 203–224.

Blanchard, G., M. Kawanabe, M. S. V. Spokoiny, and K.-R. Müller 2006. In Search of Non-Gaussian Components of a High-Dimensional Distribution. *The Journal of Machine Learning Research* 7, 247 – 28.

Blankenhorn, I., D. Meijer, and R. Delft 1992. Inter-laboratory comparison of methods used for analysing polycyclic aromatic hydrocarbons (PAHs) in soil samples. *Fresenius' Journal of Analytical Chemistry* **343** (6), 497–504.

Bose, I. and X. Chen 2009. A method for extension of generative topographic mapping for fuzzy clustering. *Journal of the American Society for Information Science and Technology* **60** (2), 363–371.

Brás, L. and J. Menezes 2006. Dealing with gene expression missing data. *IEE Proceedings-Systems Biology* **153**, 105–119.

Brereton, R. 2003. *Chemometrics: data analysis for the laboratory and chemical plant*. Wiley.

Brereton, R. 2007. *Applied Chemometrics for Scientists*. Wiley.

Broomhead, D. and D. Lowe 1988. Feed-forward neural networks and topographic mappings for exploratory data analysis. *Complex Systems* **2**, 321–355.

Bullen, R., D. Cornford, and I. Nabney 2003. Outlier detection in scatterometer data: neural network approaches. *Neural Networks* **16** (3-4), 419–426.

Chapra, S. C. 2004. *Applied Numerical Methods with MATLAB for Engineers and Scientists*. McGraw-Hill Professional.

Chatfield, C. and A. Collins 1980. *Introduction to Multivariate Analysis*. Chapman and Hall.

Christensen, J., A. Hansen, G. Tomasi, J. Mortensen, and O. Andersen 2004. Integrated methodology for forensic oil spill identification. *Environ. Sci. Technol* **38** (10), 2912–2918.

Christoffersson, A. 1970. *The one component model with incomplete data*. Ph.D. thesis, Uppsala University.

Clark, D. and C. Thayer 2004. A primer on the exponential family of distributions. In *Casualty Actuarial Society Spring Forum*, pp. 117–148.

Cooke, M., P. Green, L. Josifovski, and A. Vizinh 2001. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Communication* **34**, 267–285.

Cormen, T. H., C. E. Leirserson, and R. R. Livest 1997. *Introduction to Algorithms*. MIT Press.

Cox, T. and M. Cox 1994. *Multidimensional Scaling. Number 59 in Monographs on Statistics and Applied Probability*. Chapman and Hall.

Cruz-Barbosa, R. and A. Vellido 2008. Geodesic Generative Topographic Mapping. In *Proceedings of the 11th Ibero-American conference on AI: Advances in Artificial Intelligence*, pp. 113–122. Springer.

Curiale, J. 1994. Correlation of oils and source rocks-A conceptual and historical perspective. *Memoirs-American Association Of Petroleum Geologists* **1**, 251–251.

Demaison, G. and R. Murris 1984. *Petroleum geochemistry and basin evaluation.* American Association of Petroleum Geologists, Tulsa, OK.

Dempster, A., N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* **39**, 1–38.

Dickson, B. and A. Giblin 2007. An evaluation of methods for imputation of missing trace element data in groundwaters. *Geochemistry: Exploration, Environment, Analysis* **7** (2), 173–178.

Dray, S., N. Pettorelli, and D. Chessel 2003. Multivariate analysis of incomplete mapped data. *Transactions in GIS* **7** (3), 411–422.

Edsall, R. 2003. The parallel coordinate plot in action: design and use for geographic visualization. *Computational Statistics and Data Analysis* **43** (4), 605–619.

Farnham, I., A. Singh, K. Stetzenbach, and K. Johannesson 2002. Treatment of nondetects in multivariate analysis of groundwater geochemistry data. *Chemometrics and Intelligent Laboratory Systems* **60** (1-2), 265–281.

Farnham, I., K. Stetzenbach, A. Singh, and K. Johannesson 2000. Deciphering groundwater flow systems in Oasis Valley, Nevada, using trace element chemistry, multivariate statistics, and geographical information system. *Mathematical Geology* **32** (8), 943–968.

Fernandes, M. B., V. O. Elias, J. N. Cardoso, and M. S. Carvalho 1999. Sources and fate of n-alkanols and sterols in sediments of the Amazon shelf. *Organic Geochemistry* **30** (9), 1075–1087.

Ford, B. 1983. An overview of hot-deck procedure. *Theory and Bibliographies: Incomplete Data in Sample Surveys* **2**, 85–207.

García, D., A. Vellido, and Ê. Nebot 2007. Identification of churn routes in the Brazilian telecommunications market. *Proceedings of the 15th European Symposiun on Artificial Neural Networks,ESANN 2007* **15**, 585–590.

Ghahramani, Z. and M. I. Jordan 1994. Learning from Incomplete Data. Technical Report AIM-1509, MIT AI Lab.

Gianniotis, N. and P. Tino 2008. Visualization of Tree-Structured Data Through Generative Topographic Mapping. *IEEE Transactions on Neural Networks* **19** (8), 1468–1493.

Girolami, M. 2001. The topographic organization and visualization of binary data using multivariate-Bernoulli latent variable models. *IEEE Transactions on Neural Networks* **12** (6), 1367–1374.

Gold, T. 1985. The Origin of Natural Gas and Petroleum, and the Prognosis for Future Supplies. *Annual Review of Energy* **10**, 53–77.

Gordon, E. S. and M. A. Goñi 2003. Sources and distribution of terrigenous organic matter delivered by the Atchafalaya River to sediments in the northern Gulf of Mexico. *Geochimica et Cosmochimica Acta* **67** (13), 2359–2375.

Grimmenstein, I., K. Quast, W. Urfer, F. Statistik, B. GmbH, and C. KG 2004. Analyzing Microarray Data with the Generative Topographic Mapping Approach. In *Classification-the Ubiquitous Challenge: Proceedings of the 28th Annual Conference of the Gesellschaft Für Klassifikation EV, University of Dortmund, March 9-11, 2004*, pp. 338. Springer-Verlag New York Inc.

Grunfeld, K. 2007. The separation of multi-element spatial patterns in till geochemistry of southeastern Sweden combining GIS, principal component analysis and high-dimensional visualization. *Geochemistry: Exploration, Environment, Analysis* **7** (4), 303–318.

Grung, B. and R. Manne 1998. Missing values in principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **42** (1-2), 125–139.

Haese, K. and G. J. Goodhill 2001. Auto-SOM: Recursive Parameter Estimation for Guidance of Self-Organizing Feature Maps. *Neural Computation* **13**, 595–619.

Hahsler, M., K. Hornik, and C. Buchta 2008. Getting Things in Order: An introduction to the R package seriation. *J. Statistical Software* **25**, 1–34.

Harmeling, S. 2007. Exploring model selection techniques for nonlinear dimensionality reduction. Technical Report EDI-INF-RR-0960, Edinburgh University, Scotland.

Hartigan, J. A. and M. A. Wong 1979. A K-Means Clustering Algorithm. *Applied Statistics* **28**, 100–108.

Horsfield, B. and J. Rullkotter 1994. Diagenesis, catagenesis, and metagenesis of organic matter. *L.B. Magoon and W.G. Dow (Editors), The Petroleum System - from Source to Trap. AAPG Memoir No. 60* **60**, 189–201.

Huang, W. and W. Meinschein 1979. Sterols as ecological indicators. *Geochimica et Cosmochimica Acta* **43** (5), 739–745.

IGI-Ltd. 2009. Devon. http://www.igiltd.com.

Inselberg, A., B. Dimsdale, I. Center, and C. Los Angeles 1990. Parallel coordinates: a tool for visualizing multi-dimensionalgeometry. In *Visualization, 1990. Visualization'90., Proceedings of the First IEEE Conference on*, pp. 361–378.

Isaacs, C. 2001. *The Monterey Formation: from rocks to molecules*, Chapter Statistical Evaluation of Interlaboratory Data from the Cooperative Monterey Organic Geochemistry Study, pp. 461–524. Columbia University Press.

Jolliffe, I. 1986. *Principal Component Analysis*. Springer Verlag.

Justwan, H., D. Dahl, and G. Isaksen 2006. Geochemical characterisation and genetic origin of oils and condensates in the South Viking Graben, Norway. *Marine and Petroleum Geology* **23** (2), 213–239.

Kettaneh, N., A. Berglund, and S. Wold 2005. PCA and PLS with very large data sets. *Computational Statistics and Data Analysis* **48** (1), 69–85.

Kim, H., G. Golub, and H. Park 2005. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics* **21** (2), 187–198.

Kohonen, T. 1995. *Self-Organizing Maps*. Springer Verlag.

Krzanowski, W. 1987. Cross-validation in principal component analysis. *Biometrics* **43** (3), 575–584.

Kucklick, J., S. Christopher, P. Becker, R. Pugh, B. Porter, M. Schantz, E. Mackey, S. Wise, and T. Rowles 2002. *Description and Results of the 2000 NIST/NOAA Interlaboratory Comparison Exercise Program for Organic Contaminants and Trace Elements in Marine Mammal Tissues* (6849 ed.). National Institute of Standards and Technology, NISTIR.

Kvalheim, O. 1987a. Latent-structure decompositions (projections) of multivariate data. *Chemometrics and Intelligent Laboratory Systems. Elsevier Science, Amsterdam* **2** (4), 283–290.

Kvalheim, O. 1987b. Oil-source correlation by the combined use of principal component modelling, analysis of variance and a coefficient of congruence. *Chemometrics and intelligent laboratory systems* **2** (1-3), 127–136.

Kvalheim, O., F. Brakstad, and Y. Liang 1994. Preprocessing of analytical profiles in the presence of homoscedatic or heteroscedastic noise. *Analytical chemistry (Washington, DC)* **66** (1), 43–51.

Kvalheim, O. and T. Karstang 1987. A general-purpose program for multivariate data analysis. *Chemometrics and Intelligent Laboratory Systems. Elsevier Science, Amsterdam* **2** (1-3), 235–237.

Kvalheim, O. and N. Telnaes 1986a. Visualizing information in multivariate data: Applications to petroleum geochemistry:: Part 1. Projection methods. *Analytica Chimica Acta* **191**, 87–96.

Kvalheim, O. and N. Telnaes 1986b. Visualizing information in multivariate data: Applications to petroleum geochemistry:: Part 2. Interpretation and correlation of north sea oils by using three different biomarker fractions. *Analytica Chimica Acta* **191**, 97–110.

Lakshminarayan, K., S. Harp, R. Goldman, T. Samad, *et al.* 1996. Imputation of missing data using machine learning techniques. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 140–145.

Latini, G. and G. Passerini 2004. *Handling Missing Data*. WIT Press, Southampton, UK.

Law, M., M. Figueiredo, and A. Jain 2004. Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26** (9), 1154–1166.

Lawrence, N. D. 2005. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research* **6**, 1783–1816.

Leen, G. and C. Fyfe 2005. Training an AI player to play pong using a GTM. In *IEEE 2005 Symposium on Computational Intelligence and Games CIG*, pp. 270–276.

Liao, G. and T. Shi 2004. Condition Monitoring of Gearbox Based on Generative Topographic Mapping. *Zhengdong Ceshi yu Zhenduan(Journal of Vibration, Measurement & Diagnosis)(China)* **24** (1), 11–14.

Liechty, J. C., M. W. Liechty, and P. Müller 2004. Bayesian correlation estimation. *Biometrika* **91**, 1–14.

Little, R. J. A. and D. B. Rubin 2002. *Statistical analysis with missing data*. New York, NY, USA: John Wiley & Sons, Inc.

Lockheart, M., P. van Bergen, and R. Evershed 2000. Chemotaxonomic classification of fossil leaves from the Miocene Clarkia lake deposit, Idaho, USA based on n-alkyl lipid distributions and principal component analyses. *Organic Geochemistry* **31** (11), 1223–1246.

Lowe, D. and M. Tipping 1996. Feed-forward neural networks and topographic mappings for exploratory data analysis. *Neural Computing and Applications* **4**, 84–95.

Lowe, W. 2001. What is the Dimensionality of Human Semantic Space? In *Connectionist Models of Learning, Development and Evolution: Proceedings of the Sixth Neural Computation and Psychology Workshop, Liege, Belgium, 16-18 September 2000*, pp. 303–312. Springer.

Magnus, J. R. and H. Neudecker 1999. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley and Sons.

Magoon, L. and G. Claypool 1985. *Alaska North Slope oil/rock correlation study*. AAPG. Studies in Geology, Tulsa Oklahoma USA.

Maniyar, D. and I. Nabney 2005. Guiding local regression using visualisation. *Lecture notes in computer science* **3635**, 98–109.

Maniyar, D. and I. Nabney 2006a. Data Visualization with Simultaneous Feature Selection. In *2006 IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, 2006. CIBCB'06*, pp. 1–8.

Maniyar, D. and I. Nabney 2006b. Visual data mining using principled projection algorithms and information visualization techniques. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 643–648. ACM New York, NY, USA.

Maniyar, D., I. Nabney, B. Williams, and A. Sewing 2006a. Data visualization during the early stages of drug discovery. *J. Chem. Inf. Model* **46** (4), 1806–1818.

Maniyar, D. M., I. T. Nabney, B. S. Williams, and A. Sewing 2006b. Data Visualization During the Early Stages of Drug Discovery). *Journal of Chemical Information and Modeling (JCIM* **46**, 1806–1818.

Mann, U., S. J. Duppenbecker, A. Langen, B. Ropertz, and D. H. Welte 1991. Pore network evolution of the Lower Toarcian Posidonia Shale during petroleum generation and expulsion - a multidisciplinary approach. *Zbl. Geol. Palaont.* **1**, 51–1071.

MathWorks 2009. Internet. http://www.mathworks.com/access/helpdesk/help/toolbox/stats/index.html?/access/helpdesk/ help/toolbox/stats/boxplot.html.

McGill, R., J. W. Tukey, and W. A. Larsen 1978. Variations of Box Plots. *The American Statistician*, **32**, 12–16.

Mead, R., Y. Xu, J. Chong, and R. Jaffé 2005. Sediment and soil organic matter source assessment as revealed by the molecular distribution and carbon isotopic composition of n-alkanes. *Organic Geochemistry* **36** (3), 363–370.

Møller, M. F. 1993. A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. *Neural Networks* **6**, 525–533.

Moeller, U. and D. Radke 2006. Performance of data resampling methods for robust class discovery based on clustering. *Intelligent Data Analysis* **10**, 139–162.

Morris, S., B. Asnake, and G. Yen 2003. Optimal dendrogram seriation using simulated annealing. *Information Visualization* **2** (2), 95–104.

Morrison, A., G. Ross, and M. Chalmers 2003. Fast multidimensional scaling through sampling, springs and interpolation. *Information Visualization* **2** (1), 68–77.

Nabney, I., Y. Sun, P. Tino, and A. Kaban 2005. Semisupervised learning of hierarchical latent trait models for data visualization. *IEEE Transactions on Knowledge and Data Engineering* **17** (3), 384–400.

Nabney, I. T. 2002. *Netlab: Algorithms for Pattern Recognition*. Springer Verlag.

Napitupulu, H., L. Ellis, and R. Mitterer 2000. Post-generative alteration effects on petroleum in the onshore Northwest Java Basin, Indonesia. *Organic Geochemistry* **31** (4), 295–315.

Nelson, P., P. Taylor, and J. MacGregor 1996. Missing data methods in PCA and PLS: Score calculations with incomplete observations. *Chemometrics and intelligent laboratory systems* **35** (1), 45–65.

Nguyen, D. and D. Rocke 2004. On partial least squares dimension reduction for microarray-based classification: a simulation study. *Computational Statistics and Data Analysis* **46** (3), 407–425.

Niggemann, J. and C. J. Schubert 2006. Fatty acid biogeochemistry of sediments from the Chilean coastal upwelling region: Sources and diagenetic changes. *Organic Geochemistry* **37** (5), 626–647.

Niskanen, M. and O. Silven 2003. Comparison of dimensionality reduction methods for wood surface inspection. In *Proceedings of the 6 th International Conference on Quality Control by Artificial Vision*, pp. 178–188.

Oba, S., M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii 2003. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics* **19** (16), 2088–2096.

Odden, W. and O. Kvalheim 2000. Application of multivariate modelling to detect hydrocarbon components for optimal discrimination between two source rock types. *Applied geochemistry* **15** (5), 611–627.

Ohm, S., D. Karlson, and T. Austin 2008. Geochemically driven exploration models in uplifted areas: Examples from the Norwegian Barents Sea. *merican Association of Petroleum Geologists Bulletin* **92** (9), 119–1223.

Olier, I. and A. Vellido 2005. Comparative assessment of the robustness of missing data imputation through Generative Topographic Mapping. *Lecture Notes in Computer Science* **3512**, 787–794.

Olier, I. and A. Vellido 2006. Capturing the dynamics of multivariate time series through visualization using Generative Topographic Mapping Through Time. In *IEEE International Conference on Engineering of Intelligent Systems*, pp. 1–6.

Olier, I. and A. Vellido 2008a. Advances in clustering and visualization of time series using GTM through time. *Neural Networks* **21** (7), 904–913.

Olier, I. and A. Vellido 2008b. Variational Bayesian Generative Topographic Mapping. *Journal of Mathematical Modelling and Algorithms* **7** (4), 371–387.

Olinsky, A., S. Chen, and L. Harlow 2003. The comparative efficacy of imputation methods for missing data in structural equation modeling. *European Journal of Operational Research* **16**, 53–79.

Orphanidou, C., I. Moroz, and S. Roberts 2003. Voice morphing using the generative topographic mapping. *Proceedings of CCCT 03* **1**, 222–225.

Palarea-Albaladejo, J. and J. Martín-Fernández 2008. A modified EM alr-algorithm for replacing rounded zeros in compositional data sets. *Computers and Geosciences* **34** (8), 902–917.

Pasadakis, N., M. Obermajer, and K. Osadetz 2004. Definition and characterization of petroleum compositional families in Williston Basin, North America using principal component analysis. *Organic Geochemistry* **35** (4), 453–468.

Pena, M. and C. Fyfe 2006. The topographic neural gas. *Lecture Notes in Computer Science* **4224**, 241–248.

Peters, K. and J. Moldowan 1993. *The biomarker guide: interpreting molecular fossils in petroleum and ancient sediments*. Prentice Hall Englewood Cliffs, NJ.

Peters, K., R. L. Scott, J. Zumberge, Z. Valin, C. Scotese, and D. Gautier 2007. American Association of Petroleum Geologists Bulletin. *Marine and Petroleum Geology* **91** (6), 877–913.

Peters, K., C. Walters, and J. Moldowan 2005. *The biomarker guide*. Cambridge University Press, Cambridge.

Petrie, W. 1899. Sequences in prehistoric remains. *Journal of the Anthropological Institute of Great Britain and Ireland* **29** (3/4), 295–301.

Priam, R., M. Nadif, and G. Govaert 2008. The Block Generative Topographic Mapping. *Lecture Notes in Computer Science* **5064**, 13–23.

Raghunathan, J. E., J. M. Lepkowski, J. V. Hoewyk, and P. Solenberger 2001. A Multivariate Technique for Multiply Imputing Missing Values Using a Sequence of Regression Models. *Survey Methodology* **27**, 85–95.

Rannar, S., P. Geladi, F. Lindgren, and S. Wold 1995. A PLS kernel algorithm for data sets with many variables and few objects. Part II: Cross-validation, missing data and examples. *Journal of Chemometrics* **9** (6), 459–470.

Rasmussen, C. E. and C. K. I. Williams 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Reimann, C., P. Filzmoser, and R. Garrett 2002. Factor analysis applied to regional geochemical data: problems and possibilities. *Applied Geochemistry* **17** (3), 185–206.

Roberts, W. and R. Cordell 1980. Problems of petroleum migratio. *AAPG. Studies in Geology* **25** (10), 455–472.

Rock, N. M. S. 1988. *Numerical Geology: a source guide, bibliography and glossary to geological uses of computers and statistics*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Roweis, S. and L. Saul 2000. Locally Linear Embedding. *Science* **290**, 2323–2326.

Rubin, D. B. 1976. Inference and missing data. *Biometrika* **63**, 581–592.

Sanguinetti, G. and N. D. Lawrence 2006. Missing Data in Kernel PCA. *Lecture Notes in Computer Science* **4212**, 751–758.

Saul, L. and S. Roweis 2003. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research* **4**, 119–155.

Schafer, J. L. 1997. *Analysis of incomplete multivariate data*. Chapman and Hall.

Scheffer, J. 2002. Dealing with Missing Data. Research Letter 3, 153-160, Massey University.

Schoelkopf, B., A. Smola, and K.-R. Mueller 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* **10**, 1299–1318.

Schroeder, M., D. Cornford, P. Farrimond, and C. Cornford 2008. Addressing missing data in geochemistry: A non-linear approach. *Organic Geochemistry* **39**, 1162–1169.

Seifert, W. and J. Moldowan 1981. Paleoreconstruction by biological markers. *Geochimica et Cosmochimica Acta* **45** (6), 783–794.

Shakhnarovish, Darrell, and Indyk 2005. *Nearest-Neighbor Methods in Learning and Vision*. MIT Press.

Shawe-Taylor, J. and N. Cristianini 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Song, Q. and M. Shepperd 2007. A new imputation method for small software project data sets. *The Journal of Systems and Software* **80**, 51–62.

Sun, Y. 2002. *Non-linear Hierarchical Visualisation*. Ph.D. thesis, Aston University.

Sun, Y., T. Butler, A. Shafarenko, R. Adams, M. Loomes, and N. Davey 2007. Word segmentation of handwritten text using supervised classification techniques. *Applied Soft Computing Journal* **7** (1), 71–88.

Svensén, C. and C. Williams 1997. Magnification Factors for the GTM Algorithm. In *Proc. IEE Fifth Int'l Conf. Artificial Neural Networks*, pp. 64–69.

Telnaes, N. and B. Dahl 1986. Oil-oil correlation using multivariate techniques. *Marine and Petroleum Geology* **10** (1-3), 425–432.

Tenenbaum, J., V. de Silva, and J. Langford 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* **290**, 2319–2323.

Thió-Henestrosa, S. and J. Martın-Fernández 2003. Compositional Data Analysis Workshop CoDaWork'03. In *Proceedings. Universitat de Girona*, pp. 84–8458.

Tino, P. and I. Nabney 2002. Hierarchical GTM: constructing localized nonlinear projection manifolds in a principled way. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (5), 639–656.

Tino, P., I. Nabney, and Y. Sun 2001. Using directional curvatures to visualize folding patterns of the GTM projection manifolds. *Lecture notes in computer science* **2130**, 421–428.

Tipping, M. and D. Lowe 1997. Shadow targets: a novel algorithm for topographic projections by radial basis functions. In *Artificial Neural Networks, Fifth International Conference on (Conf. Publ. No. 440)*, pp. 7–12.

Tipping, M. E. and C. M. Bishop 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society* **B,6(3)**, 611–622.

Tresp, V., S. Ahmad, and R. Neuneier 1994. Training Neural Networks with Deficient Data. In J. D. Cowan, G. Tesauro, and J. Alspector (Eds.), *Advances in Neural Information Processing Systems*, Volume 6, pp. 128–135. Morgan Kaufmann Publishers, Inc.

Troyanskaya, O., M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman 2001. Missing value estimation methods for DNA microarrays. *Bioinformatics* **17**, 520–525.

Vellido, A. 2006a. Assessment of an Unsupervised Feature Selection Method for Generative Topographic Mapping. *Lecture Notes in Computer Science* **4132**, 361.

Vellido, A. 2006b. Missing data imputation through GTM as a mixture of T-distributions. *Neural Networks* **19** (10), 1624–1635.

Vellido, A. and P. Lisboa 2006. Handling outliers in brain tumour MRS data analysis through robust topographic mapping. *Computers in Biology and Medicine* **36** (10), 1049–1063.

Vellido, A., E. Marti, J. Comas, I. Rodriguez-Roda, and F. Sabater 2007. Exploring the ecological status of human altered streams through Generative Topographic Mapping. *Environmental Modelling and Software* **22** (7), 1053–1065.

Verbeek, J., N. Vlassis, and B. Krose 2002. Locally linear generative topographic mapping. In *Proc. of 12th Belgian-Dutch Conf. on Machine Learning*.

Vicente, D., A. Vellido, E. Marti, J. Comas, and I. Rodriguez-Roda 2004. Exploration of the ecological status of Mediterranean rivers: clustering, visualizing and reconstructing streams data using Generative Topographic Mapping. *WIT Transactions on Information and Communication Technologies* **33**, 121–130.

Walker, S., R. Dickhut, C. Chisholm-Brause, S. Sylva, and C. Reddy 2005. Molecular and isotopic identification of PAH sources in a highly industrialized urban estuary. *Organic Geochemistry* **36** (4), 619–632.

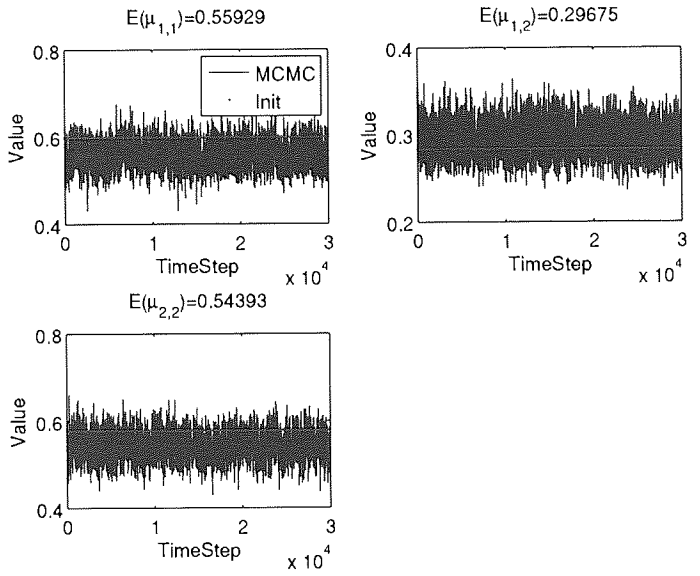
Wold, S. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* **2** (1), 37–52.

Yu, C. H. 2003. Resampling methods: concepts, applications, and justification. *Practical Assessment, Research and Evaluation* **8** (19), 1–23.

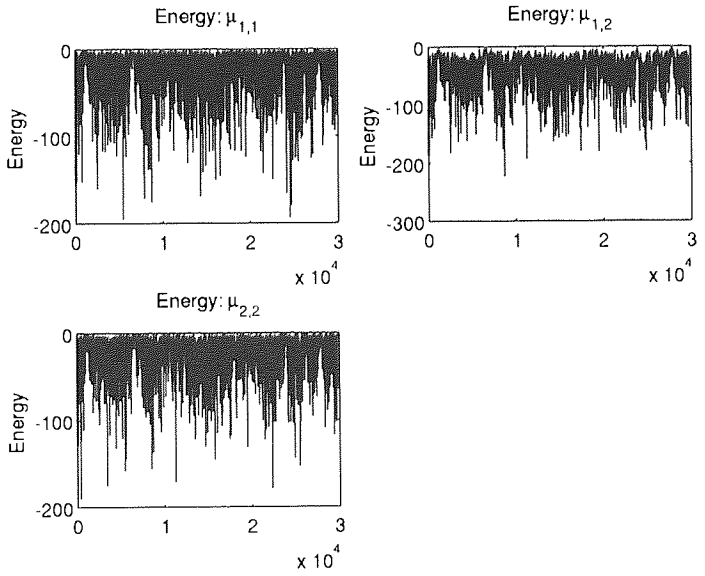
Zumberge, J. 1987. Prediction of source rock characteristics based on terpane biomarkers in crude oils: a multivariate statistical approach. *Geochimica et Cosmochimica Acta* **51**, 1625–1637.

A Additional Graphics

A.1 Chapter 5



(a)



(b)

Figure A.1: Plots used to test for the convergence of the QBCE algorithm. a) The distribution of the mean. b) The plots show the energy of the distribution of the mean and one can see that the MCMC algorithm converged for these parameters.

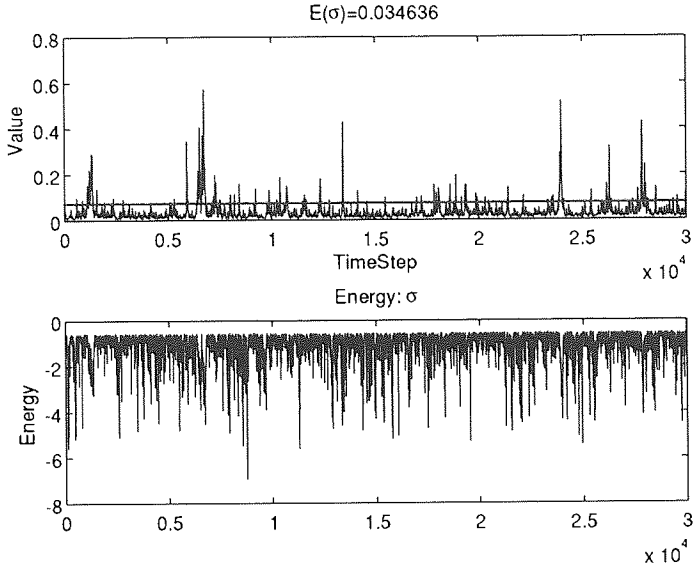
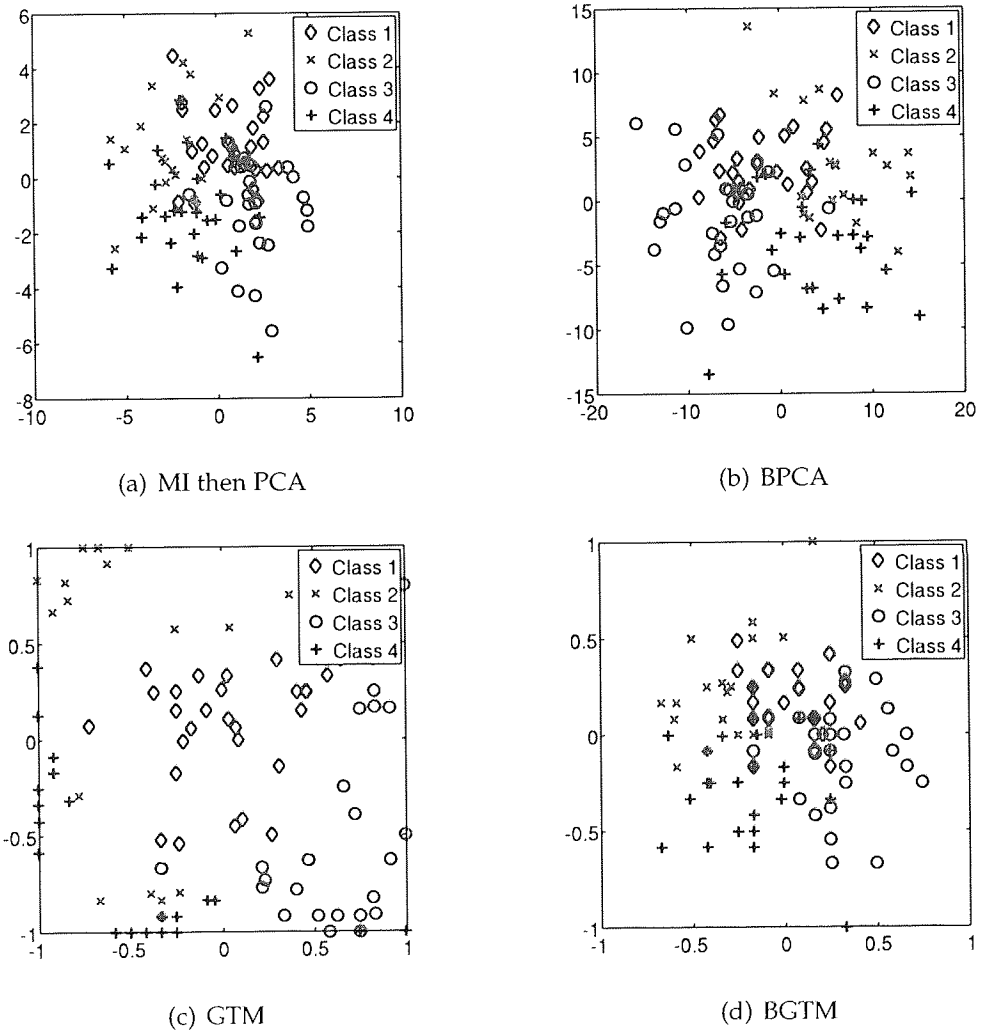


Figure A.2: Plots used to test for the convergence of the QBCE algorithm. The plots show the distribution and the energy of sigma and one can see the MCMC algorithm converged for these parameters.

A.2 Chapter 6



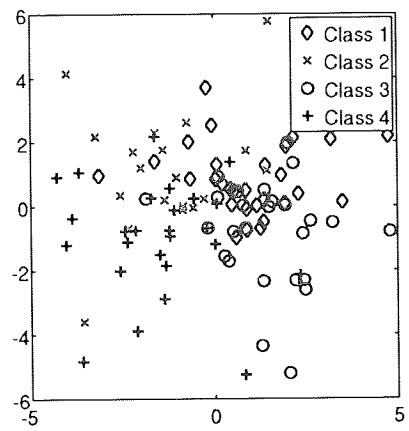
(a) MI then PCA

(b) BPCA

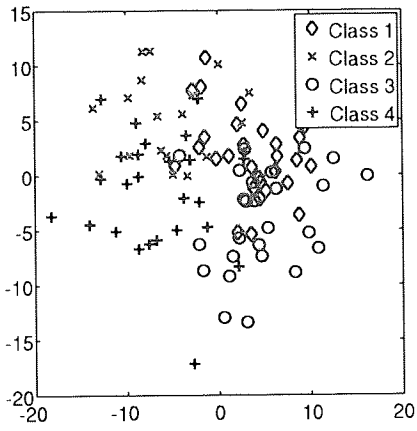
(c) GTM

(d) BGTM

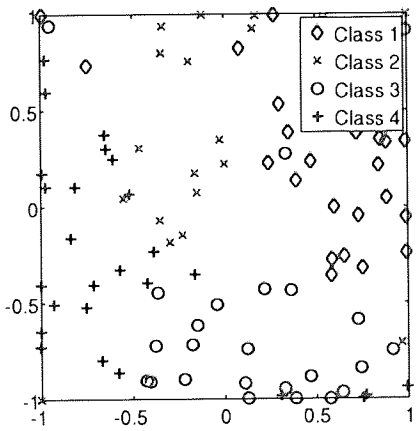
Figure A.3: Projection of the 20D toy data set data, with $p = 0.2$ as proportion of missing values. They show that only in the case of B-GTM (c) it is possible to distinguish class boundaries in the projection.



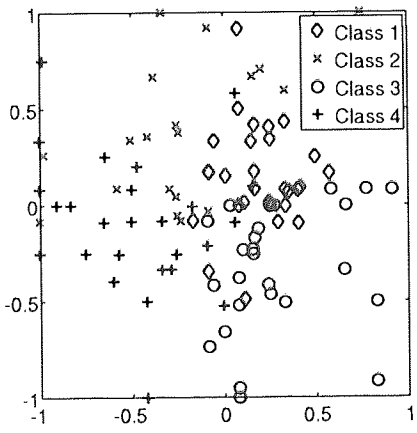
(a) MI then PCA



(b) BPCA

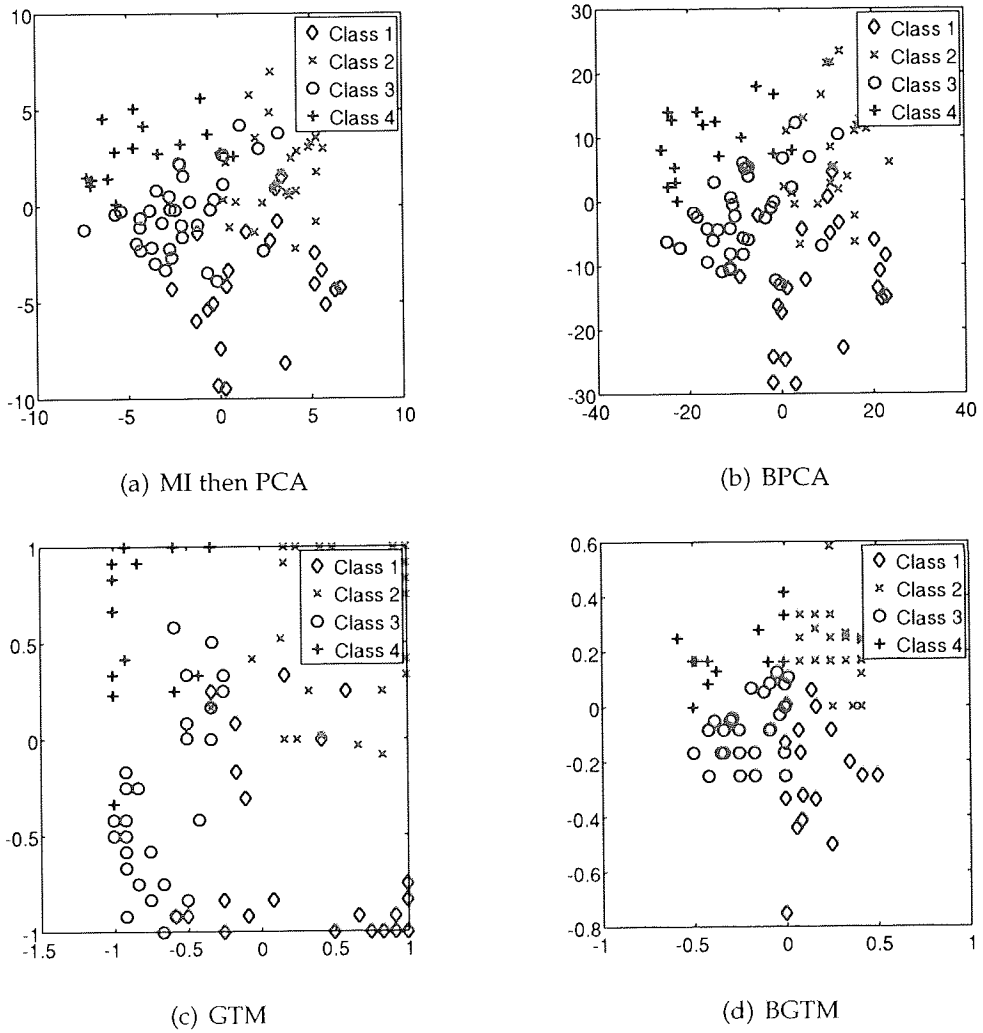


(c) GTM



(d) BGTM

Figure A.4: Projection of the 20D toy data set data, with $p = 0.6$ as proportion of missing values. They show that in no case it is possible to distinguish class boundaries in the projection.



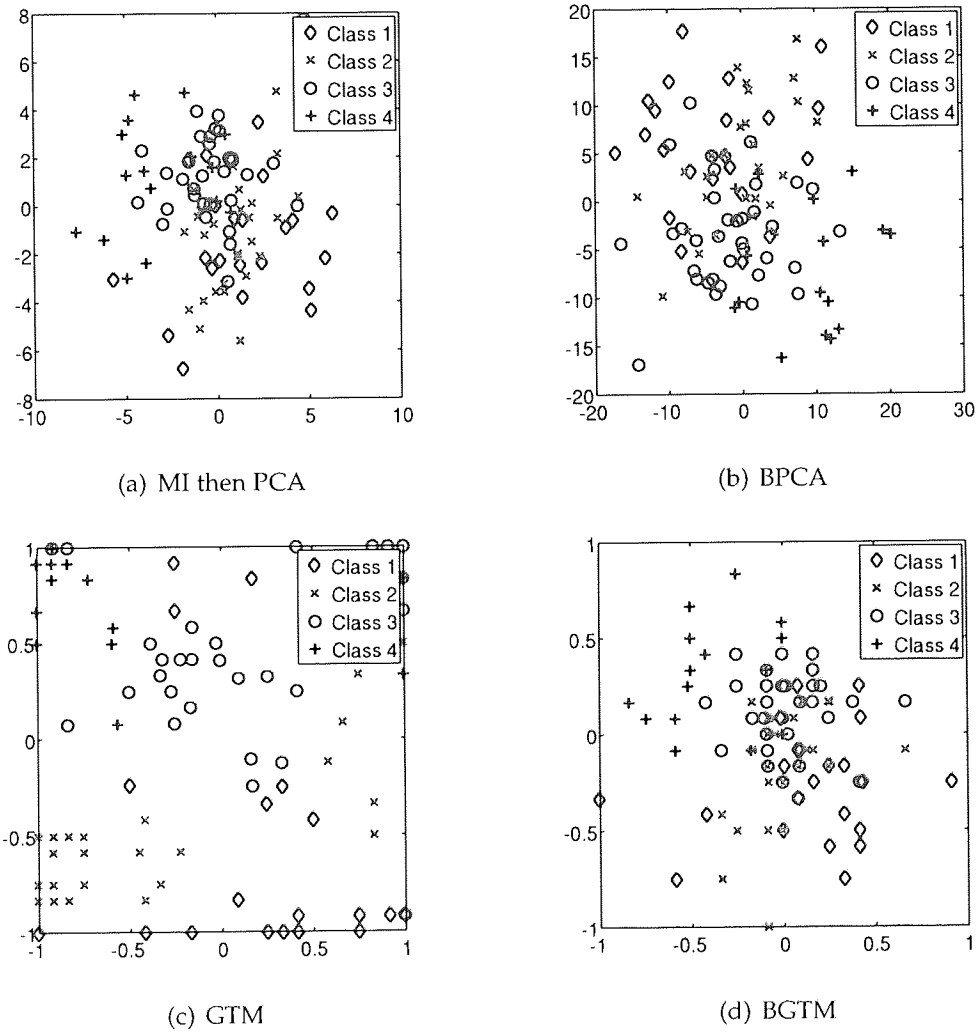
(a) MI then PCA

(b) BPCA

(c) GTM

(d) BGTM

Figure A.5: Projection of the 60D toy data set data, with $p = 0.2$ as proportion of missing values. They show that the class boundaries a very smeared in all 4 cases but that distinctions still can be made.



(a) MI then PCA

(b) BPCA

(c) GTM

(d) BGTM

Figure A.6: Projection of the 60D toy data set data, with $p = 0.6$ as proportion of missing values. They show that in no case it is possible to distinguish class boundaries in the projection.

B Data Modelling

B.1 PCA

Principal component analysis (Jolliffe, 1986) is the most widely used method for dimension reduction, and thus visualisation. The algorithm obtains a direct orthogonal projections of a point in a D-dimensional space onto a hyperplane in L-dimensional space with $L \leq D$. PCA takes a data set $\mathbf{Y} = \mathbf{y}_1, ..., \mathbf{y}_N$ and finds a new orthonormal basis $\mathbf{u}_1, ..., \mathbf{u}_D$ with its axes ordered in such a way that the first axis explains the largest variance in $\mathbf{Y}$. The second axis is orthogonal to the first and accounts for a maximum of the remaining variance in the data and the subsequent axes follow this schema.

Given that the set of observations are centred, $\sum_n^N \mathbf{y}_n = 0$, PCA finds the principal components by diagonalising the covariance matrix,

$$\mathbf{C} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{y}_n \mathbf{y}_n^T ,$$

and then finding its eigen-structure

$$\mathbf{CU} = \mathbf{U\Lambda} .$$

Here $\mathbf{U}$ is a $D \times D$ matrix which has the unit length eigenvectors, $\mathbf{u}_1, ..., \mathbf{u}_D$, as its columns and $\mathbf{\Lambda}$ is a diagonal matrix with the corresponding eigenvalues, $\lambda_1 > \lambda_2 > \ldots > \lambda_D$, along the diagonal. The principal components are the eigenvectors and the data can be projected onto these. The eigenvectors are also

termed loadings because the single elements can be seen as weight (i.e. loading) for every dimension when projecting the data onto the eigenvector. Thus the loadings are used to interpret how much a certain variable contributes to a principal component. The principal components can be described as the directions along which the data set has the biggest variance. The eigenvalues are directly related to the corresponding variances and therefore the bigger the eigenvalue the more information is stored in the eigenvector. In general one discards principal components if the eigenvalues fall beneath a certain variance threshold. However other methods like cross-validation (Krzanowski, 1987) are also employed to determine the choice of dimensionality. Calculating the projections $(\mathbf{pc}_1, \ldots, \mathbf{pc}_D)$ onto the eigenvectors is straight forward :

$$\begin{aligned} \mathbf{pc}_1 &= \mathbf{Y}\mathbf{u}_1^T \\ \mathbf{pc}_2 &= \mathbf{Y}\mathbf{u}_2^T \\ &\vdots \\ \mathbf{pc}_D &= \mathbf{Y}\mathbf{u}_D^T \end{aligned}$$

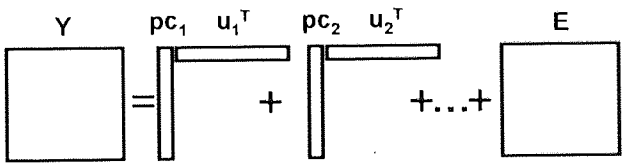


Figure B.1: Deconstruction of data space into subspaces (projections times loadings $pc_1 \times u_1$) and in the case of pruning (i.e. the omitting of negligible eigenvectors) with an error matrix **E** for the residuals.

PCA may be used to project higher-dimensional data onto a two-dimensional hyperplane to visualise it on a screen. Another way to use PCA is to cut down or prune the dimensionality to three or fewer dimensions as a pre-processing step for other analyses. In the case of pruning one will not be able to recoup the original positions of the data points in the data space and thus needs an error matrix **E** to describe the residuals. A more graphical way to envisage this deconstruction is illustrated in Figure B.1. The reasoning behind the pruning of variables is that the eigenvectors with the highest eigenvalues preserve most of the original variance. Thus the remaining eigenvectors do not contribute any real information about the data set and can be ignored. However this will only be true for highly linear data structures.

Commonly the major principal components, attributing for most of the variance, are used for the projection. A simple example of how this works can be seen

in Figure B.2 where the direction and the resulting hyperplane for the principal components are demonstrated on two-dimensional and the S-shaped data.

The fact that PCA defines a linear transformation makes it fast to compute but it is also the main drawback since non-linear structures in the data cannot always be captured. This is demonstrated in Figure B.4 where PCA manages to capture the general structure of the S-data set but fails to capture the structure of the Swiss-roll. In the given example the classes were chosen manually along the manifold of the Swill roll to highlight the difference in the model fit.

B.2 Probabilistic PCA

The probabilistic version of PCA (Tipping and Bishop, 1999) extends conventional PCA to a probabilistic framework while not changing the mapping. The maximum likelihood solution of PPCA has been shown to be the same as the one obtained through conventional PCA. In this section the model serves as a building block for other algorithms because it can easily be extended to Kernel PCA using the kernel trick or to the Gaussian Process Latent Variable Model using an appropriate covariance function. Assuming a D-dimensional data set $\mathbf{Y} = [\mathbf{y}_1, .., \mathbf{y}_N]$ and associated L-dimensional latent variables $\mathbf{x}_n$ the relationship between them may be expressed through a linear model with additive Gaussian noise,

$$\mathbf{y}_n = \mathbf{W}\mathbf{x}_n + \eta_n,$$

with the projection matrix $\mathbf{W} \in \Re^{D \times q}$. The noise values η_n are taken to be independent samples from a spherical Gaussian distribution with mean zero and covariance $\beta^{-1}\mathbf{I}$,

$$p(\eta_n) = \mathcal{N}(\eta_n | \mathbf{0}, \beta^{-1}\mathbf{I}).$$

Therefore the likelihood of a data point can be expressed as

$$p(\mathbf{y}_n | \mathbf{x}_n, \mathbf{W}, \beta) = \mathcal{N}(\mathbf{y}_n | \mathbf{W}\mathbf{x}_n, \beta^{-1}\mathbf{I}).$$

Integrating over the latent variables gives rise to the marginal likelihood

$$p(\mathbf{y}_n | \mathbf{W}, \beta) = \int \mathcal{N}(\mathbf{y}_n | \mathbf{W}\mathbf{x}_n, \beta^{-1}\mathbf{I}) p(\mathbf{x}_n) d\mathbf{x}_n. \tag{B.1}$$

Specifying the prior distribution over $\mathbf{x}_n$ to be a unit covariance, zero mean Gaussian distribution,

$$p(\mathbf{x}_n) = \mathcal{N}(\mathbf{x}_n | \mathbf{0}, \mathbf{I}), \tag{B.2}$$

gives rise to an analytic solution for the marginal likelihood for each data point,

$$p(\mathbf{y}_n | \mathbf{W}, \beta) = \mathcal{N}(\mathbf{y}_n | 0, \mathbf{W}\mathbf{W}^T + \beta^{-1}\mathbf{I}). \tag{B.3}$$

Taking advantage of the independence of the errors of the data points in the noise model, the full likelihood is given by

$$p(\mathbf{Y}|\mathbf{W},\beta) = \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{W},\beta^{-1}). \tag{B.4}$$

It can be shown that the maximum likelihood solution for the parameters $\mathbf{W}$ is achieved when the matrix $\mathbf{W}$ spans the principal sub-space of the data (Tipping and Bishop, 1999). Therefore the matrix $\mathbf{W}=(\mathbf{u}_1,...,\mathbf{u}_D)$ and thus projects the visualisation space points onto the principal components of the data space.

In the next section an alternative approach will be introduced where one marginalises over the parameters $\mathbf{W}$ and optimises with respect to the projected data points $\mathbf{X}$.

B.3 Kernel PCA

Instead of placing a prior over $\mathbf{X}$ in (B.1) we can place a simple prior , such as a spherical Gaussian, distribution over the weights $\mathbf{W}$:

$$p(\mathbf{W}) = \prod_{i=1}^{D} p(\mathbf{w}_i|\mathbf{0},\mathbf{I}) \,,$$

where $\mathbf{w}_i$ is the ith row of the matrix $\mathbf{W}$. Using this, the marginalisation of $\mathbf{W}$ is straightforward and the resulting marginal likelihood takes the form

$$p(\mathbf{Y}|\mathbf{X},\beta^{-1}) = \prod_{i=1}^{D} p(\mathbf{y}_{:,i}|\mathbf{X},\beta^{-1}) \,, \tag{B.5}$$

where $\mathbf{y}_{:,i}$ represents the dth column of $\mathbf{Y}$ and the likelihood for a single column is

$$p(\mathbf{y}_{:,d}|\mathbf{X},\beta^{-1}) = \mathcal{N}(\mathbf{y}_{:,d}|\mathbf{X}\mathbf{X}^T,\beta^{-1}\mathbf{I}) \,.$$

The negative log-likelihood is used as the objective function and minimised with respect to the latent variables

$$L = -\frac{DN}{2}\ln 2\pi - \frac{D}{2}\ln|\mathbf{K}| - \frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) \,, \tag{B.6}$$

where

$$\mathbf{K} = \mathbf{X}\mathbf{X}^T + \beta^{-1}\mathbf{I}.$$

The gradients of (B.6) with respect to X may be computed (Magnus and Neudecker, 1999) as,

$$\frac{\delta L}{\delta \mathbf{X}} = \mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1}\mathbf{X} - D\mathbf{K}^{-1}\mathbf{X}.$$

Setting the gradient to zero, the fixed point is given by the well known eigendecompositon for $\mathbf{Y}\mathbf{Y}^T$ which also solves the standard PCA problem:

$$\mathbf{X} = \mathbf{U}\mathbf{L}\mathbf{V}^T,$$

with $\mathbf{V}$ being an arbitrary rotation matrix, $\mathbf{U} \in \Re^{N \times q}$ is a matrix whose columns are the first q eigenvectors of $\mathbf{YY}^T$ and $\mathbf{L} \in \Re^{q \times q}$ is a diagonal matrix with diagonal elements $l_j = (\lambda_j - \frac{1}{\beta})^{\frac{1}{2}}$ and λ_j, the jth eigenvalue associated with the jth eigenvector of $\mathbf{YY}^T$. Solving this results in the well known PCA algorithm. Replacing $\mathbf{YY}^T$ by a kernel (i.e. weighting function) (Shawe-Taylor and Cristianini, 2004) results in the Kernel PCA algorithm (Schoelkopf *et al.*, 1998). In machine learning this is known as the kernel trick, based on Mercer's theorem (Ash, 1990). It is a method for using a linear algorithm to solve a non-linear problem by mapping the original non-linear observations into a higher-dimensional space, where the linear algorithm is subsequently used. However when utilising the kernel trick the mapping is never done but instead the kernel is expressed as a dot product which can be calculated directly.

The computational costs are the same as with PCA. To obtain the projection one at has to map the data into the higher-dimensional space and then multiply the data times the eigenvectors. Thus through KPCA a mapping function is obtained which can be used for test data as well. The downside of transforming the data into a higher dimensional feature space is that Kernel PCA loses the interpretability of the loadings for the principal components. Additionally the inverse mapping for most kernels is not know and thus one loses the possibility of projecting the points in the visualisation space back into the data space. A different approach to extend PPCA into a non linear-model will be elaborated in the next section.

B.4 Gaussian Process based data visualisation

B.4.1 Gaussian Processes

Gaussian processes (Rasmussen and Williams, 2006) are well known in the machine learning community and mainly used in regression problems. A Gaussian process can be viewed as a probabilistic model which specifies a distribution over a function space. We define a non-linear regression model with Gaussian noise

$$f(\mathbf{x}) = \mathbf{W}\phi(\mathbf{x}) ,$$
$$\mathbf{y} = f(\mathbf{x}) + \eta ,$$

where $\mathbf{x}$ is the input vector, $\mathbf{W}$ is a vector of weights, ϕ is a fixed function to a feature space, $f(\mathbf{x})$ is the function value and $\mathbf{y}$ is the observed target value. The noise η is assumed to be i.i.d. with zero mean and variance β^{-1}; therefore $\eta \sim \mathcal{N}(0, \beta^{-1})$. Assuming a prior for the weights $\mathbf{W}$ with zero mean Gaussian and covariance matrix $\sum$, $\mathbf{W} \sim \mathcal{N}(0, \sum)$ and using Bayes' rule we are able to express the posterior distribution for the weights given some known data $\mathbf{X}$ and target values $\mathbf{y}$:

$$p(\mathbf{W}|\mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{W}, \mathbf{X})p(\mathbf{W})}{p(\mathbf{y}|\mathbf{X})} ,$$
$$p(\mathbf{W}|\mathbf{y}, \mathbf{X}) \sim \mathcal{N}(\beta \mathbf{A}^{-1}\mathbf{X}\mathbf{y}, \mathbf{A}^{-1}) ,$$

with $\phi(\mathbf{x}) = \Phi$ and $A = \beta^{-1}\Phi\Phi^T + \sum^{-1}$. The mean of the distribution $p(\mathbf{W}|\mathbf{X}, \mathbf{y})$ is also its mode and is called the *maximum a posterior* (MAP) estimate of $\mathbf{W}$. The MAP estimate for $\mathbf{W}$ can now be used to make a prediction $f_* = f(\mathbf{x}_*)$ for a new value $\mathbf{x}_*$. This can be done in two ways either by using Bayes to integrate out the $\mathbf{W}$ as will be shown or by doing a **plug in** and just using the MAP as best estimate for $\mathbf{W}$ to calculate f_* directly. Using the MAP to integrate over $\mathbf{W}$ and to calculate f_* for a given point $\mathbf{x}_*$ one has:

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(f_*|\mathbf{x}_*, \mathbf{W}) p(\mathbf{W}|\mathbf{X}, \mathbf{y}) d\mathbf{W} ,$$

where we obtain in the linear case

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mathbf{x}_*(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{y}, \mathbf{x}_*^T(\beta^{-1}\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{x}_*) .$$

Here the predictive variance is quadratic and thus grows with the magnitude of the test input, as expected from a linear model. A simple way of extending the model to the non-linear case and overcome the limited expressiveness of the linear model is to first project the data into a higher dimensional space. This can be done by defining our mapping function Φ as a set of radial basis functions. This results in an alternative expression for the predictive distribution

$$p(f_*|\mathbf{x}_*, \mathbf{X}, y) = \mathcal{N}(\beta^{-1}\phi(\mathbf{x}_*)\mathbf{A}^{-1}\Phi y, \phi(\mathbf{x}_*^T)\mathbf{A}^{-1}\phi(\mathbf{x}_*)) ,$$

which can be rewritten (Rasmussen and Williams, 2006) in terms of kernel products

$$\begin{aligned} p(f_*|\mathbf{x}_*, \mathbf{X}, y) \quad &= \mathcal{N}(K(\mathbf{x}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I}]^{-1}y, \\ &K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \beta^{-1}\mathbf{I}]^{-1}K(\mathbf{X}, \mathbf{x}_*) , \end{aligned}$$

with $K(a, b) = (\phi(a)\phi(b))$. The function $K(a, b)$ is called *covariance function or kernel*. Possible choices for the kernel are the Gaussian kernel, which leads to smooth functions that become zero in the regions where there is no data

$$K_{gau}(\mathbf{x}_i, \mathbf{x}_j) = \theta_{rbf} \times \exp((\frac{\gamma}{2}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T) ,$$

and the MLP kernel, which can be seen as an multi-layer perceptron with an infinite number of hidden units.

$$K_{mlp}(\mathbf{x}_i, \mathbf{x}_j) = \theta_{mlp} \times \sin^{-1}\left(\frac{w\mathbf{x}_i^T\mathbf{x}_j + b}{\sqrt{(w\mathbf{x}_i^T\mathbf{x}_j + b + 1)(w\mathbf{x}_i^T\mathbf{x}_j + b + 1)}}\right) .$$

The MLP kernel also leads to smooth functions but differs from the Gaussian kernel in the case of regions where there is no data, since the function values do not become zero but tend to converge to a low value.

B.4.2 Gaussian Process Latent Variable Model (GPLVM)

To extend PPCA to GPLVM (Lawrence, 2005) one considers a linear Gaussian process prior over the space of mapping functions. Assuming a Gaussian noise variance $\beta^{-1}\mathbf{I}$ the covariance function, or kernel, is given by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j + \beta^{-1}\delta_{ij} ,$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are vectors in the visualisation space and and δ_{ij} is the Kronecker delta. In matrix form the kernel is written as

$$\mathbf{K} = \mathbf{X}\mathbf{X}^T + \beta^{-1}\mathbf{I} ,$$

which is the same as in (B.3) and which is associated with the covariance of the factors of the marginal likelihood in the case of probabilistic PCA in (B.1). Instead of using a kernel in the data space to substitute $\mathbf{Y}\mathbf{Y}^T$, as with KPCA, one can now introduce non-linearity to the model using a non-linear covariance function in the visualisation space instead. A possible choice would be the Gaussian kernel with

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_{rbf} \exp \frac{(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T}{2\sigma^2} ,$$

with the hyperparameter $\boldsymbol{\theta} = (\sigma, \theta_{rbf})$. Similarly to (B.5) the GPLVM likelihood is then given by

$$P(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) = \prod_{l=1}^{D} N(\mathbf{Y}_l|0, \mathbf{K}) , \tag{B.7}$$

which can be seen as a series of D Gaussian processes trying to predict the data points on each of the D dimensions respectively (columns), given the kernel based on the visualisation space. Thus the negative log-likelihood for the model is

$$-\ln P(\mathbf{Y}|\mathbf{X}, \boldsymbol{\theta}) = -\frac{DN}{2}\ln 2\pi - \frac{D}{2}\ln|\mathbf{K}| - \frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) ,$$

where $\mathrm{tr}(\mathbf{M})$ is defined as the trace of the matrix $\mathbf{M}$. The gradient for the likelihood can be found by first taking the gradient with respect to kernel $\mathbf{K}$,

$$\frac{\delta L}{\delta \mathbf{K}} = \mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\mathbf{K}^{-1} - D\mathbf{K}^{-1} .$$

and then combining it with $\frac{\delta \mathbf{K}}{\delta \mathbf{X}}$ through the chain rule. The term $\frac{\delta \mathbf{K}}{\delta \mathbf{X}}$ (e.g. the derivative of the kernel with respect to the data points) depends on the choice of kernel. Once $\frac{\delta \mathbf{K}}{\delta \mathbf{X}}$ is obtained the likelihood can be determined. Since the calculated objective function will be very complex, this will in general result in a highly non-linear likelihood landscape with multiple local minima. An optimum can then be found by using gradient based optimisation algorithms like scaled conjugate gradient (SCG) descent (Møller, 1993). It is important to note that one effectively optimises the positions of the points in the visualisation space. In the two-dimensional case the resulting density model may be described as the space of all possible hyperplanes running through the data points with the constraint

of satisfying the properties of the chosen covariance function. In reference to the explanation of the *"rubber sheet"*, used when explaining GTM, one can imagine the two-dimensional case as a distribution over all the possible *"rubber sheets"*. Another way of viewing the model with respect to (B.7) is as a constrained Gaussian Process regression on multiple dimensions. The Gaussian Processes are constrained to share the same parameters and kernels and one optimises the position of the points in the visualisation space to obtain the best prediction for the data points in every single dimension.

In general GPLVM suffers from the same problem as GTM and indeed all other probabilistic methods which need to be initialised. However since GPLVM does not optimise the mapping function but the points in the visualisation space directly one can use any available mapping for the visualisation space. This confers an advantage on GPLVM; that it can be initialised via any other visualisation method. This advantage can be seen in Figure B.5 where GPLVM is initialised by PCA and Isomap respectively and manages to correctly identify the underlying structure of the S-data set and the Swiss-roll data set. For example in the case of the Swiss-roll data set the GPLVM algorithm does so well because the Isomap algorithm is already capturing the structure correctly.

B.5 MDS

Multidimensional scaling (MDS) (Cox and Cox, 1994) describes a class of methods which provide insight into the underlying structure and relations of a data set by providing a geometry-preserving representation of this data set. In this thesis we refer to MDS as a method which uses proximity measures, or conversely dissimilarity measures, to find an interpretable lower-dimensional representation of the data set in question. In the simplest case the proximity measure between two projected points $\mathbf{x}_i$ and $\mathbf{x}_j$ in an L-dimension Euclidean space is given by:

$$d_{ij} = \left[\sum_{a=1}^{D} (x_{ia} - x_{jb})^2 \right]^{1/2}, \tag{B.8}$$

where for visualisation $L = 2$. Given a δ_{ij} which represents some sort of distance in the original data space one then tries to move the data points $\mathbf{x}_i$ so that d_{ij} optimises the so called Stress or S-function:

$$S = \left[\frac{\sum_{i=1}^{N} \sum_{j<i}^{N} (\delta_{ij} - d_{ij})^2}{\sum_{i=1}^{N} \sum_{j<i}^{N} d_{ij}^2} \right]^{1/2}, \tag{B.9}$$

where N is the number of data points. The following Neuroscale and Isomap algorithms are derived from this basic idea.

It is important to note that the optimisation of standard MDS algorithms will generally scale $O(N^3)$ (Cox and Cox, 1994) for N data points. In the case of very large data sets this might make the direct application of the algorithm problematic without using appropriate approximations (Tipping and Lowe, 1997; Morrison *et al.*, 2003) to speed up the algorithm.

B.6 Neuroscale

Neuroscale (Lowe and Tipping, 1996) is a dimension-reducing and topographic transformation to visualise and analyse high-dimensional data. This is done by setting up an RBF network to predict the locations of the data points in the visualisation space. To integrate the topographic structure into the training process of the neural network a suitable error measure has to be chosen. This error is then to be minimised by optimising the network parameters which in the end determine the transformation. Given $i = 1 : N$ data points $\mathbf{y}_i$ in the input space with the corresponding inter-point distances δ_{ij} and their transformation $\mathbf{x}_i$ in the feature space with the corresponding inter-point distances d_{ij}, the error term E is defined to be

$$E = \sum_{i<j}^{N} (\delta_{ij} - d_{ij})^2 \,, \tag{B.10}$$

$$\delta_{ij} = \sqrt{(\mathbf{y}_i - \mathbf{y}_j)^T (\mathbf{y}_i - \mathbf{y}_j)} \,, \tag{B.11}$$

$$d_{ij} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)} \,, \tag{B.12}$$

(e.g. the summation of the squared differences between the inter-point distances of the D-dimensional data space and the L-dimensional latent space).

In certain cases one may want to augment the information given by the distribution of the points with additional information (e.g. class labels) as well. In this case one can introduce a subjective dissimilarity which may be exploited in the transformation. This prior knowledge implies an alternative topology which should influence the final mapping that results from the objective distribution of the data.

One reasonable way to integrate the subjective information and allow a controlled combination with the objective information is to replace the term δ_{ij} in equation (1) with the alternative β_{ij} defined by:

$$\beta_{ij} = (1 - \alpha) \times \delta_{ij} + \alpha \times s_{ij} \,, \tag{B.13}$$

$$E = \sum_{i<j}^{N} (\beta_{ij} - d_{ij})^2 \,, \tag{B.14}$$

where s_{ij} describes the subjective dissimilarity between the two data points, for example a class label. The parameter α (where $0 \leq \alpha \leq 1$) is now responsible for controlling the ratio between objective and subjective information in the transformation. With $\alpha = 0$ the transformation is purely objective and only depends on the distribution of the data. With $\alpha = 1$ the transformation is purely subjective and only depends on the prior knowledge.

To find the transformation that minimises the error term we have to optimise the RBF network. Given the input data $\mathbf{y}_i$ the transformation is given by the non-linear function $\mathbf{x}_i = f(\mathbf{y}_i; \mathbf{W})$, effected by an RBF with the weights $\mathbf{W}$. Thus the

distance in the visualisation space is given by:

$$d_{ij} = \| f(\mathbf{y}_i) - f(\mathbf{y}_j) \| \ . \tag{B.15}$$

The number of RBF functions are fixed and chosen at the beginning. Therefore we have to optimise the error term with respect to the weights $\mathbf{W}$ of the network. This may be done with one of the many well know iterative optimisation procedures like conjugate gradients (Nabney, 2002).

A simple example of the algorithm which demonstrates how the choice of α influences the projection can be found in the paper by Lowe and Tipping (1996).

Further there is an extension for Neuroscale called *shadow targets* to deal with large data sets (Tipping and Lowe, 1997). Shadow targets takes advantage of the form of the error function to create a more efficient optimisation algorithm by learning only on a subset of the possible data points, which greatly reduces the needed runtime and due to its implementation avoids local minima.

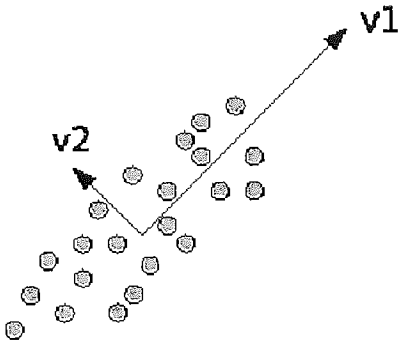
B.7 Isomap

The Isomap algorithm (Tenenbaum *et al.*, 2000) can be seen as a special type of MDS where the distances δ_{ij} are chosen to be of a particular form. These distances are called geodesic and are computed by using a neighbourhood graph over all the data. The idea is to only use local distances for every point and compute the global distances along the distribution of the data. A good way to envisage this is by considering a connected graph given by the triangle with the vertices (A, B, C) as in Figure B.6. The euclidean distance between AC is $\overrightarrow{AC}$. However instead of calculating directly the distance between (AC) we first have to look for the shortest distance given the connections in the graph. Given that the graph is only made up by the edges AB and BC we have to calculate the distance between (AC) by running over B thus $\overrightarrow{AC} = \overrightarrow{AB} + \overrightarrow{BC}$. In general the algorithm can be formulated in the following way:

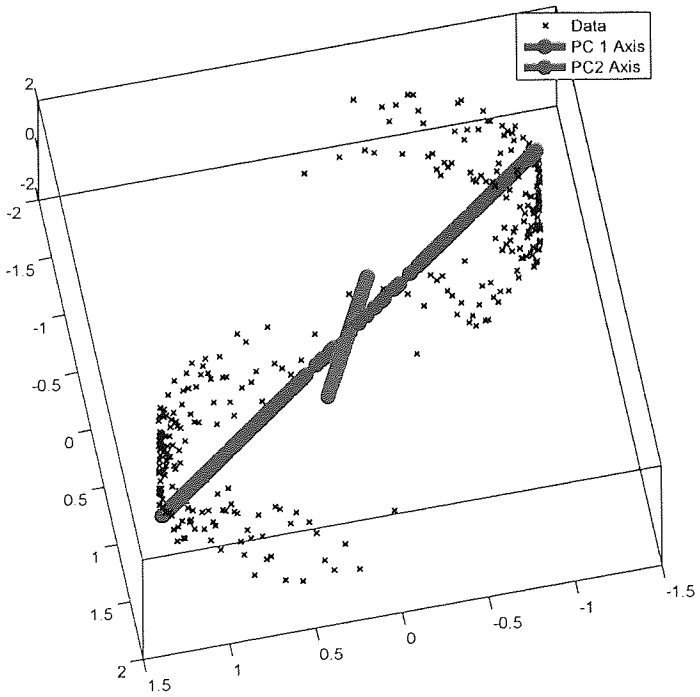
1: Select randomly l points ($l < n$), if the data set is huge to keep the measurement of the pairwise distances tractable.
2: Connect neighbouring points. Either by using the k closest points (Shakhnarovish *et al.*, 2005) or by selecting points which lie closer than a certain threshold ϵ.
3: Compute the matrix D of all pairwise geodesic distances of the k nearest points, construct a graph based on these distances and run Dijkstra's (Tenenbaum *et al.*, 2000; Cormen *et al.*, 1997) algorithm for each point to find the shortest path to all other points (the runtime is cubic to the number of points). Then store the pairwise distances between all points in the symmetric matrix $\mathbf{D}$ with $l \times l$ entries.
4: Centre D and subtract the mean over all rows and all columns.
5: Compute the eigenvalues and eigenvectors of $\mathbf{D}$ and sort the eigenvectors in descending order of the associated eigenvalues.

6: Choose p such that the residual variance accumulated in the $l - p$ last eigenvectors is sufficiently small.

The p retained eigenvectors give the coordinates of the mapped points in a p-dimensional projection space. Due to the non-linear feature of the geodesic distance it is possible to capture non-linear data structures with Isomap. The results can be seen in Figure B.7 where the Isomap algorithm correctly captures the structure of the S-data set and the Swiss-roll data set. However the algorithm is error-prone in sparse data sets or data sets with large amounts of noise. In these cases it might be hard to construct a meaningful distance graph. In the case of sparse data sets a local measure like geodesic distances is generally problematic. In the case of large noise the graph might construct *short circuits* between regions which should be much further apart. For example in the case of the Swiss-roll if one introduces enough noise the inner layer will be directly short-circuited to the outer layer.

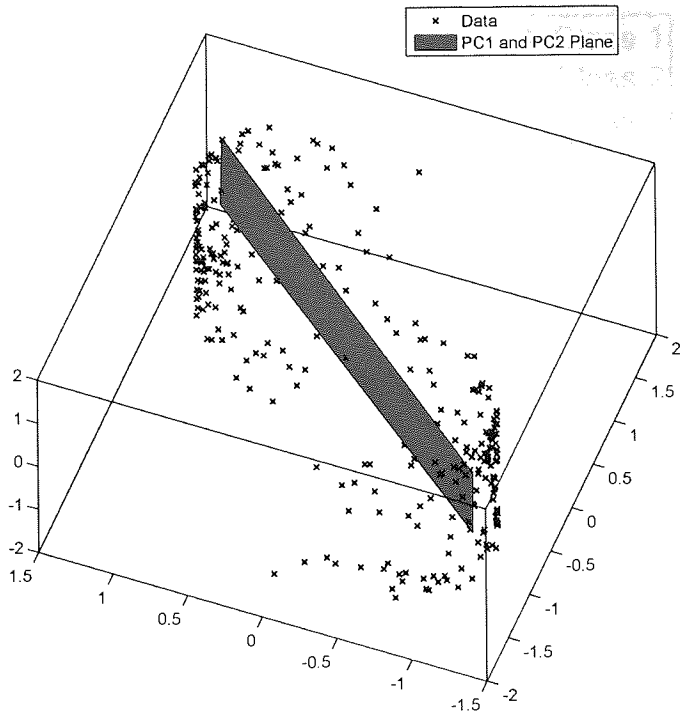


(a) Data set with PC1 and PC2

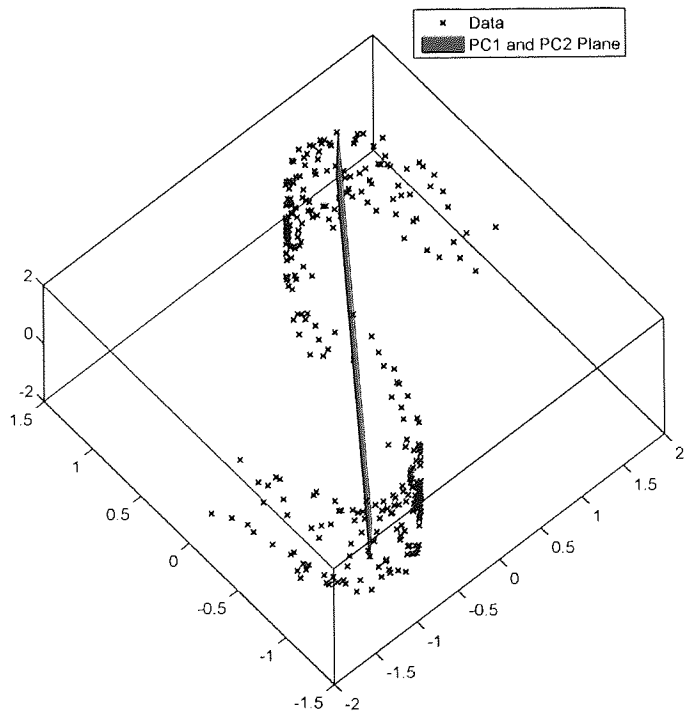


(b) S-shaped data

Figure B.2: a) The two principal components in a two dimensional data cloud. b) The two principal components for the S-shaped data marked as red and yellow bar.

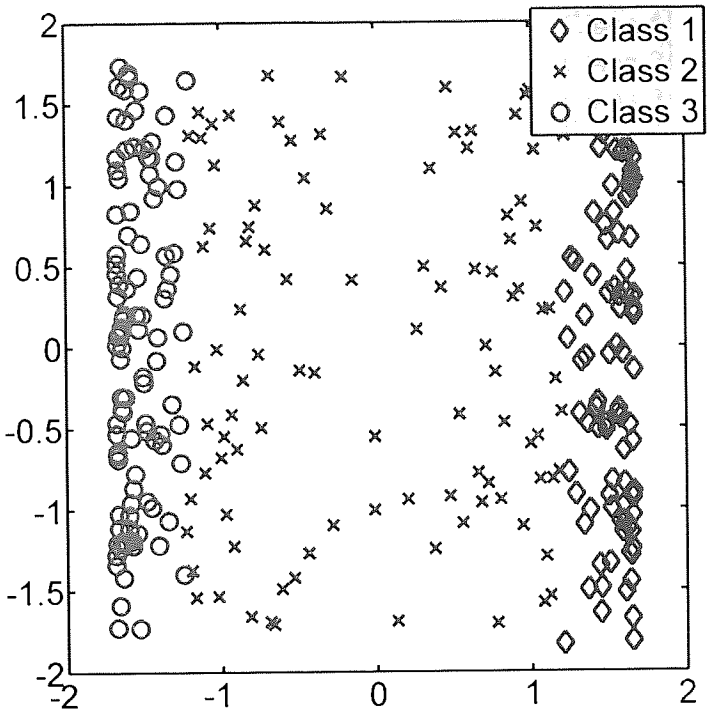


(a) S-shaped data

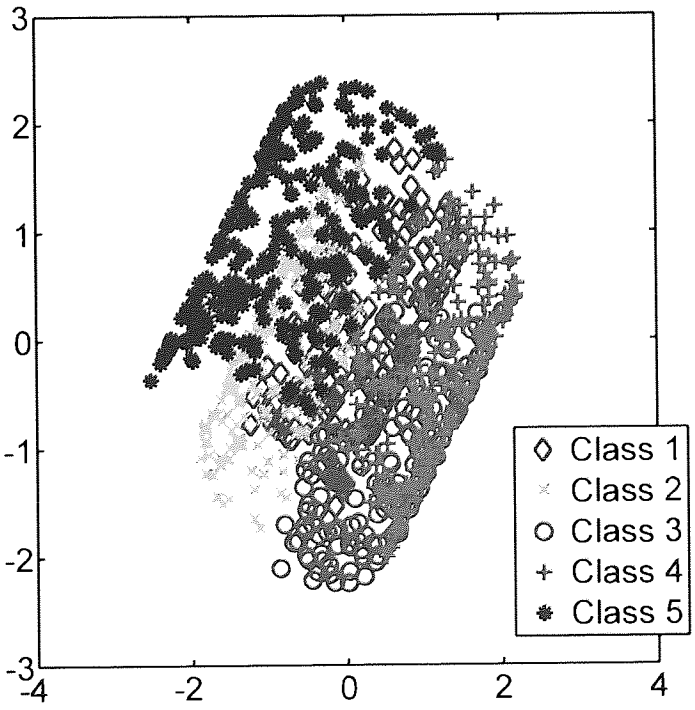


(b) S-shaped data

Figure B.3: a-b) The plane which is spanned by the two principal components for the S-shaped data from different angles.

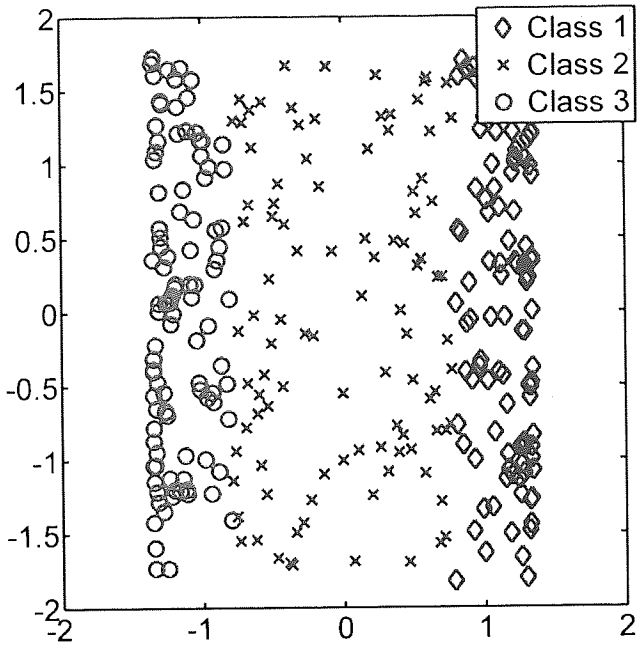


(a) S-shaped data

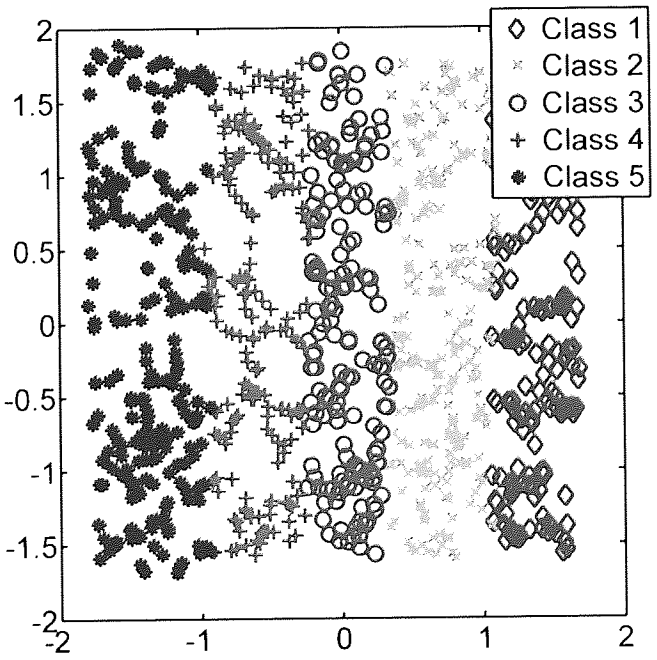


(b) Swiss-roll data

Figure B.4: Demonstration of the projection result of the PCA algorithm on simple data. The structure of the S-shaped data in (a) is captured and one can clearly see that the class structure is preserved. This is not the case with the Swiss-roll data in (b) where PCA fails to preserve the structure of the classes.



(a) S-shaped data with PCA init.



(b) Swiss-roll data with Isomap init.

Figure B.5: Demonstration of the projection result of the GPLVM algorithm on simple data. The structure of the S-shaped data in (a) is captured and one can clearly see how that the class structure is preserved. This is also the case with the Swiss-roll data in (b) where GPLVM, thanks to the Isomap initialisation, manges to capture the structure of the data correctly. As comparison the original Isomap projection used to initialise GPLVM can be found in Figure B.7.

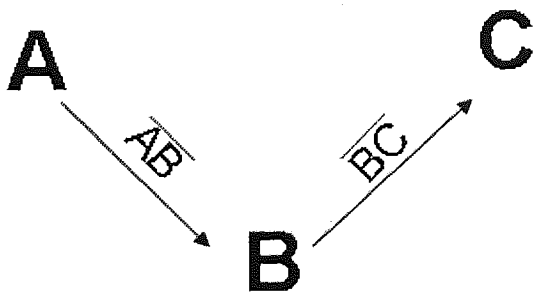
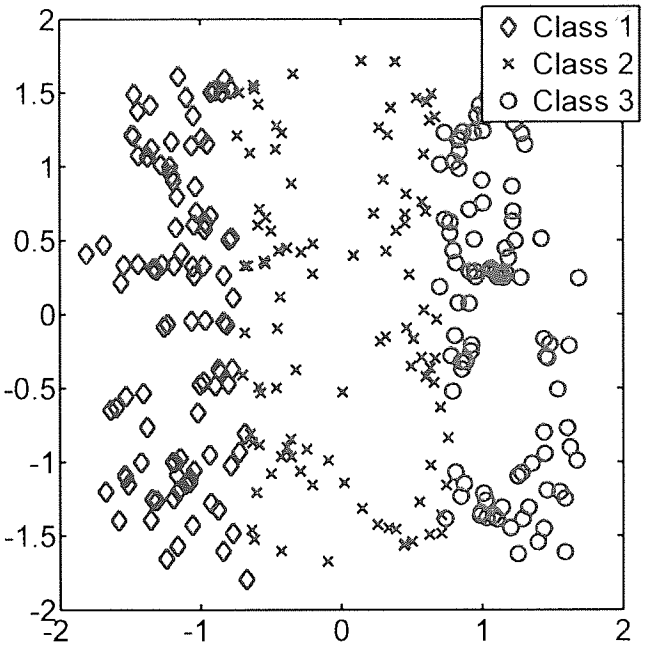
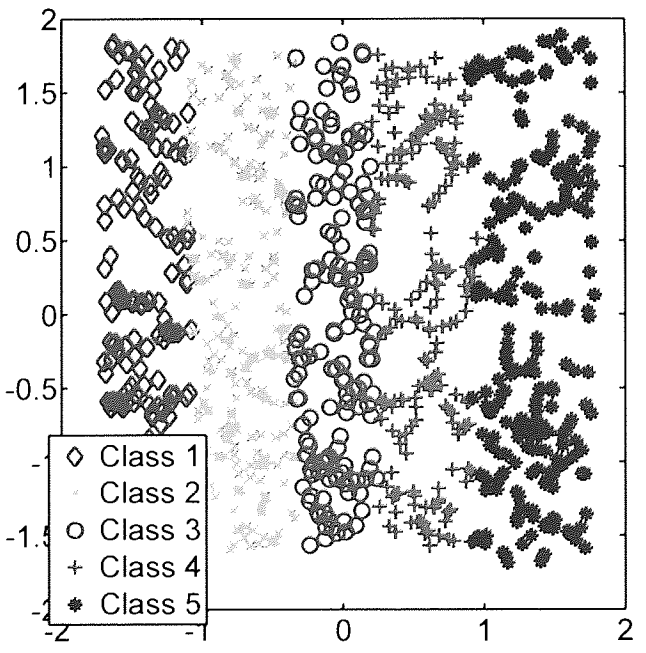


Figure B.6: Triangle with points A,B,C and the distances between them marked by (AB) and (BC). Given that in this connected graph there are only edges between A,B and B,C the geodesic distance between A and C is given by (AB) + (BC).



(a) S-shaped data



(b) Swiss roll data

Figure B.7: Demonstration of the projection result of the Isomap algorithm on simple data. The structure of the S-shaped data in (a) is captured and one can clearly see how the class structure is preserved. This is also the case with the Swiss roll data in (b) where Isomap manges to capture the structure of the data correctly.