Visualisation of Bioinformatics Datasets

SHAHZAD MUMTAZ

Doctor Of Philosophy

ASTON UNIVERSITY

March 2014

Aston University

Visualisation of Bioinformatics Datasets

Shahzad Mumtaz
Doctor Of Philosophy, 2014

Thesis Summary

Analysing the molecular polymorphism and interactions of DNA, RNA and proteins is of fundamental importance in biology. Predicting functions of polymorphic molecules is important in order to design more effective medicines. Analysing major histocompatibility complex (MHC) polymorphism is important for mate choice, epitope-based vaccine design and transplantation rejection etc. Most of the existing exploratory approaches cannot analyse these datasets because of the large number of molecules with a high number of descriptors per molecule.

This thesis develops novel methods for data projection in order to explore high-dimensional biological dataset by visualising them in a low-dimensional space. With increasing dimensionality, some existing data visualisation methods such as generative topographic mapping (GTM) become computationally intractable. We propose variants of these methods, where we use log-transformations at certain steps of expectation maximisation (EM) based parameter learning process, to make them tractable for high-dimensional datasets. We demonstrate these proposed variants both for synthetic and electrostatic potential dataset of MHC class-I.

We also propose to extend a latent trait model (LTM), suitable for visualising high-dimensional discrete data, to simultaneously estimate feature saliency as an integrated part of the parameter learning process of a visualisation model. This LTM variant not only gives better visualisation by modifying the project map based on feature relevance, but also helps users to assess the significance of each feature.

Another problem which is not addressed much in the literature is the visualisation of mixed-type data. We propose to combine GTM and LTM in a principled way where appropriate noise models are used for each type of data in order to visualise mixed-type data in a single plot. We call this model a generalised GTM (GGTM). We also propose to extend GGTM model to estimate feature saliencies while training a visualisation model and this is called GGTM with feature saliency (GGTM-FS). We demonstrate effectiveness of these proposed models both for synthetic and real datasets.

We evaluate visualisation quality using quality metrics such as distance distortion measure and rank based measures: trustworthiness, continuity, mean relative rank errors with respect to data space and latent space. In cases where the labels are known we also use quality metrics of KL divergence and nearest neighbour classifications error in order to determine the separation between classes. We demonstrate the efficacy of these proposed models both for synthetic and real biological datasets with a main focus on the MHC class-I dataset.

Keywords: Major histocompatibility complex, generative topographic mapping, Gaussian process latent variable model, latent trait model, feature saliencies

Acknowledgements

First and foremost I would like to thank my supervisor Prof. Ian T. Nabney who gave me an opportunity to work with him as a PhD student. I am deeply indepted for his patience, support, guidance, feedback and constructive criticism in completion of this thesis which I think would not be possible otherwise. I am also thankful for the freedom he gave me in discussing ideas and also giving me an opportunity to work on an industrial six months short KTP project that helped me to extend my experience working with prediction models using Gaussian processes. His hard working nature made him an inspiring role model for my future life as an academic and a researcher.

Many thanks to Dr. Darren R. Flower for introducing me to the biological side of this thesis and helping me for generating a complex biological dataset using computational biology software tools. I appreciate his efforts tremendously for always giving me time for long meetings and interpretion of the results and for making this collaboration possible.

I am also thankful of the Higher Education Commission (HEC) of Pakistan for grants they provided to The Islamia University of Bahawalpur (IUB), Punjab, Pakistan. Many thanks to the scholarships selection committee of the IUB for giving me an opportunity to do a PhD study abroad. Special thanks to the treasurer office and registrar office for releasing funds on time during this period.

Many thanks to all members of NCRG. A special thanks to Alex Brulo (Senior Server Engineer) for giving support of cluster computer and specially installing an APBS computational biology software required for my research work. I am also thankful for Vicky Bond and Kanchan Patel for giving all the administrative help that I required for the official documentation with prompt and excellent administration skills. I am also thankful of Dr. Amit K. Chattopadhyay for exciting discussions on cricket and the good quality time spent during coffee breaks in the Wolfson lab.

Thanks to all fellow PhD students. Speical thanks to Dr. Diar Nasiev for useful discussion in first year of my studies. Many thanks for Dr. Michel F. Randrianandrasana for discussions on latent trait model. Last but not the least many thanks to my best friend Mr. Zaman Ali, Lecturer at IUB, who always stood by me in difficult times and helped a lot in pursuing administration offices at IUB for the release of funds on time.

I am grateful to my mother, brothers, sister and my wife for their unconditional love and support. It was not possible for me to be here at this stage without them. I appreciate my wife a lot for alone looking after our son, Ahad Shahzad, in my absence. I dedicate this thesis to all my family.

Contents

List of Figures

List of Tables

1 Introduction

CONTENTS

In the last couple of decades, advances in bio-medical research and bio-technology (particularly gene sequencing) have created a tremendous amount of data ranging from pharmaceutical studies, genomics and, proteomics, gene functions and protein-protein interactions. This burgeoning growth in datasets has increased more than ever before the need for analysing them in order to improve our understanding of the underlying biology and this has also brought some challenges for traditional analytic approaches. Some of these challenges are because of high-dimensionality and some because of high-dimensionality with noisy descriptors and some due to heterogeneity of descriptor types.

Analysing these datasets with a latent-variable framework is one useful approach in order to assist biologists to improve their understanding of these complex datasets. In the latent-variable framework we map a high-dimensional dataset to a low-dimensional embedded space: if the embedded space is two- or three-dimensional, the data can be *visualised* in a scatter plot. It is observed that due to the high-dimensionality of much biological data, some existing visualisation models become computationally intractable (Chapter 4) and sometimes a single two-dimensional plot is also not informative enough (because of overlapping cluster) and requires to have more than one two-dimensional plots arranged in a hierarchy to get more detailed insight. In addition, it has been observed that due to noisy or irrelevant features, the low-dimensional embedded space is not informative enough: in this thesis, we show how to use integrated feature weighting approaches either to eliminate or to reduce the impact of noisy features in the parameter learning process of a model (Chapters 4 and 5). Another issue which is becoming important for the machine learning experts is to deal with heterogeneous descriptors in datasets (Chapter 6). We focus on these issues in this thesis in the context of biological datasets.

1.1 The Motivation

All unicellular and multicellular organisms are composed of molecules such as DNA, RNA and proteins, etc. All these molecules interact with each other to perform functions important for the existence of life. Most of the diseases occur due to the changes in these molecules during cells' lifetime (Aluru, 2005).

In the context of molecular biology, it is important to understand the functions of these molecules and regulation of pathways for a variety of different cellular processes. Proteins are the molecules responsible for cellular functions and processes whereas the DNA in the cell is responsible to encode the information required to produce these proteins. A 'protein-coding gene' is a sequence of nucleotide bases which are important for the encoding of amino acids necessary for building the proteins (Lees, 2008). Lees states in her thesis that only a small number of genes become active at a certain time and the process of turning on and off these genes in a cell's lifetime is called *gene regulation*. The interactions between certain proteins and DNA take place because of gene expression[1]. These interactions are fundamentally important for most of the activities within a cell.

In last couple of decades, a significant improvement in experimental methods, particularly automation to achieve a high-throughput analysis, have emerged and caused the generation of a tremendous number of large genetic and proteomic datasets. These technological advances have attracted much of our attention to understand biological processes involved at the molecular level and to understand the gene-regulation process and sequence-level changes important for understanding the causes of disease. For example, the human genome contains over 30 thousand genes in a sequence of over 3 billion pairs (Lees, 2008); each gene can have alternative forms called the *alleles* and each allelic form can be responsible for altering a protein's activity. Nowadays, thousands of allelic forms are known for each gene which lead to large datasets. Most of these allelic forms are available as a primary sequence of amino acids.

For example, a database that maintains primary sequences has grown tremendously in last two decades (see Figure 1.1 for yearly historical growth of UnitProtKB/Swiss-Prot protein database) and more specifically in the case of major histocompatibility complexes class-I (known as the human leukocyte antigen (HLA) in humans), for a single gene HLA-A there were $2,579$ allelic forms (as of the March 2014 release of IMGT[2]/HLA) and this number has doubled in the last four years. The historical growth of known primary sequences[3] of HLAs is shown in Figure 1.2 and most of these alleles relate to less than a dozen genes of HLAs. A usual practice for function prediction is to search for the most similar sequence to the query sequence with a known function. Predicting function from these sequences is important but in comparison it is believed that three-dimensional representations are more informative than primary sequences (Chothia and Lesk, 1986; Laskowski et al., 2005). Because, determining three-dimensional structures experimentally is a time consuming and costly task, therefore fewer such structures are known compared to primary sequences of amino acids. For example, in the case of HLAs, only a few hundred three-dimensional structures are known compared to thousands of known sequences (Berman et al., 2000). Recent advances in *in-silico* approaches has made it possible to predict the

[1] Gene expression is the process to encode an information from a gene in the synthesis process to give a gene product. Quite often the gene product is protein.

[2] International ImMunoGeneTics databases.

[3] A primary sequence or primary structure is the representation of the amino acids in the polypeptide chain.

three-dimensional structures using homology modelling.

Figure 1.1: The number of known primary sequences on a yearly basis and included in the UnitProtKB/Swiss-Prot protein database. The recent surge in the number of submissions received by the database is clearly shown (adapted from [http://web.expasy.org/docs/relnotes/relstat.html]).

In conclusion, this accumulation of a vast amount of data does not give as much insight into the biological interpretation as scientists would wish. Interpreting these datasets is impossible manually, though it will be very much useful to get the immediate value of the information retrieved from these growing datasets in setting future research; observing patterns at gene levels and the role of these genes in a disease and development of an organism (Lees, 2008). For example, in the case of HLAs, it is important to analyse proteins' allelic forms based on the similarities of their primary sequences and three-dimensional structural descriptors in order to develop epitope-based vaccines (Doytchinova et al., 2004; Doytchinova and Flower, 2005) and other functions such as mate choice (Havlicek and Roberts, 2009), smell recognition (Santos et al., 2005) and an important clinical role in transplantation rejection (Su et al., 2014), etc.

The major focus of this thesis is to investigate data visualisation techniques which are ways of representing a high-dimensional dataset in terms of a low-dimensional embedded space, to investigate datasets in order to improve understanding of any hidden patterns (i.e. cluster of similar patterns). We also investigate how to determine the impact of each individual descriptor while training a data visualisation model with an integrated saliency estimation approach. These data visualisation techniques and combined feature saliency estimation approaches are mainly used in this thesis for the purpose of analysing

Figure 1.2: The number of HLA alleles named each year and included in the IMGT/HLA database. The recent surge in the number of submissions received by the database is clearly shown (adapted from (Robinson et al., 2013)).

biological datasets. In the rest of the thesis, we represent each dataset as a matrix with rows representing data patterns and columns as descriptors.

1.2 Publications from the work presented in this thesis

This thesis involves and complements some work presented in earlier publications. Some of the publications are pending for the submission. Chapter numbers given explain the content of published and planned papers related to this thesis.

- S. Mumtaz, I. T. Nabney, and D. R. Flower. Novel visualisation methods for protein data. *In Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2012 IEEE Symposium*, pages 198–205, May 2012. San Diego, California, USA (Chapter 4).

- S. Mumtaz, I. T. Nabney, and D. R. Flower. Multi-level visualisation using Gaussian Process Latent Variable. *In Proceedings of the 5th International Conference Information Visualization Theory and Applications (IVAPP)*, pages 122–129, january 2014, Lisbon Portugal, SCITPRESS.

- S. Mumtaz, I. T. Nabney, and D. R. Flower. Scrutinizing Human MHC Polymorphism: Supertype Analysis using Poisson-Boltzmann Electrostatics and Clustering. *Oxford Bioinformatics Journal*, (submitted) (Chapter 4).

- S. Mumtaz and I. T. Nabney, A Generalised Generative Topographic Mapping for Visualising a mixed-type data and simultaneous feature selection. (in preparation) (Chapters 5 and 6).

1.3 Notation and Conventions

The convention, from the mathematical notation perspective, will be that scalar values (such as x_{nd}, z) are represented with an italic typeface whereas vectors are expressed as bold lower-case letters (such as $\mathbf{x}_n, \mathbf{z}_n$) and matrices are represented as bold capitals (such as $\mathbf{X}, \mathbf{Z}$). Exceptions to these conventions also appear but they are explicitly mentioned. Other symbols we use are explained in Table 1.1.

Symbol	**Explanation**
N	number of data points
n	label for a data point
D	number of data dimensions (features)
d	label for a feature
L	number of RBF centres
l	label for a RBF centre
K	number of latent points
k	label for a latent point
M	number of latent dimensions
$\mathbf{X}$	data matrix of dimension $N \times D$
$\mathbf{Z}$	latent projection matrix of dimension $N \times M$
I	identity matrix
p	probability density function
$\mathcal{L}$	likelihood
$\mathcal{R}$	real values indicator in case of mixed-type data analysis
$\mathcal{B}$	binary values indicator in case of mixed-type data analysis
$\mathcal{C}$	multi-category values indicator in case of mixed-type data analysis

Table 1.1: Notation with explanation.

1.4 The structure of this thesis

Chapter 2: This chapter reviews the basics of bioinformatics focussing on proteins. Then, we review methods of computing the electrostatic potential energy of proteins' structures and an existing analysis method of this property. We also explain the basics of the major histocompatibility complex (MHC) protein family by reviewing previous analysis outcomes in terms of supertype identification and its importance. At the end, we explain the process that we adapt for modelling three-dimensional structures of MHC class-I proteins and computation of electrostatic potential energy.

Chapter 3: We review existing visual data mining systems including DVMS, which we have re-designed and re-developed in order to improve it. We then describe a selection of data projection algorithms, mainly variants of the Generative Topographic Mapping (GTM), including GTM with feature saliency, hierarchical GTM and the latent trait model (LTM) (a generalisation of GTM). We also review other projection algorithms: principal component analysis (PCA), neuroscale (NSC) and the Gaussian process latent variable model (GPLVM). At the end we review some of the visualisation quality metrics we use in this thesis to compare models.

Chapter 4: In this chapter, we propose variants of GTM and its extensions (including GTM with simultaneous feature saliency and hierarchical GTM) where we adapt log-transformations to avoid numerical precision problems at certain steps of the model's parameter learning process. The effectiveness of these proposed variants is demonstrated both for synthetic and a real dataset of electrostatic potential values of MHC class-I. For the purpose of comparison, we also present the visualisation results of the MHC class-I dataset with other data projection algorithms.

Chapter 5: We extend a framework of estimating feature saliencies while training a visualisation model for discrete-typed data. This has been used often with GTM-like algorithms in cases of real-type datasets where the noise model is considered as Gaussian. We extend this approach with LTM-like algorithms where appropriate noise models are considered in accordance with the type of features (for example Bernoulli for binary features and multinomial for multi-category features). We demonstrate experimental results both for synthetic and real datasets in order to show their effectiveness.

Chapter 6: First, we review existing work for visualising mixed-type data and then propose the generalised GTM (GGTM) which is based on the assumption of selecting appropriate noise models for each type of features in a mixed-type dataset to visualise on a single two-dimensional plot. Then, we then present an extension of GGTM to simultaneously estimate feature saliencies. At the end, we present the experimental results and discuss them thoroughly both for synthetic and real datasets.

Chapter 7: This chapter first concludes the outcomes we learnt from each chapter in this thesis and then discusses possible future extensions of this work.

Appendix A: In this appendix, we explain the process we adapted for modelling three-dimensional protein structures using homology modelling and calculations of electrostatic potential values using a software tool called the Adaptive Poisson-Boltzmann Solver (APBS).

Appendix B: This appendix shows an MHC class-I dataset visualisation with Neuroscale model using different number of basis functions.

Appendix C: In this appendix, we explain the mixture model for the GTM-FS/LTM-FS visualisation, derivation of EM algorithm of the GTM-FS/LTM-FS and some additional results of the LTM-FS.

Appendix D: In this appendix, we explain the mixture model for the GGTM-FS, derivation of the GGTM-FS visualisation model and some additional results of the GGTM and the GGTM-FS models.

2 Bioinformatics and Major Histocompatibility Complexes (MHCs)

CONTENTS

This chapter describes a few fundamental concepts in bioinformatics focusing on the protein family of major histocompatibility complexes (MHCs) in humans (known as human leukocyte antigens (HLA)). We then briefly explain the potential benefits of analysing the protein family of MHCs. The dataset we generated for our analysis is related to the electrostatic potential energy of MHCs. We review here some existing methods for electrostatic potential energy calculation and its analysis. At the end of the chapter, we briefly explain the process that was used to generate an electrostatic potential energy dataset for MHC class-I.

2.1 Bioinformatics

Bioinformatics is the field of studying biological activity of macromolecules using computational technologies. The most important organic macromolecules are carbohydrates, lipids, proteins and nucleic acids (Campbell and Reeca, 2008). Luscombe et al. (2001) states that in general there are three aims of bioinformatics. The first aim is to maintain a database accessible for researchers to analyse, such as a protein data bank[1], for three-dimensional macromolecules, or the Swiss-prot and TremblIMGT/HLA[2] databases for maintaining HLA sequences. The second aim is to develop tools that are helpful for analysing these datasets and to understand the functions of macromolecules. The third aim is to use these analysis tools for extracting biologically meaningful information about macromolecules. In bioinformatics, systems for analysis are usually developed for particular biological contexts and these systems are compared with only a few related available similar systems (Luscombe et al., 2001). However, there is a need to develop more general analysis systems for common activities across a greater number of datasets.

In the next sub-sections, we will focus on proteins, explaining what proteins are, how they are constituted in general, which structures of proteins are known, and what are the associated functions of these different protein structures mainly focusing on the electrostatic interaction of three-dimensional protein structures.

2.1.1 Proteins

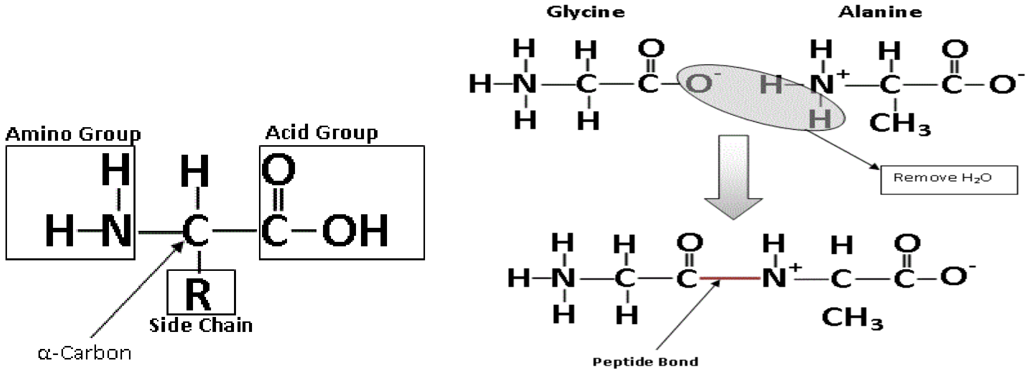
Proteins are composed of one or more polypeptide molecules. Each polypeptide molecule is composed of chains of amino acids linked together by peptide bonds. There are 20 standard amino acids. Each amino acid consists of an amino group (NH_2-), the acid group ($-CO_2H$), a side chain denoted by R, and a central carbon atom (also called the

[1] http://www.pdb.org/pdb/home/home.do

[2] http://www.ebi.ac.uk/imgt/hla/

α-carbon atom) to which the side chain is attached. The general structure of an amino acid is shown in Figure 2.1(a).

Amino acids are linked to each other with peptide bonds[3] and an example of linking two amino acids making a dipeptide is shown in Figure 2.1(b).



(a) (b)

Figure 2.1: Protein composition components. (a) General structure of an amino acid. (b) Dipeptide showing peptide bond between two amino acids.

Functions of proteins are related to their structures. Protein structures can be categorised into four levels (Langel et al., 2010) as follows (see also Figure 2.2).

Primary Structure: The primary structure or *protein sequence* of a protein represents the order in which the amino acids are joined together in a polypeptide chain. Usually primary structures are represented using an abbreviated form of each amino acid name (e.g. for Glycine three letter form 'Gly' or one letter form 'G'). The protein sequence is important for finding similarities in the amino acid sequences. If the amino acid sequences of two proteins are at least 20% similar then they are said to be homologous (Langel et al., 2010).

Secondary Structure: The secondary structure of a protein is that which is formed by hydrogen-bonding[4] patterns. The secondary structure of proteins is the localized three-dimensional organization of the polypeptide chain. There are three common secondary structures: α-helices[5], β-sheets[6] and turns or bends[7].

[3] A peptide bond is a chemical bond formed between two molecules by a chemical reaction between the acid group of one molecule with the amino group of the other molecule and thereby releasing a water molecule (i.e. H_2O).

[4] A hydrogen bond is an attractive interaction between a hydrogen atom and electronegative atoms, such as nitrogen, oxygen or fluorine, belonging to another molecule.

[5] An α helix is a spiral conformation where every backbone N-H group donates a hydrogen bond to the backbone $C = O$ group of amino acids with four earlier residues.

[6] A β-sheet is a plane composed of strands of polypeptide chains as an extended secondary structure showing the side chains of amino acids projected alternatively above and below the plane of the sheet.

[7] Turns or bends are the elements of secondary structures in proteins where the direction of the polypeptide changes or reverses.

Tertiary Structure: The tertiary structure of proteins is the three-dimensional organization of atoms in a polypeptide chain. In tertiary structure, the global organization of the polypeptide chain is given by each atom's exact position in space. There are two important factors for determining the tertiary structure. One is the primary structure and the second is the environment[8]. In this thesis, we use the term three-dimensional structure to refer to the tertiary structure.

Quaternary Structure: There are some proteins such as myoglobin[9] that have a single polypeptide chain, but many proteins are an assembly of multiple chains. Quaternary structures are concerned with the organization of polypeptide chains to make multi-subunit functional proteins (e.g. the quaternary structure of haemoglobin[10] is made up of four chains where two are α-chains and the other two are β-chains where each is similar to a myoglobin molecule).

Figure 2.2: Levels of protein organization (adapted from [https://www.genome.gov/glossary/resources/protein.pdf]).

[8] An environment is treated as a broad concept including the solution composition, all the available enzyme systems involved in post-translational modifications, and the transport system involved in transferring protein between different compartments of cells where modifications have been accomplished.

[9] The myoglobin is a protein mostly found in muscle tissue to act as an oxygen carrier.

[10] The pigment responsible for carrying oxygen in red blood cells of vertebrates.

2.1.2 Protein Function Prediction

Recent research in the field of bioinformatics has provided an extensive set of protein amino acid sequences available in the form of sequence databases such as Swiss-Prot and TrEMBL[11], IMGT/HLA[12], etc. The functions of very few protein sequences in these databases are known today. Generally, protein functions are related to water balancing, nutrient transportation and contraction of muscles. Some proteins function as enzymes and hormones, and most immune system molecules are proteins. Therefore, predicting the functions of protein sequences is important and is often achieved by searching for the most similar (homologous) sequences with already known functionality. According to the August 2010 release of Swiss-Prot and TrEMBL database, there are $519,348$ and $11,636,205$ known sequence entries respectively.

The three-dimensional structure of a protein can also be used for understanding function by comparing it with already known structures with known function (Thornton et al., 2000). Gupta et al. (2005) states that two sequences with high similarity in primary sequences are expected to have similar three-dimensional structure whereas two similar three-dimensional structures may not have a strong similarity in their amino acid sequences. The known three-dimensional structure of the human α-globin and myoglobin are very similar in their three-dimensional structure but are quite different in their amino acid sequences with 26% identity (Langel et al., 2010). Predicting protein function from structure is known to be a better way than predicting functions from amino acid sequence similarity, and there are two reasons for this. First, a protein's three-dimensional structure is more conserved than the amino acid sequence (Chothia and Lesk, 1986). Second, the regions where a protein can interact with a ligand[13] are determined and can be used for comparison with other proteins (Laskowski et al., 2005).

X-ray Crystallography (Smyth and Martin, 2000), Nuclear Magnetic Resonance Spectroscopy (NMR) (Gronwald and Kalbitzer, 2004) and Electron Microscopy (EM) (Meyers, 2007) are the standard techniques for acquiring three-dimensional protein structures. These experimental methods are costly and time consuming (Lee and Verleysen, 2007). For example, Stevens (2003) stated that the average time required to predict a soluble protein target is approximately one year for a novel structure determination but this can be achieved much faster at an increased cost. Stevens also stated that cost for drug target three-dimensional structure determination varies for example for novel drug target of

[11] Swiss-Prot and TrEMBL details are available at http://www.expasy.org/sprot/

[12] http://www.ebi.ac.uk/imgt/hla/

[13] A ligand can be an atom, molecule or ion that can bind to specific binding site of the protein. Binding is the key to protein function.

human membrane protein the cost was $2.5 million with only 10% success rate and for a soluble human protein (e.g. kinases, proteases etc) the cost was $450,000 with only 35% success rate. Therefore, very few three-dimensional protein structures are known today in comparison to the large number of known protein amino acid sequences (Langel et al., 2010). A well-known database that maintains three-dimensional protein structures is called the Protein Data Bank (Bernsten et al., 1977) and according to the September 2010 release, there are 68,288 known protein structures.

Due to the time consuming and costly experimental methods, researchers have developed computational methods such as homology (comparative) modelling for predicting the three-dimensional protein structures for known amino acid sequences. For predicting the three-dimensional structure of the protein sequence whose three-dimensional structure is not known, if the amino acid sequence of the known three-dimensional structure and target protein sequence are at least 30% similar in their length and percentage of sequence identity[14] then predicting the three-dimensional structure from the known structure is usually close to being correct (Krieger et al., 2003). The detailed procedure and steps involved for predicting the three-dimensional structure of MHC class-I proteins using homology modelling are explained in Appendix A. In this thesis, our focus is study of the electrostatic potential energy of proteins and in the next two sections, we give a brief description of electrostatic potential energy of proteins and also discuss existing analysis tools. We then discuss the protein family known as the major histocompatibility complex (MHC) and its biological importance.

2.1.3 Electrostatic Potential Energy of Proteins

Protein interactions are important for their physiological functions. These interactions are based on molecular interaction fields such as the electrostatic potential energy. This is the energy of an electrically charged particle in an electric field at any point around a protein structure. Dong et al. (2008) state that computational electrostatic systems are usually described as ‘explicit-solvent’ or ‘implicit-solvent’ methods. Explicit-solvent methods treat the solvent with full atomic detail making it computationally intensive. However, implicit-solvent methods treat the solvent in its average effect on solute and are thus much faster to compute. The later approach has opened new horizons for the researchers in the field of drug design and computational structural biology (Azuara et al., 2006). However, we use an implicit-solvent system in our research.

[14] A sequence identity is considered as the degree of correspondence between sub-sequences of the amino acid sequence whose structure is known and the target sequence whose structure we intend to predict using homology modelling.

One of the popular method of calculating the electrostatic potential energy of protein structures is by solving the Poisson-Boltzmann equation (Polozov et al., 2005). This equation represents the electrostatic potential in a solvent around the protein three-dimensional structure

$$-\bigtriangledown(\epsilon(\mathbf{r})\bigtriangledown\varphi(\mathbf{r}))=4\pi(\rho_0(\mathbf{r}))+\rho_1(\varphi(\mathbf{r})), \tag{2.1}$$

where $\mathbf{r}=(x,y,z)\in\mathbb{R}^3$, φ represents the electrostatic potential, and ϵ represents the dielectric permeability[15]. and ρ_0 represents the charge distribution defined by the molecule as,

$$\rho_0(\mathbf{r})=\sum_i ez_i\delta(|\mathbf{r}-\mathbf{r}_i|). \tag{2.2}$$

Here z_i represents the elementary charge of the ith atom of the molecule, r_i is the radius vector of the ith atom, e represents the elementary charge which is the absolute of the electron charge and δ represents the Dirac delta function.

$$\rho_1(\mathbf{r})=\sum_i n_i ez_i\exp(ez_i\varphi(r)/k_BT), \tag{2.3}$$

where n_i represents concentration of ions of the ith kind, z_i represents the charge of an ion as an elementary charge of the ith kind, k_B represents the Boltzmann constant and T is the absolute temperature, which is often assumed to be 300 K. If the electrostatic potential is small enough ($\varphi\ll k_BT/e$), then equation (2.1) reduces to its linearized form

$$-\bigtriangledown(\epsilon(\mathbf{r})\bigtriangledown\varphi(\mathbf{r}))+\kappa^2\varphi=4\pi\rho_0(\mathbf{r}), \tag{2.4}$$

where $\kappa^2=4\pi^2\sum_i n_iz_i^2/k_BT$ represents the ion density.

Dong et al. (2008) state that no analytical solution of the Poisson-Boltzmann equation is known whereas numerical solvers are available. They state that the first numerical method for solving the Poisson-Boltzmann equation was introduced by Warwicker and Watson (1982) to compute the electrostatic potential of an enzyme's active site. They also state that there are three methods (i.e. Finite-Element, Finite-Difference and Boundary-Element methods) used most often for solving the Poisson-Boltzmann equation, and all these methods use the concept of discretization by dividing the region of interest into small sub-regions. Software tools are available that implement such numerical methods. Delphi[16] and University of Houston Brownian Dynamics (UHBD)[17] software tools use finite-

[15] Permeability is the magnetization degree of a material in a magnetic field response.

[16] http://wiki.c2b2.columbia.edu/honiglab_public/index.php/Software:DelPhi

[17] http://adrik.bchs.uh.edu/uhbd.html

difference method, whereas Adaptive Poisson Boltzmann Solver (APBS)[18] and Charged Particle Optics (CPO)[19] software tools use the finite-element and the boundary-element methods respectively.

We use the Adaptive Poisson-Boltzmann Solver (APBS) for calculating electrostatic potential map for protein structures: detailed steps for its use are explained in Appendix A. The APBS software tool is based on a number of libraries including the Finite-Element toolkit (FEtK)[20], Parallel algebraic MultiGrid (PMG)[21] and Minimal Abstraction Layer for Object-Oriented C/C++ (MALOC)[22] programs. Not many systems exist to analyse proteins electrostatic potential energy. In the next section, we discuss an existing tool for the analysis of the electrostatic potential energies of a group of proteins.

2.1.4 Application for Comparing Protein Electrostatic Potential Energies

A web-based tool called WebPIPSA (Ritcher et al., 2008) allows a user to compare electrostatic potential energies for a set of protein three-dimensional structures. WebPIPSA works by first superimposing the protein structures, and then calculating electrostatic potentials using APBS or UHBD. WebPIPSA is based on a method of comparing protein structures called Protein Interaction Property Similarity Analysis (PIPSA) which was proposed by Blomberg et al. (1999). The similarity or dissimilarity of the electrostatic potential energy for a pair of proteins is calculated using similarity indices and distance measures for the region of interest on the protein structures. The similarity indices are calculated using

$$SI_{12} = \frac{2(p_1, p_2)}{(p_1, p_1) + (p_2, p_2)}, \tag{2.5}$$

where

$$(p_1, p_2) = \sum_{i,j,k} \phi_1(i,j,k)\phi_2(i,j,k). \tag{2.6}$$

Here $\phi_m(i,j,k)$ represents the electrostatic potential at grid point (i,j,k) position and m represents the protein structure index. The electrostatic distance $D_{a,b} = \sqrt{2 - 2SI_{a,b}}$ is used as input to a hierarchical clustering algorithm and the result is displayed in the form of a dendrogram and a coloured matrix (heat map). The sample output of comparing ten protein structures of the HLA-A gene using this web service is shown in Figure 2.3. These results are useful for classifying and visualizing the correlation among a set of proteins.

[18] http://www.poissonboltzmann.org/apbs/

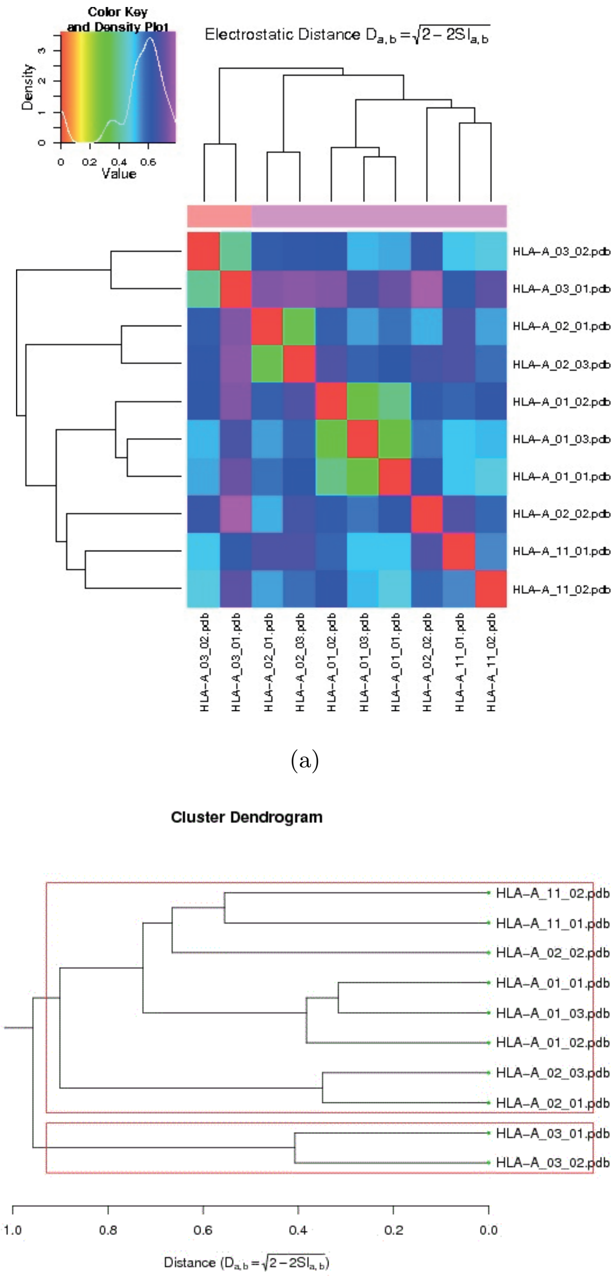
[19] http://simion.com/cpo/bem.html

[20] http://www.fetk.org/

[21] http://www.fetk.org/codes/pmg/index.html

[22] http://www.fetk.org/codes/maloc/index.html

This tool has a number of limitations: the first is that it is impractical to compare a few thousand proteins' electrostatic potential energy maps with the visualization technique of a heat map used by the tool and the second is the way the similarity indices are calculated for a pair of proteins' electrostatic potential maps that can give a false measure of similarity.



(a)

(b)

Figure 2.3: WebPIPSA results of protein classification for 10 structures of HLA-A proteins. (a) Colored matrix representation. (b) Dendrogram.

2.1.5 Major Histocompatibility Complexes (MHCs)

The Major Histocompatibility Complex (MHC) is a group of genes found in most vertebrates and is related to an immune response. Kindt et al. (2007) states that MHC was first studied as the genetic complex that has the ability of accepting or rejecting the transplanted tissue of an organism from one member to another member of the same species. MHC molecules bind with peptides and appear on the cell surface where they are recognized by T-Cell Receptor (TCR) bearing T cells. The MHC takes part in the development process of humoral[23] and cell-mediated[24] immune responses to cell surface molecules and these responses are due to the antigens. These antigens are called 'Histocompatibility Antigens'.

During the mid 1930s, Gorer identified blood-group antigens while studying inbred strains of mice and grouped these antigens into four groups from I to IV. During the 1950s, Gorer and Snell collectively conducted experiments on mice and concluded that genes in the Group-II encoded antigens and these antigens took part in the rejection of transplanted tissues. Snell called these genes as 'Histocompatibility Genes' which are now called Histocompatibility-2 or $H-2$. Snell's work on mice became the basis for the study of Human Leukocyte Antigen (HLA) which is referred to as the Major Histocompatibility Complex in humans. Snell received a Nobel prize in 1980 in physiology for this work (Kindt et al., 2007; Parham, 2000).

The Major Histocompatibility Complex is polygenic (multiple genes) and polymorphic (multiple alleles). Allelic forms of genes that lie close together are codominant[25]. A related group of MHC alleles on a single chromosome that are inherited together is called haplotype. On the basis of biological properties and chemical structure, MHC proteins are classified into two classes: Class-I and Class-II. MHC class-I molecules typically express peptides from proteins synthesized within the cell (endogenous processing pathway). MHC class-I proteins are encoded by three loci: HLA-A, HLA-B and HLA-C. MHC class-II proteins are primarily derived from the endocytosed extracellular protein (exogenous processing pathway). MHC class-II proteins are also encoded by three loci: HLA-DR, HLA-DQ and HLA-DP.

Both classes differ in their structure. MHC class-I have the heavier α-chain which is subdivided into three sub-regions (i.e. $\alpha 1$, $\alpha 2$ and $\alpha 3$) and TCR binding domain and a

[23] Humoral immune response takes place by secreted antibodies which are produced in the B cell.

[24] Cell-mediated immune response is different from humoral response, as it does not involve antibodies but involves activation related to macrophages, natural killer cells (NK), antigen-specific cytotoxic T-lymphocytes and the release of different cytokines in response to antigens (Heinonen and Perreault, 2008).

[25] Codominant means 'inherits same number of alleles maternally and paternally'.

conserved immunoglobin domain which binds CD8[26]. MHC class-II molecules have also two chains: the α chain and the β-chain. The α-chain is divided into two sub-regions (i.e. $\alpha 1$ and $\alpha 2$) and the β-chain is also divided into two regions (i.e. $\beta 1$ and $\beta 2$). The major function of MHC class-II proteins is to present extracellular antigenic peptides to the CD4[27]. Figure 2.4 shows schematic structures of both classes of MHCs.

Figure 2.4: Structures of MHC type molecules (adapted from [http://what-when-how.com/wp-content/uploads/2012/04/tmp4C9.jpg]).

Only a few three-dimensional structures of alleles of HLAs are known and these were obtained through experimental techniques like NMR or X-Ray crystallography. However, in comparison with the three-dimensional structures, a large number of primary structures (i.e. amino acid sequences) are known, and updated on a regular basis in the IMGT/HLA database (Robinson et al., 2003). According to the July 2010 release of IMGT/HLA database, there were $5,302$ allele sequences of the Major Histocompatibility Complex (MHC) and since then due to advances in measurement devices the number of known sequences has almost doubled (as of March 2014, there are $10,533$ sequences in the database). Due to this tremendous increase in the number of protein sequences in the last

[26] CD8 stands for cluster of differentiation 8. This is a transmembrane glycoprotein which works as a co-receptor for the T-cell receptor (TCR). Both CD8 and TCR bind themselves to major histocompatibility complex but CD8 only relates to MHC class-I

[27] CD4 stands for cluster of differentiation 4. This is a glycoprotein which exists on the surface of immune cells (e.g. T helper cells, monocytes, macrophages and dendritic cell).

few years, existing analysis methods face challenges and this gives rise to openings for the analysts to improve existing methods and to develop new algorithms to analyse such large datasets.

Peptide[28] specificities (characterisation) across this vast set of MHCs are thought to form distinct clusters or supertypes (Sette and Sidney, 1999). Previous studies, both experimental (Greenbaum et al., 2011) and computational (Harjanto et al., 2014), have attempted to define such supertypes. Given the number and diversity of MHCS, the only tractable approach is a computational one. Our intention is to develop tools to analyse and cluster alleles of HLAs. In the next section, we briefly explain the importance of MHC clustering with a review of previous analysis.

2.1.6 Previous work related to MHCs clustering

Doytchinova et al. (2004) state that classification of MHCs into supertypes based on the similarity in peptide-binding specificities is important for developing epitope[29]-based vaccines. They considered MHC class-I supertype clustering using $\alpha1$ and $\alpha2$ sub-regions of the α-chain (which is defined in the first 180 amino acid residues). They computed CoMSIA[30] fields and analysed them using hierarchical clustering with the Sybyl[31] software. They also computed Molecular Interaction Fields (MIF)[32] and analysed them using PCA implemented in the GRID software[33] with different probes (substances) such as water, hydrogen, etc. In their work, both the methods gave 77% consensus (similarities in grouping) in defining eight supertypes for the available alleles (i.e. 783 MHC class-I three-dimensional molecules of humans predicted using homology modelling process) in the IMGT/HLA database (Robinson et al., 2003).

Doytchinova and Flower (2005) identified supertypes for MHC class-II considering only the first 80 amino acid residues of the α chain type and the first 90 residues of the β chain type. They used a hierarchical clustering method with CoMSIA fields computed by the Sybyl software tool. In the CoMSIA field-based analysis amino acids outside the binding site were ignored. For non-hierarchical clustering, each amino acid in the binding site

[28] A peptide is a short chain of two or more amino acids that are linked by a connecting carboxyl group of one amino acid with the amino group of the other amino acid.

[29] An epitope is defined as the local region on the surface of the antigen responsible for bringing an immune response and also combines with a certain antibody to counter that response.

[30] Comparative Molecular Similarity Indices (CoMSIA) fields include properties such as steric bulk, electrostatic potential, hydrophobicity, hydrogen-bond donor and acceptor.

[31] http://tripos.com/

[32] The Molecular Interaction Field (MIF) is a uniform grid of points surrounding the whole protein or the specific regions over the proteins.

[33] http://www.moldiscovery.com/soft_grid.php

was considered using five z-descriptors: $z1$ for hydrophobicity[34], $z2$ for steric bulk[35], $z3$ for polarity[36] and both $z4$ and $z5$ for electronic effects. They used the last four levels of the hierarchy for supertype identification. A matrix was generated where each row represents a protein and a number of columns that were equal to the five times more the number of amino acids in the binding site. K-means clustering was applied using the MDL-QSAR (Quantitative Structure Activity Relationship)[37] software tool by setting K, the number of clusters, equal to the number of clusters generated by the hierarchical clustering method. The members of clusters from both the techniques were compared and based on commonality between the clusters (with 84% consensus from both the methods), twelve supertypes were defined. Following these studies, several different techniques can be developed to provide a more detailed analysis of protein structures by considering the whole structure or different fragments of the molecules.

Ghaffar and Nagarkatti and Cainelli and Vento (2002) state that the MHC contains a number of genes that control several antigens which are important in transplantation for rejecting the graft between same species. They also state that the relocation of tissue between the same species is called 'Allograft', between the same species with identical genetic makeup is called 'Isograft' and between different species is called 'Xenograft'. If the donor and recipient have maximum similarity in their MHCs there is less chance of rejection of the graft and maximum duration of the graft survival is in the order of high to low is in 'Isograft', 'Allograft' and 'Xenograft' respectively.

2.1.7 The electrostatic potential dataset of MHC class-I

A set of protein sequences of HLA class-I were collected from the IMGT/HLA database (Robinson et al., 2003) (from the July 2011 release for HLA-A, and from the November 2011 release for HLA-B and HLA-C). The IMGT/HLA database nomenclature defines six parts of the HLA allele name. At first, we excluded all those sequences which either have 'N' or 'L' or 'Q' as a suffix at the end of the allele name for the purpose of simplicity. Secondly, from the rest of the allele set we have considered only those protein sequences that either have only one known DNA substitution within the coding region or if there is more than one DNA substitution, only the sequence with maximum length was considered. After, excluding the sequences based on these criteria we selected $1,236$ sequences of HLA-A, $1,779$

[34] A molecule's physical property that is repelled from the mass of water.

[35] Steric effects are directly related to the space that each atom occupies in a molecule and in cases where the atoms come too close then they have the effects on the associated cost in energy due to the overlapping electron clouds and potentially effecting the shape of the molecule.

[36] Polarity is the capacity of forming distinctive opposing charges from the orientation of bonds in a molecule and its spatial structure.

[37] http://mdl-qsar.software.informer.com/

of HLA-B and 929 of HLA-C. For structure-modelling purposes, a homology-modelling approach was used to model 3D structures using the Modeller software tool (Sali, 2010) (details available in Appendix A). We downloaded three known reference protein structures (i.e. HLA-A*0201 (‘1I4F’ protein data bank code) for HL-A, HLA-B*0801 (‘1AGD’) and HLA-CW*0401 (‘1IM9’) retrieved from the protein data bank (Bernsten et al., 1977). The same three reference protein structures were previously used by Doytchinova et al. (2004) for the purpose of structure modelling. Selected sequences of each gene were aligned with the corresponding known reference protein structure. A few of the aligned sequences have shown some extra amino acids either at one or at both ends since there was no match for them in the reference protein structure, so we optimized alignment by removing these segments to increase the similarity to the reference protein structure. All structures of HLA-B and HLA-C type were super-positioned on one of the structure of HLA-A based on the C-Alpha carbon atom. Side chain placement was performed using SCWRL (Bower et al., 1997; Krivov et al., 2009).

After structure modelling, the electrostatic potential (EP) was calculated in two steps: in the first step the transformation from the protein data bank (PDB) format to PQR format was performed using the software tool PDB2PQR (Dolinsky et al., 2007, 2004): this prepares structures for continuum electrostatic potential calculation by placing missing hydrogen atoms (sometimes in the modelled structures from the homology modelling process, some of the molecules have missing hydrogen atoms). In the second step, the Adaptive Poisson Boltzmann Solver (APBS) (Baker et al., 2001) is used to calculate electrostatic potentials by surrounding each protein structure with a three-dimensional grid box with 17^3 points (where the coarse grid covering the complete protein has lengths 210 angstrom ($\hat{A}$) in all three dimensions and a fine grid with 72, 32 and 52 angstrom ($\hat{A}$) in the x, y and z dimensions respectively focusing on the target area (i.e. whole area around the $\alpha 1$ and $\alpha 2$ regions)) grid points. Our interest is in analysing the top region (i.e. $\alpha 1$ and $\alpha 2$) of proteins and therefore we selected the $9 \times 17^2 = 2,601$ grid points which cover this region (see Figure 2.5). The electrostatic potential outside the van der Waals surface[38] is important for interactions with other molecules and therefore we ignored electrostatic potential at all points that were inside the van der Waals surface resulting in $2,418$ grid points (see Figure 2.5) which are outside the van der Waals surface of all the modelled structures. In a data matrix, each row indicates a single protein structure whereas each column in a row indicates a grid position (descriptor) where an electrostatic potential is

[38] A van der Waals surface is defined by the union of spherical atoms with van der Waals radius for each atom in a molecule.

calculated. In the next chapter, we analyse this dataset using state-of-the-art machine learning dimensionality reduction methods.

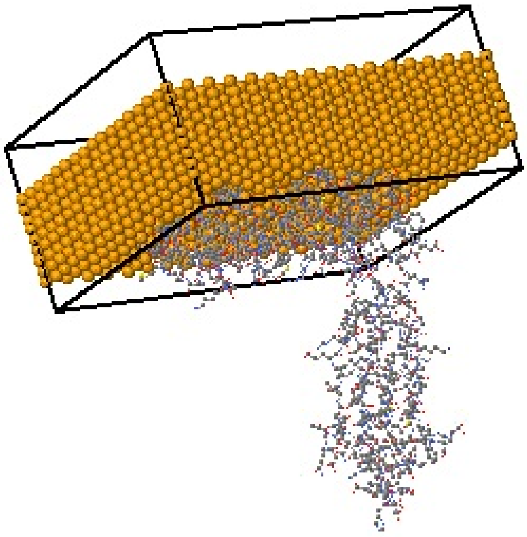


Figure 2.5: An example MHC protein three-dimensional structure with grid box to indicate region of interest for analysis (where orange dots indicates the top target surface outside the van-der Waals surface).

3 An Integrated Visual Data Mining Framework

CONTENTS

This chapter first reviews some of the existing general purpose visual data mining systems such as: the *VisuMap*, the *VisRed* and the *DVMS*. We then give theoretical details of the data projection methods that include Principal Component Analysis (PCA), Neuroscale (NSC), generative topographic mapping (GTM), GTM with simultaneous feature saliency, hierarchical GTM, latent trait model (LTM) and Gaussian process latent variable model (GPLVM). At the end of the chapter, we give theoretical details of the visualisation quality evaluation measures, we use in this thesis and these are: KL-divergence, NN classification error, trustworthiness and continuity, mean relative rank errors with respect to data and latent space and visualisation distance distortion.

3.1 Introduction

Analysing large datasets requires algorithms and techniques that are more effective than traditional statistical data summarization and management techniques which have proven to be insufficient for such complex tasks (Pal and Mitra, 2004). Traditional statistical methods fail partially because of the increase in number of objects but mostly due to the immense increase in the number of variables (Imola, 2002). The problems that arise due to high-dimensionality of data are termed the 'curse of dimensionality' (Bellman and Corporation, 1957). Such issues in the analysis of large and high-dimensional datasets have not only presented new challenges for researchers but also created new openings for theoretical developments (Donoho et al., 2000).

Nowadays, this question is becoming very important for biological experts that whether they will be able to transform tremendously increasing biological datasets into useful information with existing analytics approaches. Due to the tremendous increase in large and high-dimensional biological datasets, the need for machine learning analytics approaches have become an important area of research (Kuonen, 2003). Since, we know it is a difficult task for a human to visualise data in more than three dimensions therefore one of the effective way of representing a high-dimensional dataset is data visualisation approach which is usually considered a useful tool to explore such complex high-dimensional datasets. The term data visualisation (also known as data projection/dimensionality reduction) is used here for a mapping of a high-dimensional dataset onto a low dimensional manifold (which is usually $2D$ or $3D$) to explore intrinsic structures to help a user in understanding data better. In the last two decades, a lot of focus has been given to the machine learning approaches in order to identify more effective ways of transforming data into knowledge but still it requires a lot improvements in the existing methods and development of new analysis approaches.

Machine Learning is a field which uses a machine (i.e. computer) to construct a model from data: it usually comprises of techniques and theory of statistics, optimisation and algorithms (Kelchtermans et al., 2014). Machine learning tasks fall into three categories: *supervised* learning where a a pair of values ($\mathbf{x}_n$,$\mathbf{z}_n$) with $\mathbf{x}_n$ as inputs and $\mathbf{z}_n$ as output are involved and the goal is to model the relationship between each input vector $\mathbf{x}_n$ with the corresponding output $\mathbf{z}_n$; *reinforcement* learning is a process of learning what to do and how the situations needs to be mapped with actions in order to maximise a reward and in this method a learner is not told what actions needs to be taken rather it discovers which actions were responsible for the maximum reward, and *unsupervised* learning is the process where we might not have any specific given target but we are interested in understanding the intrinsic structure of the dataset. One of the well-known unsupervised learning tasks is data visualisation where the goal is to determine a faithful low-dimensional representation of the high-dimensional data space.

The main goal of information visualisation is to assist users with interactive visual tools (e.g. interactive scatter plots, interactive parallel coordinates etc.) which help users to use domain knowledge in exploring a dataset, gaining detailed insight, understanding the structure of the data better, and drawing useful conclusions. As argued by Maniyar and Nabney (2006b), visual representation tools are not good enough on their own to replace the analytical non-visual mining algorithms for the representation of the high-dimensional data to get a useful information. Instead it is useful to combine the approaches of different domains in order to get better understanding of these datasets. Maniyar et al. (2006) also argued that it is useful to combine approaches from the data mining, information visualisation and interactivity fields to explore large high-dimensional datasets by performing tasks such as identifying clusters, analysing data patterns appearing in different clusters, etc. Such a combined framework is known as Visual Data Mining (Keim, 2002).

According to Ankerst (2001), visual data mining techniques are classified into three groups. The first group uses visualisation techniques independent of the data mining methods. The second group uses data mining methods first and then uses visualisation methods to give a graphical view of the structure of the data. The third group provides an additional advantage to the second group by supporting interaction with the user during the mining and visualization process to improve the results. Ankerst also states that most visual data mining systems are based either on the first or on the second approach. We briefly review some of existing visual data mining systems (including the one we extended in this project) in Section 3.2. All these systems fall into the category of the third type of visual data mining systems and give strong integration of data mining and information visualization techniques used at different stages of data exploration to understand overall structure of the data where a detailed insight can be explored with the interactive features.

Traditional dimensionality reduction methods such as PCA (Pearson, 1901; Hotelling, 1933; Jolliffe, 2002), multidimensional scaling (MDS) (Cox and Cox, 1994), locally linear embedding (LLE) (Roweis and Saul, 2000), self-organizing map(SOM) (Kohonen, 1982) etc. are widely used and are reviewed by van der Maaten et al. (2009). Since the late 1990s, probabilistic dimensionality reduction methods are becoming popular and quite often outperformed traditional methods (Maniyar et al., 2006). Probabilistic dimensionality reduction methods include probabilistic PCA (Tipping and Bishop, 1999), generative topographic mapping (GTM) (Bishop and Svensen, 1998), latent trait model (LTM) (Kabán and Girolami, 2001) and Gaussian process latent variable model (GPLVM) (Lawrence, 2005).

Recently GTM has been extended to estimate feature saliency while training a visualisation model (Maniyar and Nabney, 2006a). It is also possible to compute geometric properties of the visualisation manifold, for example, we can compute local magnification factors both for GTM (Bishop et al., 1997) and LTM (Sun et al., 2001) which explain the stretch level on the visualisation space when mapped back to the data space. For the GTM, geometric properties such as local directional curvature can also be calculated for the projection manifold and they are used to monitor the amount of folding and neighbourhood preservation. Some of the results for magnification factor and local directional curvature are given in Chapter 4. In the GPLVM model, we can compute a mapping precision which explain how well a neighbourhood is preserved in the visualisation space compared to that of data space (Lawrence, 2005).

It has been argued that quite often it is difficult to understand the hidden structure of large datasets using a single two-dimensional plot. In the late 1990s, the concept of drilling down the probabilistic visualisations using a tree-like hierarchical structure was introduced by Bishop and Tipping (1998) where the basic building block was probabilistic PCA and a mixture of PPCA models was used to build the hierarchy. A similar approach was extended to GTM (Tino and Nabney, 2002) and LTM (Sun et al., 2001) like models.

The structure of the rest of the chapter is as follows: we reviewed some existing general purpose visual data mining systems in section 3.2. Section 3.3 gives a theoretical review of the data visualisation methods. At the end of the chapter, in section 3.4, we review the visualisation quality evaluation measures that we use in the rest of the thesis.

3.2 Visual Data Mining Systems

We review here three visual data mining systems each of which is based on a combination of techniques from data mining, information visualisation and interactivity support: the *VisuMap*, the *VisRed* and the *DVMS*.

3.2.1 VisuMap-A high-dimensional data visualiser

A *VisuMap*[1] is a general purpose visual data-mining system used to visualise high-dimensional data in a two-dimensional or three-dimensional space. A set of dimensionality-reduction algorithms like PCA (Pearson, 1901), Sammon Mapping (Sammon, 1969) and Curvilinear Component Analysis (CCA) (Demartines and Hérault, 1995, 1997) have been implemented in this system. The Sammon Mapping is an MDS-based[2] non-linear method for reducing the dimensionality of the data. It can use any gradient-based non-linear optimization algorithm. CCA is an improvement of Sammon's mapping by preserving more short distances by relaxing the constraints due to the long distance information. CCA uses a gradient descent optimization algorithm to minimize the stress function.

This software generates standard visualization graphs by representing data on a two- or three-dimensional scatter plot with more interactive features. This interactivity is provided with region selection on the scatter plot and other features such as zooming, data labelling, data point colour change, and data point shape change. VisuMap also supports a set of clustering algorithms on the visualization space such as K-means (MacQueen, 1967), Agglomerative clustering, Self-Organizing Map (SOM) (Kohonen, 1982), Self-Organizing Graph (SOG)(Meyer, 1998), Metric Sampling and Affinity propagation (Frey and Dueck, 2007). However, the dimensionality reduction approaches used in this tool are very basic.

3.2.2 VisRed-Visualisation by space reduction

The *VisRed* system was developed by Dourado et al. (2007): this software performs data visualisation by first reducing a high-dimensional data space to a low-dimensional space (i.e. two- or three-dimensional). Techniques such as linear/non-linear PCA and Multidimensional Scaling (MDS) are used for dimensionality reduction. Non-linear PCA was implemented using a Bottleneck Neural Network (BNN) (Kramer, 1991). The BNN is a neural network that has usually one to three neurons in the central layer covered by a symmetric architecture of hidden, input and output neurons (see Figure 3.1). Here the

[1] http://www.visumap.net/

[2] Multidimensional Scaling (MDS) is a collection of statistical approaches based on preserving inter-point distances in the projected data.

high-dimensional input data is transformed to a low-dimensional representation at the bottleneck layer and then the inverse representation is performed to re-construct the original high-dimensional data at the output layer. Classical MDS finds a distribution of points for

Figure 3.1: A non-linear PCA using auto-associative neural network architecture (adapted from (Kramer, 1991; Bellamine and Elkamel, 2008)).

a D-dimensional space in an M-dimensional space ($M << D$) where the Euclidean distance between the dissimilarity matrix of D-dimensional space and M-dimensional space can be minimized using a least squares method. MDS is an optimisation process with the purpose of minimizing a distance between the dissimilarity matrices. A set of clustering techniques on the visualisation space such as hierarchical, k-means, fuzzy k-means and SOM have also been incorporated in this software. For visualisation purpose, the software provides two or three-dimensional scatter plots.

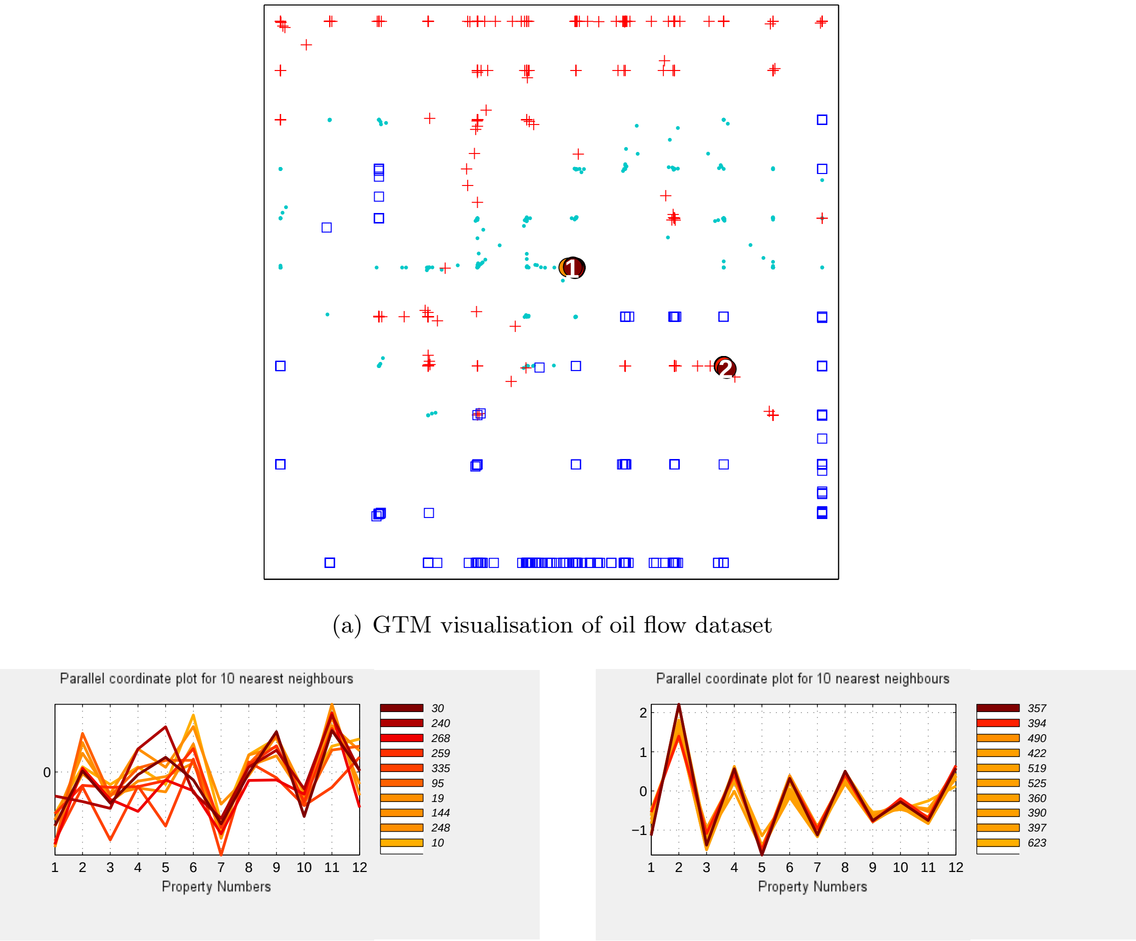
3.2.3 DVMS-Data visualisation and modelling system

A well-known framework for information visualization systems is Shneiderman's mantra (Shneiderman, 1996): it states 'Overview first, zoom and filter, then details on demand'. Based on this mantra, Maniyar et al. (2006) proposed a system called the *Data Visualisation and Modelling System (DVMS)* that uses principled projection algorithms like PCA, Neuroscale, GTM, GTM-FS, LTM and hierarchical GTM for dimensionality reduction along with information visualization techniques like scatter plots and parallel coordinates. DVMS uses the Matlab toolbox Netlab (Nabney, 2002) for the machine-learning algorithms. DVMS allows the user to perform dimensionality reduction from a high-dimensional data to a two-dimensional space. The projected data can then be visualised using scatter plots to get an overview of the structure of the data.

The system provides interactive scatter plots in which the user can select any point from the region of interest on the plot and a number of neighbouring data points around the selected point: this group of points is visualised using parallel coordinates[3] (see Figure 3.2), providing the user with a more detailed view of a data space. We recently re-designed and re-developed DVMS to improve its usability using the partially object-oriented facilities provided in Matlab, and have released this tool on the web[4]. We included the variants of the algorithms proposed in this thesis as part of DVMS, as well. We extended this system by providing additional interactive features such as highlighting of classes to observe overlapping structures, generating multiple parallel coordinate plots either based on the selection regions with nearest neighbourhood points based on the Euclidean distance or by drawing polygons to select clusters and re-training the models using the data related to selected regions, etc. The DVMS also supports visualisation of new (or test) data with the same number and type of descriptors using the previously trained visualisation model.

[3] Parallel coordinates are usually used for analysis of multivariate data by showing lines on the 2D plot where on x-axis each vertical line represents feature and on y-axis show the values of selected data rows where for each data row one colour is assigned.

[4] `http://www.aston.ac.uk/ncrg`



(a) GTM visualisation of oil flow dataset

(b) Parallel coordinate plot-1

(c) Parallel coordinate plot-2

Figure 3.2: Demonstration of parallel coordinates plots using DVMS. (a) GTM 2D visualisation of 12-dimensional 'oil flow' dataset (Bishop and James, 1993) indicate two selected regions labelled as 1 and 2 indicating region of interest to generate parallel coordinate plots shown in (b) and (c) respectively with 10 nearest neighbour data points to help understand data points of the selected regions in the dataspace.

3.3 Data Visualisation (Projection) Algorithms

This section reviews some data projection algorithms such as principal component analysis (PCA), neuroscale (NSC), generative topographic mapping (GTM), GTM with simultaneous feature saliency, hierarchical GTM (HGTM), latent trait model (LTM) and Gaussian process latent variable model (GPLVM).

3.3.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) was proposed as a linear data projection method to map a high-dimensional dataset onto a low-dimensional space (Pearson, 1901; Hotelling, 1933; Bishop, 2006). According to Hotelling (1933), PCA can be defined as an orthogonal projection of high-dimensional data to a lower-dimensional space in such a way that there will be a maximum variance in the projected data.

Consider the task of mapping a dataset of vectors $\mathbf{x}_n$, where $n = 1, 2, \ldots, N$ in a D-dimensional space, to corresponding vectors $\mathbf{z}_n$ in an M-dimensional space (usually $M = 2$ or $M = 3$ for the purpose of visualisation). Relative to standard orthonormal basis $\{\mathbf{e}_1, \ldots, \mathbf{e}_D\}$, $\mathbf{x}_n$ is represented by the vector $\{x_{n1}, \ldots, x_{nD}\}$ which is equivalent to $\mathbf{x}_n = \sum_{d=1}^{D} x_{nd} e_d$. We write the vector $\mathbf{x}_n$ as a linear combination of D orthonormal vectors $\mathbf{u}_d$

$$\mathbf{x}_n = \sum_{d=1}^{D} \alpha_{nd} \mathbf{u}_d, \tag{3.1}$$

This represents a rotation of the coordinate system to a new system defined by the $\mathbf{u}_d$. The proposed PCA is to learn the basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_D\}$ to optimise the projection according to some criteria. According to the orthonormal property,

$$\mathbf{u}_d^T \mathbf{u}_{d'} = \delta_{dd'}, \tag{3.2}$$

where $\delta_{dd'}$ is a Kronecker delta representation. Now, we can define

$$\alpha_{nd} = \mathbf{x}_n^T \mathbf{u}_d, \tag{3.3}$$

and so without loss of generality, we can write

$$\mathbf{x}_n = \sum_{d=1}^{D} (\mathbf{x}_n^T \mathbf{u}_d) \mathbf{u}_d. \tag{3.4}$$

Our goal is to map the vectors to an M-dimensional sub-space, so we choose an approximation vector $\widetilde{\mathbf{x}}_n$ and write it as

$$\widetilde{\mathbf{x}}_n = \sum_{d=1}^{M} z_{nd}\mathbf{u}_d + \sum_{d=M+1}^{D} b_d\mathbf{u}_d, \tag{3.5}$$

where the z_{nd} is dependent on the particular nth data point and b_d are taken as constants for all the data points. We are free to choose the $\mathbf{u}_d$, the z_{nd} and the b_d to minimize the error introduced by dimensionality reduction. The error measure we take is the squared distance, between the data point $\mathbf{x}_n$ and its approximation $\widetilde{\mathbf{x}}_n$ considering the average of the whole dataset, for the purpose of minimizing

$$E = \frac{1}{N}\sum_{n=1}^{N} ||\mathbf{x}_n - \tilde{\mathbf{x}}_n||^2. \tag{3.6}$$

As $\mathbf{u}_d$ are orthonormal, setting the derivative of E with respect to z_{nd} to zero gives

$$z_{nd} = \mathbf{x}_n^T\mathbf{u}_d, \tag{3.7}$$

where $d = 1, \dots, M$. Similarly, now setting the derivative of E with respect to b_d to zero gives

$$b_d = \bar{\mathbf{x}}^T\mathbf{u}_d. \tag{3.8}$$

where $d = M+1, \dots, D$. If we substitue z_{nd} and b_d into equation (3.5) and make use of the general expression of equation (3.4), we get

$$\mathbf{x}_n - \widetilde{\mathbf{x}}_n = \sum_{d=M+1}^{D} \left\{(\mathbf{x}_n - \bar{\mathbf{x}}_d)^T\mathbf{u}_d\right\}\mathbf{u}_d, \tag{3.9}$$

which explains that the displacement vector from $\mathbf{x}_n$ to $\widetilde{\mathbf{x}}_n$ lies in the space orthogonal to the principal subspace, because this is a linear combination of $\mathbf{u}_d$ for $d = M+1, \dots, D$. This is to be expected as the projected points $\widetilde{\mathbf{x}}$ must lie within the principal subspace which corresponds to be aligned with the eigenvectors corresponding to the largest eigenvalues. Now, the error function (3.6) takes the form

$$\begin{aligned} E &= \frac{1}{N}\sum_{n=1}^{N}\sum_{d=M+1}^{D} (\mathbf{x}_n^T\mathbf{u}_d - \bar{\mathbf{x}}^T\mathbf{u}_d)^2 \\ &= \frac{1}{N}\sum_{d=M+1}^{D} \mathbf{u}_d^T\boldsymbol{\Sigma}\mathbf{u}_d, \end{aligned} \tag{3.10}$$

where Σ represents the covariance matrix of the data. Applying Lagrange multipliers, it is observed that stationary points of E are present at the eigenvectors of Σ showing that $\Sigma u_d = \lambda_d u_d$. Putting such vectors into Equation (3.10), the residual error equation becomes

$$E = \frac{1}{2} \sum_{d=M+1}^{D} \lambda_d. \tag{3.11}$$

This indicates that the minimum error is obtained with the selection of the $D-M$ smallest eigenvalues whereas the data is projected onto the space spanned by the first M eigenvectors corresponding to the largest eigenvalues. These eigenvectors are known as the M principal components. PCA is simple to apply but only useful when there is a linear structure in the dataset.

3.3.2 Neuroscale (NSC)

A Neuroscale model (Lowe and Tipping, 1996) is a neural-network based data visualisation (projection) algorithm which is related to Sammon's mapping (Sammon, 1969) and Multidimensional scaling (Kruskal, 1964). For mapping a high-dimensional observation space to the projected space, it uses a radial basis function (RBF) network (see Figure 3.3, adapted from (Lowe and Tipping, 1996)). This algorithm preserves the optimal topographic structure in the transformed space and the realization of this constraint is that it attempts to make the inter-point distances in the projected space as similar as possible to the corresponding inter-point distances in the data space. A common practice is to use Euclidean distance for this purpose (for data space $d^*_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||$ and for projected space $d_{ij} = ||\mathbf{z}_i - \mathbf{z}_j||$). Neuroscale uses the following *stress metric* (similar to that of Sammon's

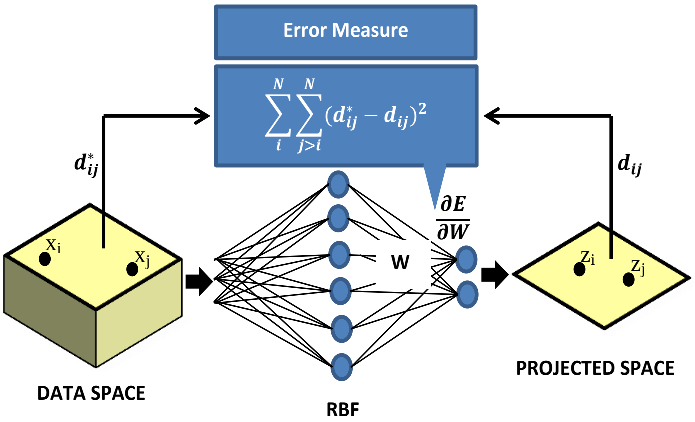

Error Measure
$\sum_{i}^{N}\sum_{j>i}^{N}(d^*_{ij} - d_{ij})^2$
d^*_{ij}
d_{ij}
$\frac{\partial E}{\partial W}$
W
X_i
X_j
Z_i
Z_j
DATA SPACE
RBF
PROJECTED SPACE

Figure 3.3: The NEUROSCALE architecture.

mapping) for minimizing the error

$$E = \sum_{i}^{N} \sum_{j>i}^{N} (d_{ij}^{*} - d_{ij})^2. \tag{3.12}$$

The RBF network for predicting a latent space point has the following form

$$\mathbf{z} = \mathbf{\Phi}(\mathbf{x})\mathbf{W}, \tag{3.13}$$

where $\mathbf{z}$ is a $1 \times M$ projected space vector, $\mathbf{\Phi}(\mathbf{x})$ is a $1 \times L$ basis functions vector and $\mathbf{W}$ is a $L \times M$ weight matrix.

The Neuroscale model needs to set the number and locations of basis function centres. To initialise the basis function centres, we first apply the Gaussian mixture model[5] with spherical covariance to the input data with the number of components equal to the number of basis functions. The GMM centres are then transferred to the centres of basis function centres (Nabney, 2002). As a common practice, the number of basis functions are taken to be close to the number of data points in the training set to represent each data point by the centre of a basis function. The Neuroscale map is learned by optimizing the RBF network parameters to minimize the stress metric defined in equation (3.12).

3.3.3 Generative Topographic Mapping (GTM)

The generative topographic mapping (GTM) was proposed by Bishop and Svensen (1998) as an alternative to the SOM which estimates a generative probability distribution. GTM is a non-linear method for mapping a low-dimension space to the high-dimensional space. The primary objective of the latent variable model is to estimate the probability distribution $p(\mathbf{x})$ that represents data $\mathbf{x} \in \mathbb{R}^D$ using latent variables $\mathbf{z} \in \mathbb{R}^M$. For a GTM model, the latent space, $\mathcal{H}$, is covered with an array of K latent space centres, $\mathbf{z}_k \in \mathcal{H}$, $k = 1, 2, \ldots, K$. A radial basis function is used as a non-linear mapping function to map a latent point $\mathbf{z}$ in the M-dimensional latent space to a corresponding point $\mathbf{x}$ in a D-dimensional data space (see Figure 3.4) as,

$$f(\mathbf{z}; \mathbf{W}) = \mathbf{\Phi}(\mathbf{z})\mathbf{W}, \tag{3.14}$$

where $\Phi(\mathbf{z})$ is the image of $\mathbf{z}$ under L basis functions and $\mathbf{W}$ is a $L \times D$ weight matrix. For a GTM the latent space is considered to be the bounded Euclidean space $[-1, 1] \times [-1, 1]$. In reality, it is impossible for the data to lie exactly on a low-dimensional manifold, and

[5] Before training a Gaussian mixture model (GMM), K-means algorithm is used to set the data density.

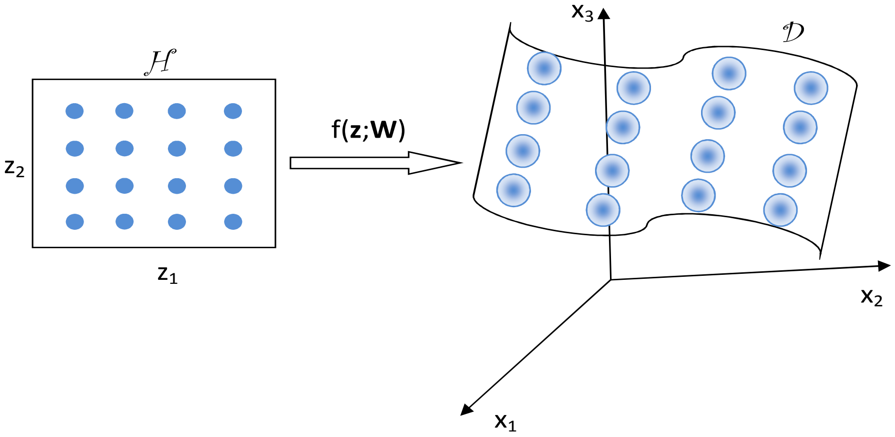


Figure 3.4: The GTM schematic representation which shows the latent space to the data space mapping using a non-linear mapping function f($\mathbf{z}$; $\mathbf{W}$).

it is therefore appropriate to consider a noise model for the $\mathbf{x}$ data vector. We consider the distribution of data vector $\mathbf{x}$ for a given latent point $\mathbf{z}$ and a weight matrix $\mathbf{W}$ as a spherical Gaussian with centre $f(\mathbf{z};\mathbf{W})$ and variance β^{-1}:

$$p(\mathbf{x}|\mathbf{z},\mathbf{W},\beta) = \left(\frac{\beta}{2\pi}\right)^{\frac{D}{2}} \exp\left(-\frac{\beta}{2}||\mathbf{x} - f(\mathbf{z};\mathbf{W})||^2\right). \tag{3.15}$$

The distribution of $\mathbf{x}$ with a given weight matrix, $\mathbf{W}$, can be computed by integrating over the distribution of latent variables, $\mathbf{z}$,

$$p(\mathbf{x}|\mathbf{W},\beta) = \int p(\mathbf{x}|\mathbf{z},\mathbf{W},\beta)p(\mathbf{z})\, d\mathbf{z}. \tag{3.16}$$

For a dataset consisting of N data points with elements $\mathbf{x}_n$ (where $n = 1,\ldots,N$), the parameter weight matrix, $\mathbf{W}$ and the inverse variance β can be determined using the log likelihood

$$\mathcal{L}(\mathbf{W},\beta) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\mathbf{W},\beta). \tag{3.17}$$

For tractability of the integral, $p(\mathbf{z})$, is taken to be a sum of delta functions: these are usually placed on the nodes of regular grid in the latent space

$$p(\mathbf{z}) = \frac{1}{K}\sum_{k=1}^{K} \delta(\mathbf{z} - \mathbf{z}_k). \tag{3.18}$$

We can now map each latent point, $\mathbf{z_k}$, to the data space using the mapping functions given in equation (3.14) each of which acts as the centre of a Gaussian density function (see Figure 3.4). Note that the model considers that all the components in a mixture share the same variance β^{-1} and the fixed mixing coefficient (i.e. $\pi_k = \frac{1}{K}$) (Svénsen, 1998). The data distribution can now be defined from equations (3.16) and (3.18),

$$p(\mathbf{x}_n|\mathbf{W},\beta) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}_n|\mathbf{z}_k,\mathbf{W},\beta). \tag{3.19}$$

The log-likelihood now takes the form

$$\mathcal{L}(\mathbf{W},\beta) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k p(\mathbf{x}_n|\mathbf{z}_k,\mathbf{W},\beta). \tag{3.20}$$

3.3.3.1 Expectation Maximization (EM) for GTM

GTM is based on a constrained mixture of Gaussians, and therefore it is easy to estimate the parameters of the model using an expectation-maximization (EM) algorithm.

In the **E-step**, we use the current set of parameters to compute the posterior probabilities (i.e. responsibilities) for each latent space component for the nth data point using Bayes' theorem,

$$\begin{aligned} r_{kn} &= p(\mathbf{z}_k|\mathbf{x}_n,\mathbf{W},\beta) \\ &= \frac{p(\mathbf{x}_n|\mathbf{z}_k,\mathbf{W},\beta)}{\sum_{k'=1}^{K} p(\mathbf{x}_n|\mathbf{z}_{k'},\mathbf{W},\beta)}. \end{aligned} \tag{3.21}$$

In the **M-Step**, we use the posterior probabilities, $\mathbf{R}$ (computed at the E-step), to re-estimate parameters of the weight matrix, $\mathbf{W}$, using the following set of linear equations (detailed derivations are available in (Bishop and Svensen, 1998)),

$$\widehat{\mathbf{W}} = (\mathbf{\Phi}^T\mathbf{E}\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{R}\mathbf{X}, \tag{3.22}$$

where $\mathbf{\Phi}$ is a $K \times L$ matrix with elements, $\phi_l(\mathbf{z}_k)$, $\mathbf{R}$ is a $K \times N$ matrix with elements, r_{kn}, $\mathbf{X}$ is an $N \times D$ data matrix and the diagonal matrix $\mathbf{E}$ contains the values

$$e_{kk} = \sum_{n=1}^{N} r_{kn}. \tag{3.23}$$

Equation (3.22) can now be used to determine the updated weight matrix, $\widehat{\mathbf{W}}$, and $\mathbf{\Phi}$ remains constant (and can be computed before the optimization starts) and the re-estimation formula for the β can now be defined as (a detailed derivation is given in (Bishop and Svensen, 1998)),

$$\frac{1}{\widehat{\beta}} = \frac{1}{ND} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{\Phi}(\mathbf{z}_k)\widehat{\mathbf{W}} - \mathbf{x}_n||^2. \tag{3.24}$$

3.3.3.2 Data visualisation using GTM

GTM gives the posterior distribution r_{kn} resulting from Bayes' theorem. It is difficult to visualise such a posterior distribution jointly for all the points in a dataset since it gives too much information (since this would require a distinct 2D plot for each data point), and it is therefore necessary to use a summary statistic, usually the mean

$$\langle \mathbf{z} | \mathbf{x}_n, \mathbf{W}, \beta \rangle = \sum_{k=1}^{K} r_{kn} \mathbf{z}_k. \tag{3.25}$$

Further details of the GTM model are available in (Bishop and Svensen, 1998).

3.3.3.3 Magnification factors (MF) and Directional curvature (DC)

Determining the geometry of the projection manifold is considered as a useful tool. The advantage of using the GTM based models is that computation of the magnification factor (MF) (Bishop et al., 1997) and directional curvature (Tino et al., 2001) is analytically possible. For a GTM projection manifold, MFs can be calculated as the determinant of the jacobian of the GTM map (Bishop et al., 1997). The magnification factors are useful in determining the stretch of the manifold in different parts of the latent space and help the user to understand the data space, outlier detection and separation of clusters. MFs can be represented in the gray colors where lighter regions indicate more stretch in the projection manifold (for example see Figure 3.5). A closed-form formula for the directional curvature

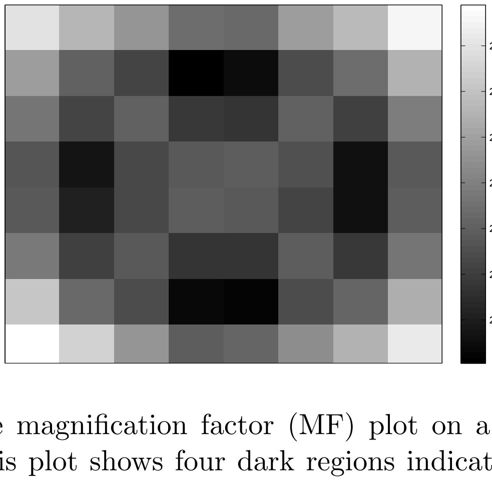


Figure 3.5: An example magnification factor (MF) plot on a $\log_{10}$ scale for the GTM visualisation model. This plot shows four dark regions indicating less stretches whereas the edges and centre of this plot are lighter to indicate more stretches.

of the GTM projection manifold for a latent space point $\mathbf{z} \in \mathcal{H}$ and a directional vector $\mathbf{h} \in \mathcal{H}$ was derived by Tino et al. (2001). Directional curvature is useful in determining the direction and amount of folding in the GTM manifold which helps the user in locating the regions where the projection manifold does not fit the data well. It is possible that a set of data points which are far apart when projected appear close together due to high-folding in the manifold. Such neighbourhood preservation can be observed with a strong curvature band on the related directional curvature plot. The direction of the folding is represented as a small line for each part of the projection manifold in the directional curvature (for example see Figure 3.6). For this example and for other curvature plots presented in this thesis, direction curvature are calculated in 16 specific directions. A maximal curvature is represented with a small line for each region. The length and shade of the background colour represents the folding magnitude. The longer line and lighter background indicates high folding (curvature).

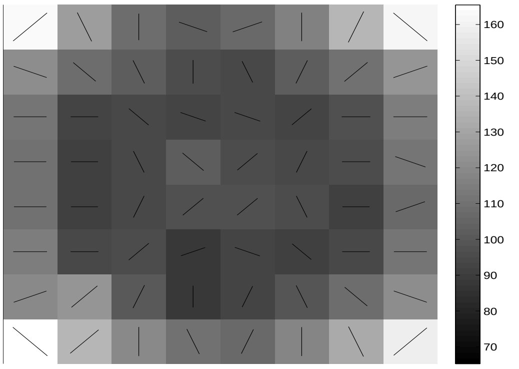


Figure 3.6: An example directional curvature (DC) plot for the GTM visualisation model where lighter regions and longer lines indicates more stretched regions.

3.3.4 Generative Topographic Mapping with simultaneous feature selection (GTM-FS)

To estimate feature saliency with GTM, it is assumed that features are conditionally independent given the mixture component label (Maniyar and Nabney, 2006a). Specifically for a mixture of Gaussians, such independence can be achieved using diagonal covariance matrices. Therefore, GTM-FS uses a mixture of diagonal Gaussians and the probability density function can be expressed as

$$p(\mathbf{x}_n|\pi, \theta) = \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} p(x_{nd}|\theta_{kd}), \tag{3.26}$$

where K represents the number of components, as in GTM, π_k is a mixing coefficient that is taken as fixed for each kth component to $\frac{1}{K}$, D represents the number of features, $\mathbf{x}_n$ represents the nth data point (in $\mathbb{R}^D$), and $p(x_{nd}|\theta_k)$ represents the probability density function of the dth feature for the kth component with the mean and variance parameters $\theta_{kd} = \{f(z_k; W), \beta_d\}$. It is also assumed that for each dimension, $d = 1, \ldots, D$, β_d is the same for all the components in the mixture. The dth feature is considered as irrelevant only if the distribution of the feature is independent of the mixture component labels and is then modelled by a common density of the form $q(x_{nd}|\lambda_d)$ which is considered as a diagonal Gaussian with λ_d parameters. We use $\Psi = (\psi_1, \ldots, \psi_D)$ to denote a set of binary values where ψ_d is equal to 1 for a relevant feature and 0 for an irrelevant feature. With these definitions, the probability density function is defined as

$$p(\mathbf{x}_n|\Delta) = \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} \left[p(x_{nd}|\theta_{kd})\right]^{\psi_d} \left[q(x_{nd}|\lambda_d)\right]^{(1-\psi_d)}, \tag{3.27}$$

where $\Delta = \{\pi_k, \theta_{kd}, \lambda_d, \psi_d\}$.

The concept of feature saliency is represented as follows.

- The ψ_ds are treated as missing variables in the EM algorithm.
- The probability of the relevant feature is represented as $\rho_d = p(\psi_d = 1)$.

Now the resultant model can be written as

$$p(\mathbf{x}_n|\Omega) = \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} \left[\rho_d p(x_{nd}|\theta_{kd}) + (1-\rho_d) q(x_{nd}|\lambda_d)\right], \tag{3.28}$$

where $\Omega = \{\pi_k, \theta_{kd}, \lambda_d, \psi_d\}$ represents the set of all parameters of the model. An intuitive way is to represent $\left[p(x_{nd}|\theta_{kd})\right]^{\psi_d} \left[q(x_{nd}|\lambda_d)\right]^{(1-\psi_d)}$ as $\left[\psi_d p(x_{nd}|\theta_{md}) + (1\psi_d) q(x_{nd}|\lambda_d)\right]$ because ψ_d is a binary indicator variable. A detailed derivation of equation (3.28) from equation (3.27) is given in Appendix C.1 (this is adapted from the derivation given in (Law et al., 2004)).

A schematic representation of the GTM-FS visualisation model is presented in Figure 3.7. This shows a three-dimensional feature set where the first two features (i.e. $\mathbf{x}_1$ and $\mathbf{x}_2$) are relevant and the third feature (i.e. $\mathbf{x}_3$) is irrelevant. Fitting a constrained mixture model with four diagonal-covariance components using equation (3.26), (represented as a two dimensional manifold, $\mathcal{H}$), there are larger variances along features $\mathbf{x}_1$ and $\mathbf{x}_2$ whereas for the feature $\mathbf{x}_3$ the variance is close to zero. The shared distribution $q(.|\lambda)$ used to represent irrelevant feature $\mathbf{x}_3$ is an ellipsoid in the centre of the data. The log-likelihood

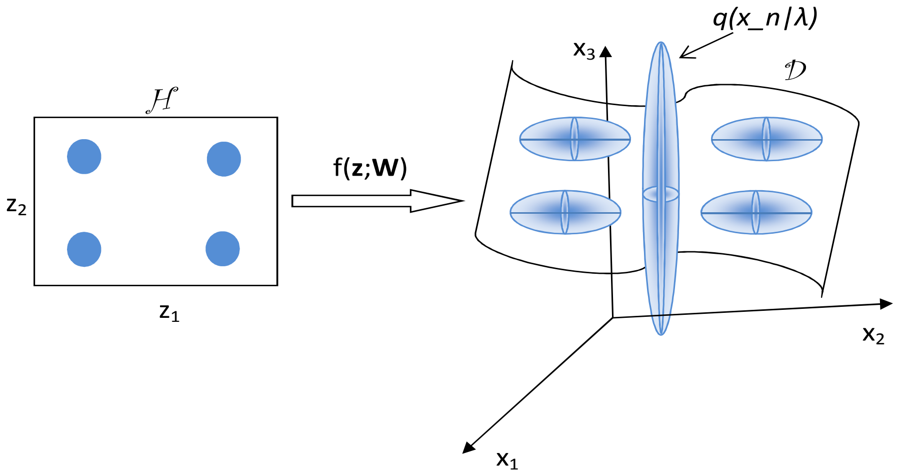


Figure 3.7: The GTM-FS schematic representation where $\mathbf{x}_1$ and $\mathbf{x}_2$ have highest saliencies and $\mathbf{x}_3$ has low saliency.

is defined as,

$$\mathcal{L}(\Omega) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\Omega), \tag{3.29}$$

where N represents the total number of data points.

3.3.4.1 Expectation Maximization (EM) for GTM-FS

Latent variable structure of GTM can also be exploited to estimate feature saliency, where parameters of the model can also be computed using the expectation-maximization (EM) algorithm. For this, we consider flipping of a biased coin where each feature has the probability of head as ρ_d; if we get the head then we consider the fact that the feature is generated from the mixture component, $p(.|\theta_{kd})$, otherwise the component, $q(.|\lambda_d)$, is responsible. A component label, y, is taken as a missing variable and then in the **E-step** using the current set of parameters, Ω, we can compute the posterior probabilities (i.e. responsibilities), $r_{nk} = p(y_n = k|\mathbf{x}_n)$, of each nth data point that it belongs to kth mixture component using Bayes' theorem,

$$r_{nk} = \frac{\prod_{d=1}^{D} \rho_d p(x_{nd}|\theta_{kd}) + (1-\rho_d) q(x_{nd}|\lambda_d)}{\sum_{k=1}^{K} \prod_{d=1}^{D} \rho_d p(x_{nd}|\theta_{kd}) + (1-\rho_d) q(x_{nd}|\lambda_d)}. \tag{3.30}$$

The responsibility matrix, $\mathbf{R}$, is used to compute $u_{nkd} = p(\psi_d = 1, \mathbf{y}_n = k|\mathbf{x}_n)$ which explains how relevant the nth data point is for relating to the kth component when the dth feature is considered and $v_{nkd} = p(\psi_d = 0, y_n = k|\mathbf{x}_n)$ shows the irrelevance (noise) of the nth data point relating to the kth component when the dth feature is considered and these measures can be computed as follows

$$u_{nkd} = \frac{\rho_d p(x_{nd}|\theta_{kd})}{\rho_d p(x_{nd}|\theta_{kd}) + (1-\rho_d) q(x_{nd}|\lambda_d)} r_{nk}, \tag{3.31}$$

$$v_{nkd} = r_{nk} - u_{nkd}. \tag{3.32}$$

In the **M-step**, we can use $\mathbf{U}$ to re-estimate the weight matrix $\mathbf{W}$ following a set of linear equations for each dth feature,

$$\widehat{\mathbf{w}}_d = (\mathbf{\Phi}^T \mathbf{E}_d \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{U}_d \mathbf{x}_d, \tag{3.33}$$

where $\mathbf{\Phi}$ is a $K \times L$ matrix, $\mathbf{U}_d$ is a $K \times N$ matrix computed using equation (3.31), $\mathbf{x}_d$ is a $N \times 1$ data vector and a diagonal matrix $\mathbf{E}_d$ can take the values

$$e_{kkd} = \sum_{n=1}^{N} u_{nkd}. \tag{3.34}$$

Now, we can straightforwardly re-estimate parameters of the mixture model using the re-estimated weight matrix, $\widehat{\mathbf{W}}$: then first we re-estimate the centres of mixture model in the data space (see equation (3.35)) and then we use these re-estimated centres to update the diagonal Gaussian width in each direction (for each dth feature) (see equation (3.36))

$$\widehat{\text{Mean}\,\theta_k} = \widehat{\mathbf{m}_k} = \mathbf{\Phi}(\mathbf{z}_k)\widehat{\mathbf{W}}, \tag{3.35}$$

where $\widehat{\mathbf{m}_k}$ is a $1 \times D$ vector.

$$\frac{1}{\widehat{\beta_d}} = \frac{\sum_k \sum_n u_{nkd}(x_{nd} - \widehat{m_{kd}})^2}{\sum_k \sum_n u_{nkd}}. \tag{3.36}$$

Common density parameters, λ_d, can be updated as follows,

$$\widehat{\text{Mean}\,\lambda_d} = \frac{\sum_n (\sum_k v_{nkd}) x_{nd}}{\sum_{nk} v_{nkd}} \tag{3.37}$$

$$\widehat{\text{Var}\,\lambda_d} = \frac{\sum_n (\sum_k v_{nkd} (x_{nd} - \widehat{Mean\lambda_d})^2}{\sum_{nk} v_{nkd}} \tag{3.38}$$

The feature saliency parameters for the continuous features set can be updated using

$$\widehat{\rho}_d = \frac{\sum_{nk} u_{nkd}}{\sum_{nk} u_{nkd} + \sum_{nk} v_{nkd}}. \tag{3.39}$$

If we take the Dirichlet-type, but improper, prior for the feature saliencies (the same is used by Law et al. (2004) for clustering data with simultaneous feature saliency),

$$p(\rho_1, \cdots, \rho_D) \propto \prod_{d=1}^{D} \rho_d^{-\frac{KP}{2}} (1-\rho_d)^{-\frac{T}{2}}. \tag{3.40}$$

then the feature saliency measure can be updated by

$$\widehat{\rho}_d = \frac{\max(\sum_{nk} u_{nkd} - \frac{KP}{2}, 0)}{\max(\sum_{nk} u_{nkd} - \frac{KP}{2}, 0) + \max(\sum_{nk} v_{nmd} - \frac{T}{2}, 0)}, \tag{3.41}$$

where P and T are the number of parameters in θ_{kd} and λ_d respectively.

3.3.4.2 Computational considerations for GTM and GTM-FS models

Considering the process required to update the parameters (for example winning nodes or responsibilities), we observe that the distance calculation between data points and mixture of reference vectors (used while calculating $p(\mathbf{x}_n|\Omega)$) is the same both for GTM and GTM-FS models. While updating parameters both GTM (in equation (3.22)) and GTM-FS (in equation (3.33)) requires a matrix inversion of an $L \times L$ matrix, where L indicates the number of basis functions, followed by a matrix multiplication. The matrix inversion scales as $\mathcal{O}(L^3)$, whereas the matrix multiplication scales as $\mathcal{O}(KND)$[6], where K indicates the number of latent space grid points. GTM-FS model requires to process an extra loop for D features to update weight vector $\widehat{\mathbf{w}}_d$ in the parameter learning process.

3.3.5 Hierarchical Generative Topographic Mapping (HGTM)

The hierarchical GTM (HGTM) is a tree-like structure, $\mathcal{T}$, which is composed of GTMs and their two-dimensional visualisation plots (for example see schematic representation in Figure 3.8) (Tino and Nabney, 2002). The first node in the tree at level-1 is called the *Root* node. A node $\mathcal{N}$ at $Level(\mathcal{N}) = l$ has children at level $l+1$ (i.e. $Level(\mathcal{M}) = l+1$ for all $\mathcal{M} \in Children(\mathcal{N})$). Except the *Root* model in the hierarchy, each model $\mathcal{M}$ has the parent-conditional mixture coefficient prior $\pi(\mathcal{M}|Parent(\mathcal{M}))$. The priors are non-negative and also fulfil the consistency condition (i.e. $\sum_{\mathcal{M} \in Children(\mathcal{N})} \pi(\mathcal{M}|\mathcal{N}) = 1$). The unconditional prior for the root level is taken to be $\pi(Root) = 1$ whereas for all other

[6] To be exact, such a matrix multication scales as $\mathcal{O}(KLD + KND)$, where usually the number of basis functions L are less than that of number of data points N.

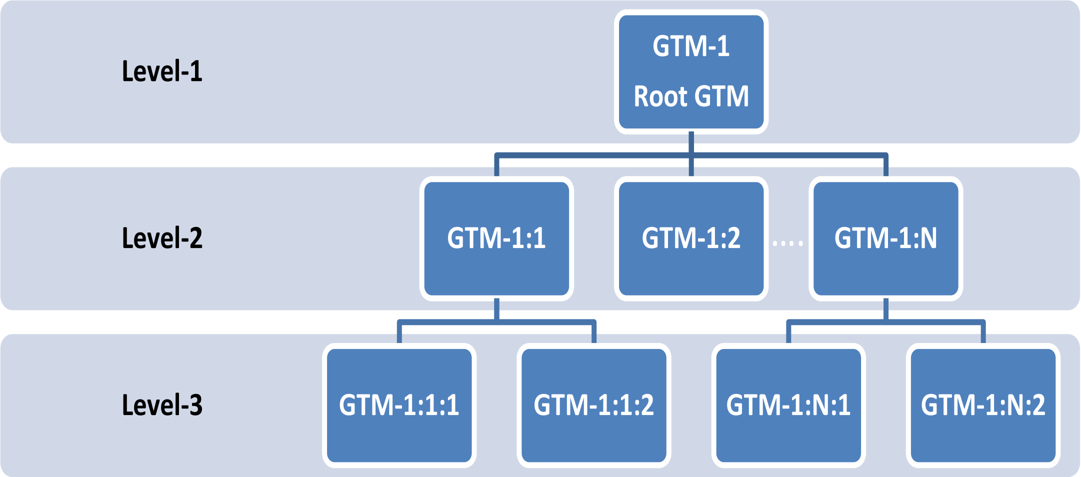


Figure 3.8: The HGTM schematic representation.

models they are computed recursively as,

$$\pi(\mathcal{M}) = \prod_{i=2}^{Level(\mathcal{M})} \pi(Path(\mathcal{M})_i | Path(\mathcal{M})_{i-1}), \tag{3.42}$$

where $Path(\mathcal{M}) = (Root, \cdots, \mathcal{M})$ contains the nodes on the path from the $Root$ to the node $\mathcal{M}$ in the tree, $\mathcal{T}$. The distribution of the data vector $\mathbf{x}$ given a tree can now be represented as,

$$p(\mathbf{x}|\mathcal{T}) = \sum_{\mathcal{M} \in Leaves(\mathcal{T})} \pi(\mathcal{M}) p(\mathbf{x}|\mathcal{M}). \tag{3.43}$$

This gives a *soft* assignment of the input space to the leaf models of the HGTM. The model is trained using a variant of the EM algorithm to maximise the likelihood with respect to the given data $\mathbf{X}$. The hierarchy of GTMs is trained in a recursive way where we use interactivity to select regions of interest on the visualisation at any level plot. Details of HGTM are given in (Tino and Nabney, 2002).

3.3.6 Latent Trait Model (LTM)

A generalisation of GTM was proposed by Kabán and Girolami (2001) to model different types of data under a unified generative latent variable formalism, by considering non-Gaussian distributions from the exponential family for modelling noise. Their main focus was to visualise discrete data in a continuous latent visualisation space and they called this model the latent trait model (LTM).

The functional form of the exponential family of distributions can be defined by

$$p_{\mathcal{G}}(\mathbf{x}|\theta) = \exp\left\{\mathbf{x}\theta - \mathcal{G}\left(\theta\right)\right\} p_0(\mathbf{x}). \tag{3.44}$$

In our case, the conditional exponential family of distribution for a data point $\mathbf{x}_n$ given latent point $\mathbf{z}_n$ and a weight matrix $\mathbf{W}$ can be defined as,

$$p_{\mathcal{G}}(\mathbf{x}_n|\mathbf{z}_k, \mathbf{W}) = \exp\Big\{\mathbf{x}_n f(\mathbf{z}_k, \mathbf{W}) - \mathcal{G}\left(f(\mathbf{z}_k, \mathbf{W})\right)\Big\} p_0(\mathbf{x}_n), \tag{3.45}$$

where $\mathcal{G}(.)$ is the cumulant function and is defined as

$$\mathcal{G}\left(f(\mathbf{z}_k, \mathbf{W})\right) = \log\left(\int \exp(\mathbf{x} f(\mathbf{z}_k, \mathbf{W})) p_0(\mathbf{x})\, d\mathbf{x}\right). \tag{3.46}$$

The *natural parameter* θ of the exponential family of the distribution is taken to be a linear mixing of the latent vectors with respect to the weight matrix $\mathbf{W}$

$$\theta_k = f(\mathbf{z}_k; \mathbf{W}) = \mathbf{\Phi}(\mathbf{z}_k)\mathbf{W}, \tag{3.47}$$

where $\mathbf{W}$ is the weight matrix of the trait model. The gradient of the cumulant function with respect to the natural parameter (i.e. $f(\mathbf{z}_k; \mathbf{W})$) is

$$\mathbf{m}_k = g(f(\mathbf{z}; \mathbf{W})) = \nabla_{f(\mathbf{z}_k;\mathbf{W})}\mathcal{G}(f(\mathbf{z}_k; \mathbf{W})), \tag{3.48}$$

where ∇ represents the gradient operation and the function $g(.)$ is the link function (Kabán and Girolami, 2001). Like GTM, the distribution in the latent space is modelled as a regular grid of latent points with elements $\mathbf{z}_k$ (where $k = 1, \cdots, K$) and the prior for this latent space can be taken from the equation (3.18) (i.e. $\frac{1}{K}$) for each of the latent component. The log-likelihood for the density of mixture under the exponential family can be defined as,

$$\begin{aligned}\mathcal{L} &= \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} p_{\mathcal{G}}(\mathbf{x}_n|\mathbf{z}_k, \mathbf{W}) p(\mathbf{z_k})\right) \\ &= \sum_{n=1}^{N} \log\left(\pi_k \sum_{k=1}^{K} p_{\mathcal{G}}(\mathbf{x}_n|\mathbf{z}_k, \mathbf{W}))\right).\end{aligned} \tag{3.49}$$

3.3.6.1 An EM algorithm for LTM

Kabán and Girolami (2001) derived a general EM algorithm for the exponential family in the latent variable formalism of GTM-like models. In the **E-step**, posterior probabilities (i.e. responsibilities) can be computed using the current set of parameters using

$$r_{kn} = p_{\mathcal{G}}(\mathbf{z}_k|\mathbf{x}_n, \mathbf{W}) = \frac{\pi_k p_{\mathcal{G}}(\mathbf{x}_n|\mathbf{z}_k, \mathbf{W})}{\sum_{k'}^{K} \pi_{k'} p_{\mathcal{G}}(\mathbf{x}_n|\mathbf{z}_{k'}, \mathbf{W})}. \tag{3.50}$$

In relation with EM algorithm, Kabán and Girolami (2001) used maximisation of the relative likelihood instead of maximizing the log-likelihood, which does not contain the log of sum. The relative log likelihood between old and new set of parameters can be calculated as,

$$\begin{aligned} Q &= \sum_{n=1}^{N}\sum_{k=1}^{K} r_{kn} \log\left\{p_{\mathcal{G}}(\mathbf{x}_n|\mathbf{z}_k, \mathbf{W}) p(\mathbf{z}_k)\right\} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} r_{kn} \Big\{\mathbf{x}_n f(\mathbf{z}_k, \mathbf{W}) - \mathcal{G}\left(f(\mathbf{z}_k, \mathbf{W})\right) + \log(p_0(\mathbf{x}_n)) + \log(p(\mathbf{z}_k))\Big\} \end{aligned} \tag{3.51}$$

In the **M-step** it is now straightforward to maximize the function Q with respect to $\mathbf{W}$,

$$\frac{\partial Q}{\partial \mathbf{W}} = \mathbf{\Phi}^T \left[\mathbf{R}\mathbf{X} - \mathbf{E}g(\mathbf{\Phi}\mathbf{W})\right], \tag{3.52}$$

where $\mathbf{\Phi}$ is a $K \times L$ matrix, $\mathbf{R}$ is a $K \times N$ matrix calculated using equation (3.50), $\mathbf{X}$ is an $N \times D$ data matrix and the diagonal matrix $\mathbf{E}$ contains the values

$$e_{kk} = \sum_{n=1}^{N} r_{kn}. \tag{3.53}$$

In case of an isotropic Gaussian with unit variance, the matching function $g(.)$ becomes the identity and setting the derivative equal to zero, we obtain the closed form M-step of standard GTM (i.e. equation (3.22)) (Bishop and Svensen, 1998) model as,

$$\widehat{\mathbf{W}} = (\mathbf{\Phi}^T\mathbf{E}\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{R}\mathbf{X}, \tag{3.54}$$

In general a non-linear optimization approach may be required (e.g. iterative least-square methods can be employed for this purpose). However, a Generalised EM (GEM) (McLachlan and Krishnan, 1997) algorithm is a more appropriate choice because of the fact that the convergence to the local maximum is guaranteed on increasing without maximizing the relative likelihood (Kabán and Girolami, 2001). A simple gradient-base update can be obtained for $\mathbf{W}$ from the equation (3.52) as,

$$\Delta\mathbf{W} \propto \mathbf{\Phi}^T \left[\mathbf{R}\mathbf{X} - \mathbf{E}g(\mathbf{\Phi}\mathbf{W})\right], \tag{3.55}$$

where this can be used as an inner loop in the M-step and the weight matrix $\widehat{\mathbf{W}}$ update involves matrix multiplication which scales as $\mathcal{O}(LND + LK(N + D + K))$. In this thesis, we employed a gradient inner loop M-step and the correlations between dimensions of $\boldsymbol{\phi}_l$ responsible for preserving the neighbourhood are required for a topographic organization. We explain here how the natural parameter θ modifies under the gradient update of the weight matrix $\mathbf{W}$ as,

$$\theta_k = \boldsymbol{\phi}_k \widehat{\mathbf{W}} = \boldsymbol{\phi}_k \mathbf{W} + \eta \sum_{n=1}^{N} \sum_{k'=1}^{K} r_{k'n} \boldsymbol{\phi}_k \boldsymbol{\phi}_{k'}^T (\mathbf{x} - \mathbf{m}_{k'}). \tag{3.56}$$

This is considered analogous to the Self Organizing Map (SOM) (Kohonen, 1995) update. The neighbourhood relationship width maintained by $\mathbf{\Phi}\mathbf{\Phi}^T$ is also controlled by the responsibility matrix, r_{kn}, as previously discussed in (Kabán and Girolami, 2001; Bishop and Svensen, 1998).

For example the Bernoulli density is

$$p(x|m) = m^x (1-m)^{1-x}. \tag{3.57}$$

To convert equation (3.57) to the general exponential form (equation (3.44)) we re-write it as

$$\begin{aligned} p(x|m) &= \exp\left\{ \log\left(m^x (1-m)^{1-x} \right) \right\} \\ &= \exp\left\{ x \log m + (1-x) \log(1-m) \right\} \\ &= \exp\left\{ x \log \frac{m}{1-m} + \log(1-m) \right\}, \end{aligned} \tag{3.58}$$

where $\theta = \log \frac{m}{1+m}$, $G(\theta) = -\log(1-m)$, and $p_0(x) = 1$.

3.3.7 Gaussian Process Latent Variable Model (GPLVM)

The Gaussian process latent variable model (GPLVM) is a non-linear extension of probabilistic PCA and uses a smooth mapping from the latent space to the data space. In the GPLVM instead of optimizing weights they are marginalized out and instead of marginalizing over the latent space it is optimized (i.e. the position of each point in the latent space is optimized). A conjugate prior over the weights is chosen, taking the form of a spherical Gaussian distribution for each dimension

$$p(\mathbf{W}) = \prod_{i=1}^{D} N(\mathbf{w}_i | 0, I), \tag{3.59}$$

where $\mathbf{w}_i$ is the ith row vector of the weight matrix $\mathbf{W}$ and the likelihood after marginalizing the weights is

$$p(\mathbf{X}|\mathbf{Z},\beta) = \prod_{d=1}^{D} p(\mathbf{x}_{(:,d)}|\mathbf{Z},\beta), \tag{3.60}$$

where $p(\mathbf{x}_{(:,d)}|\mathbf{Z},\beta) = N(\mathbf{x}_{(:,d)}|0, \mathbf{Z}\mathbf{Z}^T + \beta^{-1}I)$ represents a distribution over a single feature in the data space. GPLVM uses the following log likelihood function to optimize the latent variables (similar to the likelihood used in (Tipping and Bishop, 1999))

$$L = -\frac{DN}{2}\log(2\pi) - \frac{D}{2}\log(\det K) - \frac{1}{2}\operatorname{tr}(K^{-1}\mathbf{X}\mathbf{X}^T). \tag{3.61}$$

If $K = \mathbf{Z}\mathbf{Z}^T + \beta^{-1}I$ is a linear kernel, then it is similar to PPCA. But for the GPLVM a non-linear RBF kernel is used

$$\begin{aligned} k(z_i, z_j) = \theta_{rbf} \exp -(\frac{\gamma}{2}(z_i - z_j)^T((z_i - z_j))) \\ + \theta_{bias} + \theta_{white}\delta_{ij}, \end{aligned} \tag{3.62}$$

as explained in Lawrence (2005, 2004). Then the optimization of the latent points can be achieved by first taking the gradient of the log likelihood with respect to the kernel

$$\frac{\partial L}{\partial K} = K^{-1}\mathbf{X}\mathbf{X}^T K^{-1} - DK^{-1} \tag{3.63}$$

and then combining this with $\frac{\partial K}{\partial z_{n,j}}$ using the chain rule. The gradient calculation uses the inverse of the kernel matrix (see equation (3.63)); it has $\mathcal{O}(N^3)$ complexity thereby making it less practical for large datasets. Due to this computational complexity, a GPLVM is usually trained using sparse approximations where a small subset of data points of size $k << N$ known as 'inducing points' or the 'active set' is used to reduce the complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(k^2N)$. The number of data points in the active set is set by the user. In earlier work with GPLVM the information vector machine (IVM) (where data points are chosen sequentially based on the reduction of the posterior process‘s entropy (Lawrence et al., 2003)) was used. Further algorithmic details can be found in (Lawrence, 2005, 2004).

Improvements in approximations for Gaussian process regression based on the unified view proposed by Quionero-Candela et al. (2005) suggested new methods for their use with GPLVM which improved the results compared to IVM-based approximation. Further details of these new approximations applied to the GPLVM can be found in (Lawrence, 2008).

3.3.7.1 Dissimilarity and local-distance preservation with GPLVM

The standard GPLVM uses a smooth[7] mapping from the latent space to the data space. Hence, this mapping does not constrain points which are close in the data space to be close in the latent space: rather, it ensures that the points which are distant in the data space cannot be mapped from points which are close in the latent space. Otherwise there would be a discontinuity in the mapping. Hence, the standard GPLVM can be considered as a dissimilarity-preserving approach. It is also unfortunate that for dimensionality reduction, often it is difficult to accurately maintain both local distances and dissimilarities. When users visualise data, it is the local structure that is most relevant to their analysis (for example, when they identify clusters).

Therefore, we use a variant of GPLVM where a constrained smooth mapping is employed to overcome the problem of local distance preservation because the data points $\mathbf{z}$ are no longer freely optimized (as in equation (3.60)). Instead they are the image of points $\mathbf{x}$ in the data space under the non-linear function like a Radial Basis function (RBF) kernel or multi-layer perceptron (MLP) (i.e. $z_{nj} = f_j(\mathbf{x}_n; \mathbf{w})$). This constrained mapping (also known as a back-constraint) ensures that the data points which are close in the visualisation space that are also close in the data space. We used an MLP kernel as a back-constraint in our experiments.

Other approaches such as Kernel PCA and Neuroscale (Lowe and Tipping, 1996) also perform a smooth mapping from the data space to the latent space (i.e. the reverse direction to that of GPLVM). Further details on preserving local distances with GPLVM can be found in (Lawrence, 2006).

3.4 Visualisation Quality Evaluation Methods

Dimensionality reduction methods are usually studied in the context of unsupervised learning so evaluating the quality of visualisation is not an easy task. However, for some datasets, class labels are given for data points and typically we are interested to show better separation between those classes on the visualisation plots. The class information is not used while training a visualisation model though it is only used for better presentation (i.e. with colours or marker style or both). From the visualisations with such a colour plots, we can only observe the effectiveness of a projection however it is hard to compare visualisations resulting from different visualisation approaches. In this thesis,

[7] The term smooth is used here to explain that the points which are close in the latent space to the points which are also close in the data space. We can employ many different covariance functions to use with Gaussian process for this purpose.

we exploit the use of two such measures to work in the presence of known labels such as the Kullback-Leibler divergence which we use to determine the separation between classes and nearest-neighbour-classification error. We also use some other visualisation quality evaluation measures which do not require the labels. These can be divided into two categories: the first is the distance distortion measure and the second is the class of rank-based measures which includes trustworthiness, continuity, and mean relative rank errors with respect to data space and/or latent space. Each of these is explained in the following sub-sections.

3.4.1 Kullback-Leibler (KL) divergence

It is useful to analytically measure the separation between different classes in the projected space. For obtaining such an analytic measure, we first apply a Gaussian mixture model (GMM) (Bishop, 1995a) to each class on the projected space and then calculate the Kullback-Leibler (KL) divergence (Cover and Thomas, 1991) between the fitted GMMs using

$$D_{KL}(p_a||p_b) = \sum_x p_a(x) \log \frac{p_a}{p_b}, \tag{3.64}$$

where p_a and p_b are GMMs of the classes a and b respectively. KL-divergence is an asymmetric measure, therefore we calculate D_{KL} for each class pair in both directions and sum up all the values to get a symmetric measure S_{KL} using

$$S_{KL} = \sum_{a=1}^{C} \sum_{b=1}^{C} D_{KL}(p_a||p_b) \tag{3.65}$$

where C indicates the number of classes. Higher values of the KL divergence sum indicate better separation between the classes.

3.4.2 Nearest Neighbour (NN) classification error

Another measure of the quality of a visualisation is the quality of a classifier trained on the projected data. A simple form of this can be achieved if we classify each data point according to its class with the class of nearest neighbour on the visualisation results of the data projection algorithm.

3.4.3 Trustworthiness and Continuity

In the information visualisation domain, two well-known visualisation quality measures based on comparing neighbourhoods in the data space X and projection space $\mathbf{Z}$ are *trustworthiness* and *continuity* (Venna and Kaski, 2001). Trustworthiness measures the fraction of data points distant in the data space that become neighbours in the projection space and continuity measures the fraction of neighbouring data points in the data space that become distant in the projection space. Both for trustworthiness and continuity, there is no way to select automatically the number of neighbourhood points (i.e. the value of k) to be used. Therefore, later in this thesis, we use range of neighbourhood sizes to compute these measures and then take the average over the corresponding values of k.

Suppose that $R^{\mathbf{X}}_{i,j}$ is the rank of the jth data point from the corresponding ith data point with respect to the distance measure in the high-dimensional data space $\mathbf{X}$, and $U_k(i)$ represents the data points in the k-nearest neighbourhood of the ith data point in the latent space $\mathbf{Z}$ but *not* in the data space $\mathbf{X}$. Trustworthiness with k neighbours can be calculated as

$$1 - \frac{2}{\gamma_k} \sum_{i=1}^{N} \sum_{j \in U_k(i)} (R^{X}_{i,j} - k). \tag{3.66}$$

For measuring the continuity, we define $R^{\mathbf{Z}}_{i,j}$ to be the rank of the jth data point from the ith data point with respect to the distance measure in the visualisation space $\mathbf{Z}$ and $V_k(i)$ to be the set of data points in the k-nearest neighbourhood of the ith data point in the data space $\mathbf{X}$ but *not* in the visualisation space $\mathbf{Z}$. The continuity with k neighbours can be calculated as

$$1 - \frac{2}{\gamma_k} \sum_{i=1}^{N} \sum_{j \in V_k(i)} (R^{Z}_{i,j} - k). \tag{3.67}$$

Both for trustworthiness and continuity, we take the normalising factor (γ_k) as

$$\gamma_k = \begin{cases} Nk(2N - 3k - 1) & \text{if } k < N/2, \\ N(N-k)(N-k-1) & \text{if } k \geqslant N/2, \end{cases} \tag{3.68}$$

where the γ_k ensures that the value of trustworthiness and continuity lie between 0 and 1. The higher the measure the better the visualisation as this implies that local neighbourhoods are better preserved by the projection.

3.4.4 Mean Relative Rank Errors

Two more quality measures which work on the same principle as trustworthiness and continuity are mean relative rank errors (MRREs) with respect to data space and latent space respectively (Lee and Verleysen, 2008). Using the same notation, the MRREs are defined with respect to data space as

$$MRRE^X(k) = \frac{1}{\tau_k}\sum_{i=1}^{N}\sum_{j\in N_k^X(i)}\frac{|(R_{i,j}^Z - R_{i,j}^X)|}{R_{i,j}^X}, \tag{3.69}$$

and with respect to latent space as

$$MRRE^Z(k) = \frac{1}{\tau_k}\sum_{i=1}^{N}\sum_{j\in N_k^Z(i)}\frac{|(R_{i,j}^X - R_{i,j}^Z)|}{R_{i,j}^Z}. \tag{3.70}$$

where the the normalisation factor τ_k for both types of MRREs is $\tau_k = N\sum_{k'=1}^{k}\frac{|N-2k'|}{k'}$. However, in this case, the lower the MRRE is, the better the projection quality is.

3.4.5 Visualisation Distance Distortion

The visualisation distance distortion (VDD) measure is used to compare the distances between the points in the data space X and the projection space Z between each data point and its k nearest neighbours. Functions such as $Dist^{\mathbf{x}}$ and $Dist^{\mathbf{z}}$ are taken as distance functions for computing distance between any pair of elements between the data space $\mathbf{X}$ and the projection space $\mathbf{Z}$. For a given point $\mathbf{x} \in \mathbf{X}$, a number k is considered as $1 < k < N$ of nearest neighbours. Let $i_{\mathbf{x},0}, \cdots, i_{\mathbf{x},k}$ represent the sorted nearest neighbours' list where the first element index $i_{\mathbf{x},0}$ is the index of the data point $\mathbf{x}$. Now the vector of nearest neighbours in data space $\mathbf{X}$ can be considered as,

$$Dist_{\mathbf{x},k}^{\mathbf{X}} = \langle Dist^X(\mathbf{x}, \mathbf{X}[i_{\mathbf{x},1}]), \cdots, Dist^X(\mathbf{x}, \mathbf{X}[i_{\mathbf{x},k}])\rangle. \tag{3.71}$$

If we take the point $\mathbf{z} \in \mathbf{Z}$ as the projection of the point $\mathbf{x} \in X$ then we can define a vector comprised of distances between a data point $\mathbf{z}$ and the projection of k nearest neighbours in the data space X as

$$Dist_{\mathbf{x},k}^{\mathbf{Z}} = \langle Dist^Z(\mathbf{z}, \mathbf{Z}[i_{\mathbf{x},1}]), \cdots, Dist^Z(\mathbf{z}, \mathbf{Z}[i_{\mathbf{x},k}])\rangle. \tag{3.72}$$

Using these distance vectors, the VDD measure for the $\mathbf{x}$th data point with k nearest neighbours can be calculated as

$$VDD(\mathbf{x}, k) = \left\| \frac{Dist_{\mathbf{x},k}^{\mathbf{X}}}{\left\|Dist_{\mathbf{x},k}^{\mathbf{X}}\right\|} - \frac{Dist_{\mathbf{x},k}^{\mathbf{Z}}}{\left\|Dist_{\mathbf{x},k}^{\mathbf{Z}}\right\|} \right\|. \tag{3.73}$$

This visualisation distance distortion measure is also known as a projection precision score (PPS) (Schreck et al., 2010). VDD is calculated as the norm of the difference vectors between the scaled distances in the data space and the visualisation latent space. The scaled distances are used to make the distance comparable between the data space and the latent visualisation space. The average visualisation distance distortion ($AVDD$) factor can be calculated using

$$AVDD(\mathbf{X}, k) = \frac{1}{N}\sum_{n=1}^{N} VDD(\mathbf{x}_n, k), \tag{3.74}$$

where N is the number of data points. The lower the $AVDD$ value the better the proximity is preserved.

3.5 Summary

In this chapter we first reviewed some general purpose visual data mining systems. We then explained briefly the software engineering work that we caried out on the Data Visualisation and Modelling System (DVMS) in order to make it easier to integrate new data projection algorithms. We then reviewed the projection algorithms that are supported by this system: principal component analysis (PCA), Neuroscale (NSC), generative topographic mapping (GTM) and its variants such as GTM with simultaneous feature saliency, hierarchical GTM, latent trait model (generalisation of GTM that was developed for discrete data) and the Gaussian process latent variable model (GPLVM). In the remaining chapters of the thesis we will exploit and extend some of these appraoches for analysing complex datasets. Data projection methods are considered as unsupervised learning methods and therefore measuring and comparing their performance is difficult. We also reviewed here some of the methods that we used in the rest of the thesis to evaluate the visualisation quality.

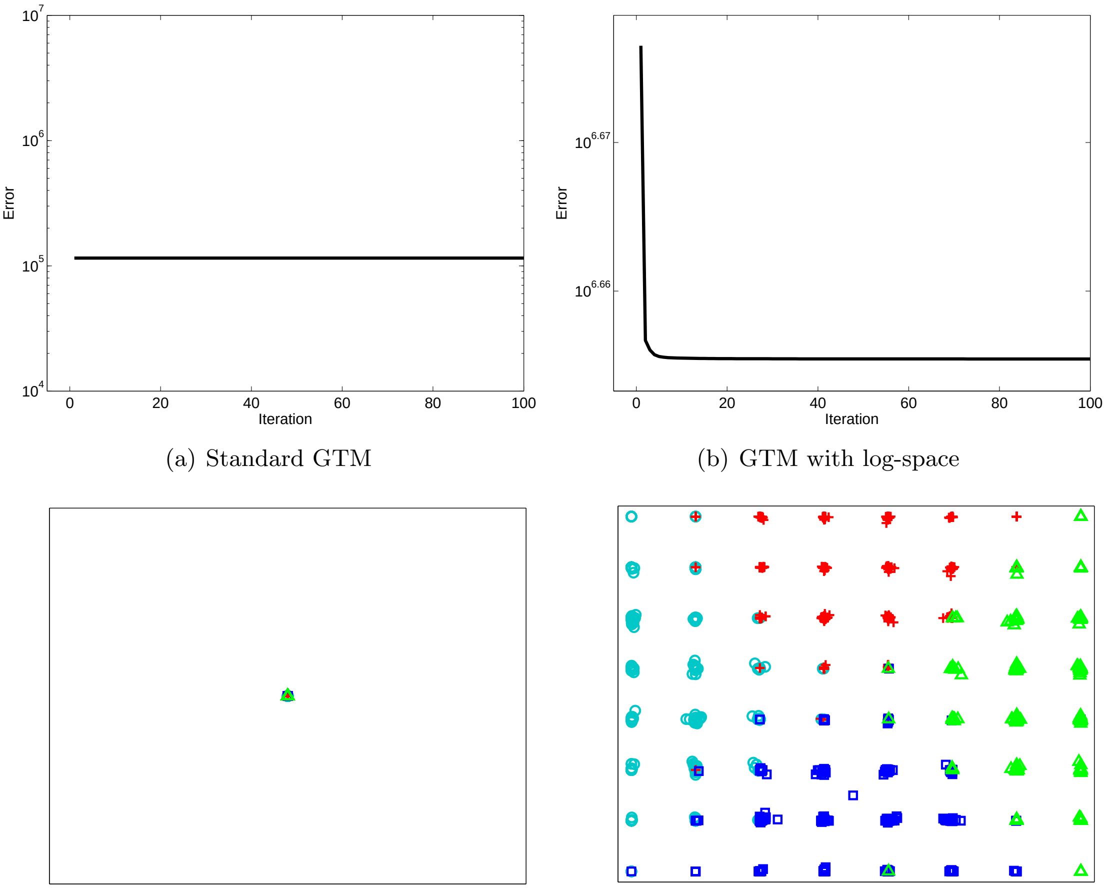
4 Effective Visualisation for High-Dimensional Datasets

CONTENTS

In this chapter, we propose variants of non-linear data visualisation methods (Generative Topographic Mapping (GTM), Hierarchical GTM (HGTM) and GTM with simultaneous feature saliency (GTM-FS)) that are adapted to be effective on very high-dimensional data. The adaptations use log-space values to avoid numerical problems that are observed usually at certain steps of the *expectation-maximization* (EM) algorithm and in the visualisation process while dealing with high-dimensional dataset. The proposed algorithms are tested both for synthetic and real high-dimensional datasets. The real dataset, we use is related to Major Histocompatibility Complex (MHC) class-I proteins. The experiments show that the adaptation worked successfully with data of more than 2000 dimensions and we also compare the results with other linear/non-linear projection methods: principal component analysis (PCA), Neuroscale (NSC) and the Gaussian process latent variable model (GPLVM).

4.1 Introduction

Nowadays, there is a frequent practical need to analyse datasets with high-dimensions in the bioinformatics domain related to 'omics': proteomics, genomics and metabolomics. With existing algorithms, we often find it difficult to analyse such datasets due to limited computational resources and numerical precision issues. We, in this chapter, address the issue of numerical precision which causes the algorithms to fail while working with the high-dimensional dataset. We consider the generative topographic mapping (GTM) (Bishop and Svensen, 1998) and two of its extensions: GTM with simultaneous feature saliency (GTM-FS) (Maniyar and Nabney, 2006a) and hierarchical GTM (HGTM) (Tino and Nabney, 2002). The GTM visualisation algorithm was proposed as an alternative to the SOM to estimate a generative probability distribution. The GTM-FS was proposed to estimate feature saliencies as an integral part of the training process whereas HGTM was proposed to visualise a set of GTMs in a tree like structure. The standard versions of GTM, GTM-FS and HGTM suffers numerical precision problems while working with high-dimensional dataset (Schroeder, 2009) and therefore, we here propose that these problems can be avoided by using log-space transformations at certain steps of the training process to make them work effectively with high-dimensional datasets. We briefly explain the log-transformations that we require to use while extending these models.



(a) Standard GTM (b) GTM with log-space

(c) Standard GTM (d) GTM with log-space

Figure 4.1: Learning curves and projections of the standard GTM and the GTM with log-space (LogGTM) on synthetic high-dimensional dataset.

4.2 Log-transformations

Because GTM is a constrained mixture of Gaussians, the E-step is the same as for a standard Gaussian mixture model. Thus the likelihoods computed for very high-dimensional datasets can be zero for all components (due to rounding error). This implies that the EM algorithm fails to converge to a sensible solution (i.e. the learning curve of stnadard GTM using 500-dimensional synthetic dataset appears as a flat line over all the iterations, see Figure 4.1(a)) and often leads to a visualisation plot with all the points mapped to the centre of the plot (see Figure 4.1(c))).

We propose here to use log-transformations at certain steps in the GTM training process to avoid such numerical problems. Instead of working with probability density value t_i, the log value $\log t_i$ is used instead. This gives greater precision for the very small values that often occur in high-dimensional data.

To make this effective, we also have to modify the EM algorithm to use the transformed values by using two identities. The first identity is well known: the product of real-space values is equivalent to the sum of log-space values.

$$\log\left(\prod_i t_i\right) = \sum_i \log t_i. \tag{4.1}$$

The second identity is used less frequently and shows how a sum of probabilities can be computed in log-space (Bishop, 2006)

$$\log\left(\sum_i t_i\right) = \eta + \log\left(\sum_i \exp(\log t_i - \eta)\right) = \underset{i}{\mathcal{S}}(\log t_i), \tag{4.2}$$

where $\eta = \max_i \log t_i$ in order to avoid numerical precision errors and $\mathcal{S}$ is the operator for representing log over sum operator. The operator $\mathcal{S}$ is also used for adding two matrices (like $\mathcal{S}(-,-)$) with elements on the log scale.

4.3 GTM with Log-Space Probabilities

In this section we demonstrate how the log transformation can be applied to a mixture of spherical Gaussians in the log-space (as shown in equation (4.3)) and hence be used with a GTM to compute the probability that a data point $\mathbf{x}_n$ is generated by the kth component.

$$\begin{aligned}\log p(\mathbf{x}_n|\mathbf{z}_k,\mathbf{W},\beta_k) &= \log\left[\left(\frac{\beta_k}{2\pi}\right)^{\frac{D}{2}} \exp\left\{-\frac{\beta_k}{2}||\mathbf{x}_n - f(\mathbf{z}_k,\mathbf{W})||^2\right\}\right] \\ &= -\frac{\beta_k}{2}||\mathbf{x}_n - f(\mathbf{z}_k,\mathbf{W})||^2 + \frac{D}{2}\log\left(\frac{\beta_k}{2\pi}\right).\end{aligned} \tag{4.3}$$

The posterior probabilities (i.e. component responsibilities), $p(\mathbf{z}_k|\mathbf{x}_n,\mathbf{W},\beta) = r_{kn}$, in the log-space are computed first and then converted back to real space using

$$\begin{aligned}r_{kn} &= \exp\big(\log p(\mathbf{z}_k|\mathbf{x}_n,\mathbf{W},\beta)\big) \\ &= \exp\left(\big(\log p(\mathbf{x}_n|\mathbf{z}_k,\mathbf{W},\beta_k) + \log(\pi_k)\big) - \left(\underset{k}{\mathcal{S}}\big(\log p(\mathbf{x}_n|\mathbf{z}_k,\mathbf{W},\beta) + \log(\pi_k)\big)\right)\right)\end{aligned} \tag{4.4}$$

Once we obtain the component responsibilities (i.e. r_{kn}) from equation (4.4) then the rest of the EM algorithm of GTM remains the same (i.e. as explained in equations 3.22-3.24). Figures 4.1(b) and 4.1(d) also demonstrates the improvement in the learning process and visualisation plot respectively when log-space transformations are used with GTM model. In this case using a 500-dimensional four class synthetic dataset with 3200 data points shows that all the points are visualised close together on the centre of latent space (indistinguishably to the eye) using the standard GTM whereas the use of the log-space spreads them across the 2D space (see FigurestandardGTMError). However the fact that the points tend to be located on the Gaussian centres giving a 'grid-like' appearance to the visualisation with the log-space GTM variant. This can be the subject of further research in order to improve the visualisation results.

4.4 GTM-FS with Log-Space Probabilities

Like GTM, GTM-FS also faces numerical precision issues while working with high-dimensional datasets. We propose, in the following subsection, a variant of the EM algorithm for GTM-FS that uses the log-space values to avoid any such numerical problems.

4.4.1 An EM Algorithm for GTM-FS using Log-Space

We present here a variant of the EM training algorithm for GTM-FS that uses log-space and is able to deal high-dimensional data both for visualization and feature saliency estimation purposes.

In the EM algorithm of GTM-FS, the dth feature is considered to be relevant with probability $\rho_d = (\psi_d = 1)$: in that case, a mixture component $p(.|\theta_{kd})$ is used to generate its value; otherwise a common density represented by $q(.|\lambda_d)$ is used. The EM algorithm for standard GTM-FS has already been explained in section 3.3.4.1.

We take y (the hidden component labels) and ψ_d to be the missing variables. In the **E-Step** using the current parameter set Ω, log-space posterior probabilities (i.e. $\log r_{nk} = \log p(y_n = k|\mathbf{x}_n)$) can be calculated for the kth Gaussian component for each data point as

$$\begin{aligned}\log r_{nk} = &\left[\log \pi_k + \sum_{d=1}^{D} \Big(\mathcal{S} \left((\log \rho_d + \log p(x_{nd}|\theta_{kd})\right), \right.\\ &\left.(\log(1-\rho_d) + \log q(x_{nd}|\lambda_d)))\Big)\right] \\ &- \underset{k}{\mathcal{S}} \left[\log \pi_k + \sum_{d=1}^{D} \Big(\mathcal{S} \left((\log \rho_d + \log p(x_{nd}|\theta_{kd})\right), \right.\\ &\left.(\log(1-\rho_d) + \log q(x_{nd}|\lambda_d)))\Big)\right].\end{aligned} \tag{4.5}$$

Some of the terms used in equation (4.5) are defined in equations (4.6) and (4.7).

$$\begin{aligned}\log p(x_{nd}|\theta_{kd}) = &-\frac{\beta_{kd}}{2}||x_{nd}-\mu_{kd}||^2 \\ &+\log(\sqrt{\beta_{kd}})-\log(\sqrt{2\pi}),\end{aligned} \tag{4.6}$$

$$\begin{aligned}\log q(x_{nd}|\lambda_{d}) = &-\frac{\beta_{d}}{2}||x_{nd}-\mu_{d}||^2 \\ &+\log(\sqrt{\beta_{d}})-\log(\sqrt{2\pi}).\end{aligned} \tag{4.7}$$

Based on the responsibility matrix, $\mathbf{R}$, (as shown in equation (4.5)), the value $u_{nkd} = p(\psi_d = 1, y_n = k|\mathbf{x}_n)$ can be calculated which explains how relevant the nth data point is to the kth component when the dth feature is considered and $v_{nkd} = p(\psi_d = 0, y_n = k|\mathbf{x}_n)$ shows the irrelevance (noise) of the nth data point relating to the kth component when the dth feature is considered.

$$\begin{aligned}u_{nkd} =& \exp[\log r_{nk} + \{\log\rho_d + \log p(x_{nd}|\theta_{kd}) \\ &- \mathcal{S}(\log\rho_d + \log p(x_{nd}|\theta_{kd}), (\log(1-\rho_d) + \log q(x_{nd}|\lambda_d))\}],\end{aligned} \tag{4.8}$$

$$v_{nkd} = \exp(\log r_{nk}) - u_{nkd}. \tag{4.9}$$

Now, in the $M-$step these posterior probabilities (i.e. responsibilities) are used for estimating the weight matrix $\mathbf{W}$ by solving the following set of linear equations for the dth feature,

$$\widehat{\mathbf{w}}_d = (\mathbf{\Phi}^T\mathbf{E}_d\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{U}_d\mathbf{x}_d, \tag{4.10}$$

where $\mathbf{\Phi}$ is a $K \times L$ matrix, $\hat{\mathbf{w}}_d$ is an $L \times 1$ weight vector, $\mathbf{U}_d$ is a $K \times N$ matrix calculated using equation (4.8), $\mathbf{x}_d$ is an $N \times 1$ data vector, and $\mathbf{E}_d$ is a $K \times K$ matrix with elements

$$e_{kkd} = \sum_{n=1}^{N} u_{nkd}. \tag{4.11}$$

Now, using the re-estimated matrix $\widehat{\mathbf{W}}$, the centres of the mixture components in the data space can be updated using the mapping function

$$\widehat{\text{Mean}\,\theta}_k = \widehat{\mu}_k = \mathbf{\Phi}_k\widehat{\mathbf{W}}, \tag{4.12}$$

where μ_k represents a $1 \times D$ updated mean vector. After updating the centres for the mixture components in the data space, the variance of the Gaussian for the dth feature can be calculated

$$\frac{1}{\widehat{\beta}_d} = \exp\left(\underset{k}{\mathcal{S}}\underset{n}{\mathcal{S}} \left(\log u_{nkd} + \log[(x_{nd} - \mu_{kd})^2] \right) - \underset{k}{\mathcal{S}}\underset{n}{\mathcal{S}}(\log u_{nkd}) \right). \tag{4.13}$$

The parameters λ of the common density $q(x_{nd}|\lambda_d)$ and feature saliencies are updated using a similar formula as in the original GTM-FS algorithm (using equation (3.37) for the mean of the common density, equation (3.38) for the variance of the common density and equation (3.41) for the feature saliencies updates).

4.5 Hierarchical GTM (HGTM) with Log-Space Probabilities

A hierarchical GTM is a tree-structured visualisation model which attempts to improve the visualisation by adding more levels in a hierarchy. The fundamental building block of HGTM is a GTM, and like standard GTM it also suffers from numerical problems when working with high-dimensional datasets. We propose that using log-transformations at certain steps of HGTM training process will also help to avoid such numerical problems.

4.5.1 An EM Algorithm for HGTM using Log-Space

The detailed derivation of the original HGTM *expectation maximization* (EM) training algorithm is given in (Tino and Nabney, 2002) and here, we explain the main steps in the summarised form where we propose to use log-transformations to avoid numerical precision errors.

E-Step:

In the E-step of HGTM training process, posteriors over all hidden variables are estimated using the current parameters set of HGTM using the log-space transformation to avoid numerical errors. For a given data point $\mathbf{x}_n$ we compute *model* responsibilities which define the competitive relationship among models that belong to the same parent node in the tree using

$$\begin{aligned} \log p(\mathcal{M}|Parent(\mathcal{M}), \mathbf{x}_n) = {} & \big(\log \pi(\mathcal{M}|Parent(\mathcal{M})) + \log p(\mathbf{x}_n|\mathcal{M})\big) \\ & - \left(\underset{\mathcal{N}\in[\mathcal{M}]}{\mathcal{S}} \big((\log \pi(\mathcal{N}|Parent(\mathcal{M})) + \log p(\mathbf{x}_n|\mathcal{M})\big) \right) \end{aligned} \tag{4.14}$$

where $[\mathcal{M}] = Children(Parent(\mathcal{M}))$. We also recursively compute unconditional (on parent) model responsibility using

$$\log p(\mathcal{M}|\mathbf{x}_n) = \log p(\mathcal{M}|Parent(\mathcal{M}), \mathbf{x}_n) + \log p(Parent(M)|\mathbf{x}_n) \tag{4.15}$$

and now for the $\mathcal{M}$th model, responsibilities of the latent space centres (i.e. $\mathbf{z}_k$ with $k = 1, \ldots, K$) are computed in the log-space using

$$\begin{aligned}\log(R_{kn}^{\mathcal{M}}) &= \log p(\mathbf{z}_k^{\mathcal{M}}|\mathbf{x}_n, \mathbf{W}_{\mathcal{M}}, \beta_{\mathcal{M}}) \\ &= \Big(\log p(\mathbf{x}_n|\mathbf{z}_k^{\mathcal{M}}, \mathbf{W}_{\mathcal{M}}, \beta_{\mathcal{M}}) + \log(\pi_k)\Big) - \Big(\underset{k}{\mathcal{S}}\left(\log p(\mathbf{x}_n|\mathbf{z}_k, \mathbf{W}, \beta_{\mathcal{M}}) + \log(\pi_k)\right)\Big)\end{aligned} \tag{4.16}$$

The responsibilities can be rescaled in the log-space form using equations (4.15) and (4.16) and these are then converted back to real-space

$$(\mathbf{R}_{\mathcal{M}})_{kn} = \exp\Big(\log p(\mathcal{M}|\mathbf{x}_n) + \log(R_{kn}^{\mathcal{M}})\Big) \tag{4.17}$$

M-Step:

Now the parameters can be re-estimated using the posteriors computed at the E-step. At first parent-conditional mixture coefficient are evaluated using

$$\pi(\mathcal{M}|Parent(\mathcal{M})) = \exp\Big(\underset{n}{\mathcal{S}}\left(\log p(\mathcal{M}|\mathbf{x}_n)\right) - \underset{n}{\mathcal{S}}\left(\log p(Parent(\mathcal{M})|\mathbf{x}_n)\right)\Big) \tag{4.18}$$

The weight matrix, $\widehat{\mathbf{W}_{\mathcal{M}}}$, for each $\mathcal{M}$th model can be updated using the following set of linear equations

$$\widehat{\mathbf{W}_{\mathcal{M}}} = (\mathbf{\Phi}_{\mathcal{M}}^T \mathbf{E}_{\mathcal{M}})^{-1} \mathbf{\Phi}_{\mathcal{M}}^T \mathbf{R}_{\mathcal{M}} \mathbf{X} \tag{4.19}$$

where $\mathbf{\Phi}_{\mathcal{M}}$ is a $K_{\mathcal{M}} \times L_{\mathcal{M}}$ matrix with elements, $(\mathbf{\Phi}_{\mathcal{M}})_{kl} = \phi_l(\mathbf{z}_k^{\mathcal{M}})$, $\mathbf{R}_{\mathcal{M}}$ is $K_{\mathcal{M}} \times N$ and these scaled responsbilites are updated in the E-Step using equation 4.17, $\mathbf{X}$ is an $N \times D$ data matrix and $\mathbf{E}_{\mathcal{M}}$ is $K_{\mathcal{M}} \times K_{\mathcal{M}}$ with elements

$$(\mathbf{E}_{\mathcal{M}})_{kk} = \sum_{n=1}^{N} (\mathbf{R}_{\mathcal{M}})_{kn} \tag{4.20}$$

Finally the inverse variance β can be re-estimated using

$$\frac{1}{\widehat{\beta_{\mathcal{M}}}} = \exp\left(\log\left(\sum_{n=1}^{N}\sum_{k=1}^{K} (\mathbf{R}_{\mathcal{M}})_{kn} ||\mathbf{W}_{\mathcal{M}} \phi(\mathbf{x}_k^{\mathcal{M}}) - \mathbf{x}_n||^2\right) - \Big(\log(D) + \underset{n}{\mathcal{S}}\left(\log p(\mathcal{M}|\mathbf{x}_n)\right)\Big)\right). \tag{4.21}$$

4.6 Experiments

We have carried out a series of experiments in order to demonstrate the effectiveness of our proposed GTM variants on high-dimensional synthetic and real datasets. In these experiments, we initialised the weight matrix, $\mathbf{W}$, using PCA. We took a latent grid of size 8×8 for both real and synthetic datasets and used an RBF grid of size 6×6 for the synthetic dataset and 4×4 for the real dataset. For the dataset of MHCs, we compared visualisation results of LogGTM and LogGTM-FS with other visualisation algorithms such as PCA, Neuroscale and GPLVM. Label information was only used to colour the data points on the visualisation space and to measure the visualisation quality metrics of KL-divergence and NN classification error. We also computed other visualisation quality measures as explained in Chapter 3 to evaluate the quality of visualisations.

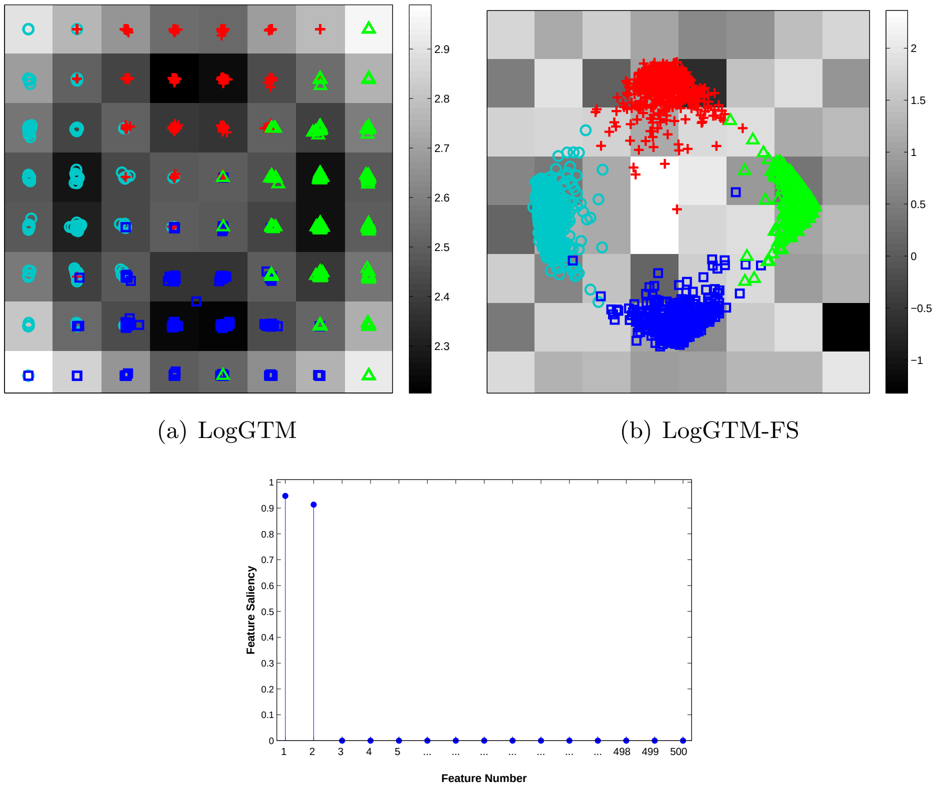
4.6.1 Synthetic dataset

The synthetic dataset we generated is of $3,200$ data points from an equiprobable mixture of four two-dimensional Gaussians, $\mathcal{N}(\mathbf{m}_k, I)$ with $k = 1, \dots, 4$ with the following mixture means: $\mathbf{m}_1 = \binom{0}{3}$, $\mathbf{m}_2 = \binom{1}{9}$, $\mathbf{m}_3 = \binom{6}{4}$ and $\mathbf{m}_4 = \binom{7}{10}$. We then generated 498 noisy features (where each feature is sampled from $\mathcal{N}(0, I)$ density) and appended these to the data leading to a 500-feature dataset with $3,200$ data points.

Visualisation results of the LogGTM and LogGTM-FS are presented in Figure 4.2(a) and 4.2(b) respectively: the background gray colour indicates stretch level in the visualisation manifold (i.e. the lighter the colour the more the stretch in the visualisation manifold). Feature saliencies estimated from the LogGTM-FS are given in Figure 4.2(c). Visualisation results from the LogHGTM are shown in Figure 4.3 with the magnification factors for both levels in the hierarchy.

4.6.1.1 Discussion

As discussed earlier in section 4.3, with the visualisation of high-dimensional data (usually with greater than a few hundred dimensions) using GTM with log-space values (i.e. LogGTM), tight clusters around the centres of the latent visualisation grid were observed (see Figure 4.2(a)). However, we were still able to observe separation of the four classes in the visualisation space with some parts overlapping across class boundaries. The issue of tight clusters around the grid centres with GTM like models require further research in order to improve visualisation and better understand the hidden structure in the dataset. In the particular case of synthetic dataset here only two features have four clusters whereas



(a) LogGTM

(b) LogGTM-FS

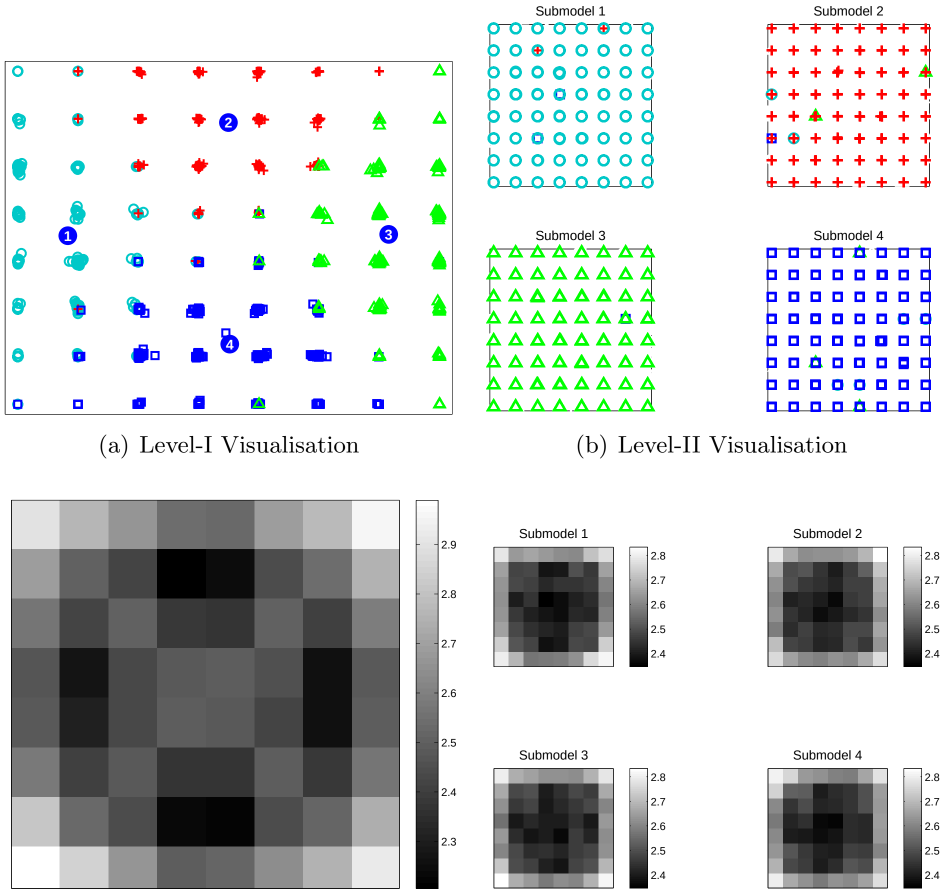
(c) LogGTM-FS (estimated saliencies)

Figure 4.2: Demonstration of visualisation of 500 dimensional synthetic dataset in the presence of irrelevant ('noisy') features using the LogGTM (see (a)) and the LogGTM-FS (see (b)) variants. The figure (c) show saliencies estimated from the LogGTM-FS model.

the remaining 498 are just the noisy features added in the dataset for the purpose of demonstrating usefulness of GTM-FS model with log space values (i.e. LogGTM-FS).

As expected, on the LogGTM-FS manifold there are four coherent (compact) clusters (see Figure 4.2(b)) (compared to LogGTM visualisation where each class was observed to be more spread across the latent space (see Figure 4.2(a)) with tight clusters of projected data points onto the centres of latent space grid). This compactness is achieved with the LogGTM-FS by reducing the impact of irrelevant ('noisy') features in the model learning process by modelling them with a shared distribution $q(.|\lambda)$. The saliencies estimated with LogGTM-FS proved that there were only two informative features whereas the remaining 498 features are irrelevant with saliencies equal to zero (see Figure 4.2(c)).

The results of LogHGTM are also interesting with four sub models (initialised interactively on the root-level visualisation (see Figure 4.3(a))) where each sub-model is showing data points of each class (see Figure 4.3(b)) with a few points from other classes (due to the noisy features). The LogHGTM results at each level in the hierarchy also appeared as tight clusters on the centres of the latent grid. Extending LogHGTM to simultaneously



(a) Level-I Visualisation

(b) Level-II Visualisation

(c) Level-I (MF)

(d) Level-II (MF)

Figure 4.3: Demonstration of a high-dimensional synthetic dataset visualisation using log-based hierarchical GTM model (where (a) and (b) for level-I and level-II visualisations respectively) and magnification factor (MF) plots on $\log_{10}$ to demonstrate stretches on the visualisation plots (where (c) and (d) are magnification factor plots for level-I and level-II respectively).

estimate feature saliencies would be an interesting extension to model high-dimension datasets in the presence of noisy features. The extension is non-trivial because a parent GTM is a mixture of its child GTMs, so it would be necessary to enforce the same saliencies over the whole tree of models.

4.6.2 Electrostatic potential dataset of MHC class-I

A real dataset related to MHC class-I, as described in Section 2.1.7, was used to demonstrate the effectiveness of our proposed variants. We aim to identify similarities in HLA-A, HLA-B and HLA-C genetic alleles based on an electrostatic potential map around the top surface covering the $\alpha 1$ and $\alpha 2$ regions (where the electrostatic potential calculated at each grid point over the target area is taken as a descriptor).

The visualisation results obtained with PCA, Neuroscale, LogGTM , LogGTM-FS and

GPLVM are shown in Figure 4.4. The background grey colour in Figures 4.4(c) and 4.4(d) indicates magnification factors (on a $\log_{10}$ scale) for the visualisation (projection) manifolds (i.e. the lighter the colour the more stretch in the visualisation manifold) whereas the background colour in Figure 4.4(g) indicates mapping precision (i.e. the lighter regions show better precision in mapping). Directional curvature plots for the GTM variants (i.e. for LogGTM and LogGTM-FS) are also presented in Figure 4.4 (where longer lines and lighter background indicates high folding (curvature)). We also observed that only 182 descriptors have a saliency of less than 0.5 so $2,236$ descriptors are selected as important from the original total of $2,418$ descriptors.

The visualisation results obtained from LogGTM and LogGTM-FS appeared better than that of PCA and NSC visualisation methods but not better than GPLVM. As discussed with structural biologists, the GPLVM visualisation was useful for the purpose of identifying clusters of alleles among all three gene classes. The visualisation quality evaluation metrics are computed and are presented in Figure 4.5. Visualisation results obtained from the LogHGTM with two level hierarchy along with magnification factors are also presented in Figure 4.6. LogHGTM results have improved separation between classes at lower levels but still suffer the problem of tight cluster of projected data points around the latent grid centres (see Figure 4.6(a)).

4.6.2.1 Comparison to previous supertype analysis of MHC class-I

Although our analysis is generated using different means, it is instructive to compare our results briefly to previous work (Doytchinova et al., 2004). It should be stressed that our results do not use any identifying characteristic of the MHC proteins other than their electrostatic potential map data. Moreover, the present analysis differs from previous work, which focused on the peptide binding site only (Doytchinova et al., 2004), so we should not expect a large overlap in the results; yet all are reasonably similar, and these commonalities between observed clusters are reassuring. With the exception of a few individual alleles, our current analysis effected a near complete separation of HLA-A, B, and C loci. Exceptions to this were clusters 8 and 12 (Figure 4.4(g) of GPLVM visualisation), which may be indicative of some commonality of structural properties corresponding to convergent evolution of HLA-A and HLA-B alleles. Even here, alleles were almost completely separated but in a continuous distribution (i.e. in terms of electrostatic potential map) that is hard to separate further without prior knowledge. It is interesting that HLA-C forms six well separated clusters, which contrasts sharply with previous results, and is perhaps suggestive of the greater variability in the extended surface analysed here relative to HLA-A and B. This is functionally consistent with the interactions made by HLA-C with a wider range of receptors other simply TCRs.

By rigorous state-of-the-art visual analysis of data (using GPLVM visualisation), we have identified clusters corresponding to the three class I human MHC loci, and sub-groups therein. It is notable that the analysis recovers the HLA-A, HLA-B, and HLA-C alleles using only their property distributions, without prior knowledge of a division by loci; the latter information was used only when labelling the result plots. This gives confidence to any assertion we might make regarding the division of the allele population into structurally and functionally similar sub-groups. This information will inform accurate identification of T-cell epitopes, a crucial step when developing epitope ensemble vaccines.

The three different class I HLA loci are possessed of functional differences, such as binding NK receptors, system differences, such as the breadth of anti-HIV responses of different HLA loci (Kiepiela et al., 2004), as well as structural ones, including the observation that different loci have peptide repertoires that are distinct in their size and specificity (Paul et al., 2013). Thus our ability to distinguish the three loci so unequivocally is notable. It implies that the differences are sufficiently strong to be obvious at the level of projected properties alone, and this gives credence to our identification of further subsets within the individual loci.

4.6.2.2 Discussion

The visualisation results we got from the PCA and Neuroscale are very similar and appeared like a blob with most alleles from all three classes overlapped and did not improve our understanding of the grouping structure. GPLVM and our proposed variant LogGTM have shown better results, compared to PCA and Neuroscale, both in terms of visual inspection and in terms of visualisation quality evaluation measures such as trustworthiness, mean relative rank errors with respect to data (in case of LogGTM, only for cases when neighbourhood k is less than 50), nearest-neighbour-classification error and KL divergence. We note that visualisation results from GPLVM are much better than LogGTM because LogGTM has a problem of tight clusters of projected data points around latent grid centres.

Each mapping algorithm makes a tradeoff between trustworthiness and continuity[1]: algorithms like PCA and Neuroscale often have higher continuity than trustworthiness whereas GTM and GPLVM with back-constraints have higher trustworthiness than con-

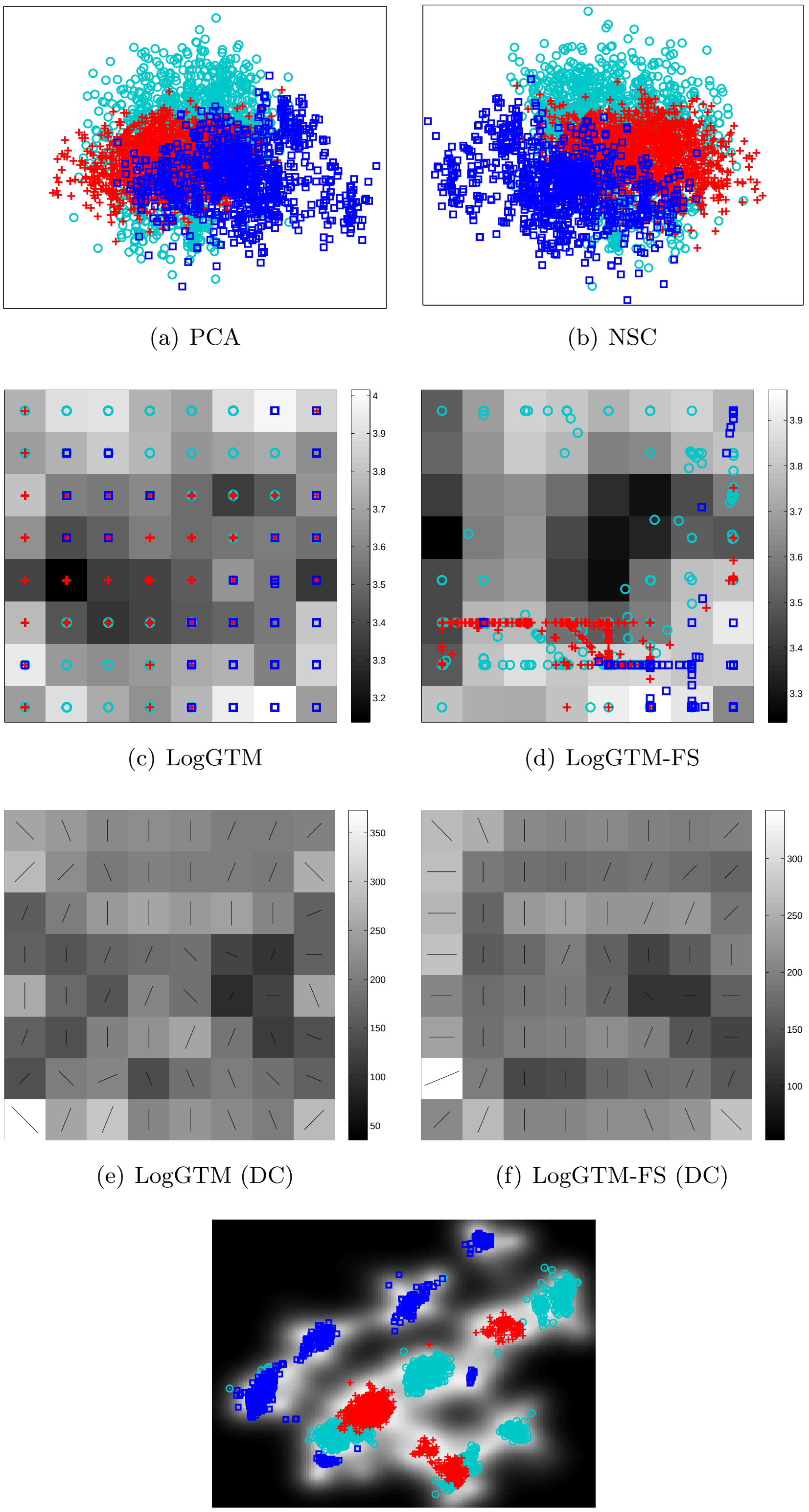
[1] A mapping is said to be trustworthy if k-neighbourhood in the visualised space matches that in the data space but if the k-neighbourhood in the data space matches that in the visualised space it maintains continuity.

tinuity as they focus on preserving local distances (Venna and Kaski, 2005). When users visualise data, it is the local structure that is most relevant to their analysis (for example, when they identify clusters) and therefore trustworthiness is considered to be the better measure than continuity when comparing the visualisation models. The other two quality evaluation measures are mean relative rank errors with respect to data and latent space, which are similar to trustworthiness and continuity but differ because they use rank differences in the k-neighbourhood. In a visualization context, trustworthiness and mean relative rank error with respect to data are more important than continuity and mean relative rank error with respect to latent space as they ensure data points in the k-neighbourhood of visualization space are also neighbours in the data space (Kaski et al., 2003) so any cluster structure that is seen in the latent space is genuine.

The distance distortion measure was also computed and Neuroscale was expected to have better distance distortion with lower values because of the fact that this algorithm attempts to ensure that points which are distant in the data space are projected to points that are distant in the latent space (i.e. attempts to maintain proximity of data space in the visualisation latent space) whereas algorithms like LogGTM, LogGTM-FS and GPLVM with back-constraints were expected to have higher values of distance distortion. Neuroscale is non-linear, but if the projection is defined as a linear mapping then the stress metric yields PCA, so the two algorithms are related. Because PCA and Neuroscale have very similar results in terms of trustworthiness, continuity, mean relative rank errors and distance distortion (with overlapping values on the graphs) to conclude that this is wide evidence for non-linearity in topographic mappings. Therefore for the purpose of clarity we plot results for these only for one of those (i.e. Neuroscale). The Neuroscale results presented here use 60 basis functions and we also repeated experiments with different number of basis function but the results did not vary significantly (see Appendix B).

The proposed variant LogGTM-FS results have shown better preservation of exact ranks in terms of mean relative rank errors with respect to data and latent space and this variant has also shown better KL-divergence whereas for measures such as trustworthiness, continuity and nearest-neighbour-classification error the results were not satisfactory.

We observe that using LogHGTM visualisation for MHC dataset, we get better separation between classes but it still suffers from the problem of projected data points in tight clusters on the centres of latent grid components. It is also observed that first three sub-models of LogHGTM model have shown high-level of stretches for the second level visualisation manifolds by showing the magnification factor plots almost as white (see Figure 4.6(d)).



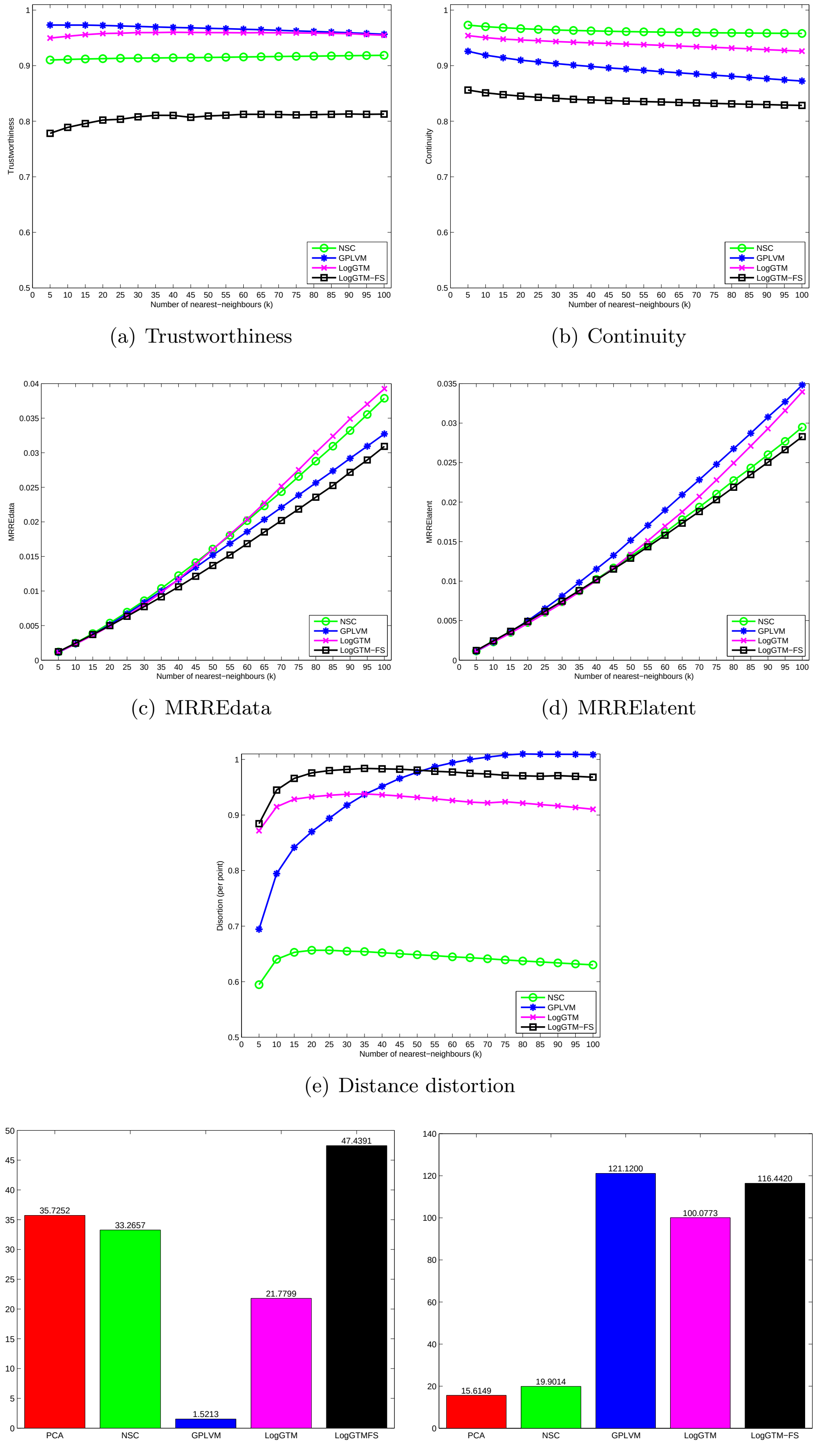
(a) PCA (b) NSC

(c) LogGTM (d) LogGTM-FS

(e) LogGTM (DC) (f) LogGTM-FS (DC)

(g) GPLVM

Figure 4.4: Demonstration of visualisation of the MHC class-I dataset. Cyan circles ('o') for HLA-A, red plus sign ('+') for HLA-B and blue squares ('□') for HLA-C. (c) and (d) show LogGTM and LogGTM-FS visualisations respectively with simultaneous magnification factors (MF) plots on a $\log_{10}$ scale. The lighter grey background regions on MF plots show more stretches in the projection manifold. The (e) and (f) show directional curvature plots for LogGTM and LogGTM-FS respectively where the lighter regions with longer lines show high folding (curvature). The (g) show GPLVM visualisation with grey background indicating mapping precision (the lighter regions correspond to better precision in mapping).



(a) Trustworthiness

(b) Continuity

(c) MRREdata

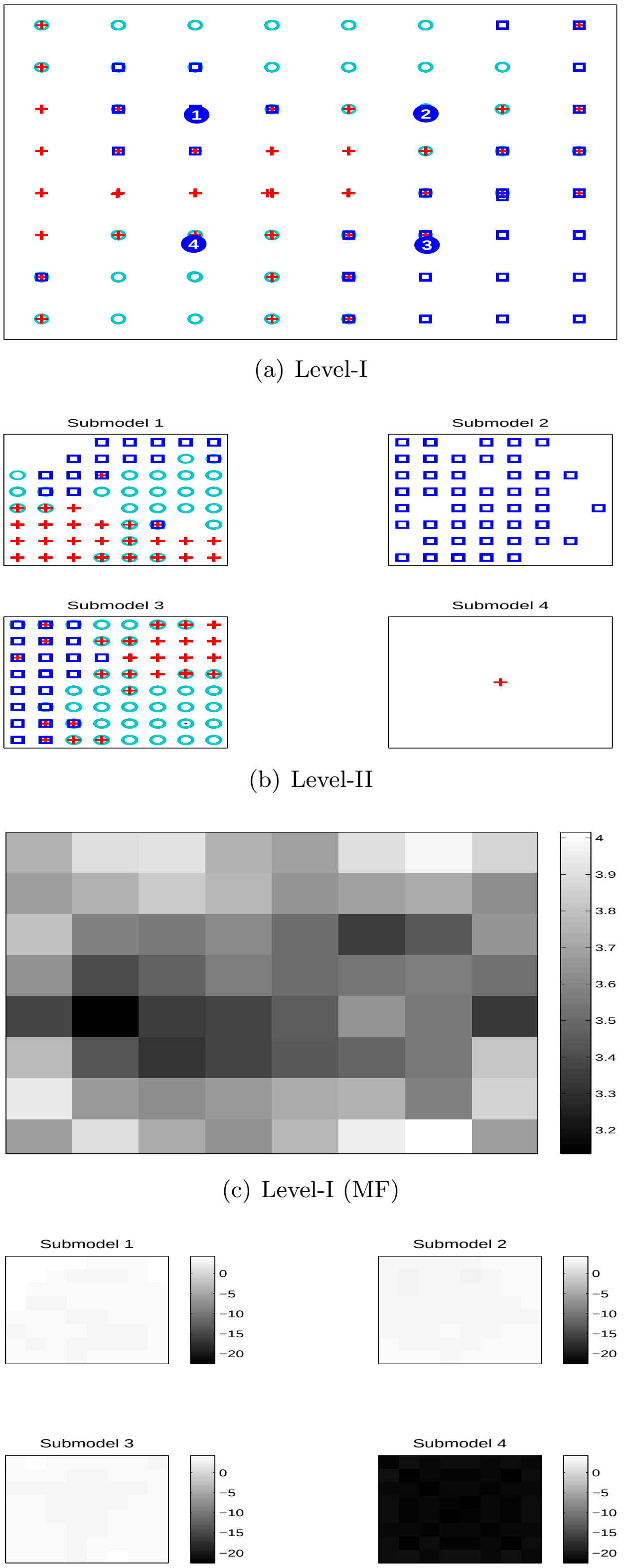
(d) MRRElatent

(e) Distance distortion

(f) NN-Error (%)

(g) KL divergence

Figure 4.5: Visualisation quality evaluation metrics for the MHC class-I dataset. The trustworthiness, continuity and KL divergence, the higher the better the visualisation whereas MRREdata, MRRElatent, distance distortion and NN-error the lower the better the visualisation.



(a) Level-I

(b) Level-II

(c) Level-I (MF)

(d) Level-II (MF)

Figure 4.6: Demonstration of the MHC class-I dataset visualisation using log-based hierarchical GTM model (where (a) and (b) for level-I and level-II visualisations respectively and here legend same as in Figure 4.4) and magnification factor (MF) plots on $\log_{10}$ to demonstrate stretches on the visualisation plots (where (c) and (d) are magnification factor plots for level-I and level-II respectively). In (d), the first three MF sub-plots show high-level of stretches with light grey (nearly white) and the fourth MF sub-plot show low-level stretches for the corresponding visualisations plots respectively.

4.7 Conclusion

Due to the recent advances in the generation process of bioinformatics datasets, large high-dimension datasets are becoming available, bringing a new challenge for the analysts to perform analysis due to the limited computational resources with the existing analysis methods. It is important to modify these methods to make them workable to analyse such datasets within the existing resources and not to suffer any computational issues. The visualisation algorithm GTM and two of its extensions (i.e. GTM-FS and hierarchical GTM) are very useful tools to project multivariate datasets onto the latent visualisation space but issue is that they suffer from some numerical problems while visualising high-dimension datasets with dimensions greater than a few hundred (i.e usually greater than 400).

We are successful in deriving variants of these algorithms (i.e. GTM, GTM-FS and hierarchical GTM) where we use log-transformations at certain steps of the EM parameter learning process in order to avoid the numerical problems that were observed with the standard algorithms. The proposed variant LogGTM is tested successfully with a few thousand dimensions, although we observed data projected in tight clusters around the latent grid centres. The other successful proposed variant is LogGTM-FS. It is observed that this proposed variant provided good visualisation results (with no tight clusters around the grid centres like in LogGTM) in the presence of a large number of irrelevant ('noisy') features in the high-dimensional datasets. In this chapter, usefulness of LogGTM-FS over LogGTM is clearly observed in case of the synthetic dataset but in case of the real dataset not many features are observed to be irrelevant and therefore not much improvement is observed. The LogHGTM variant is also tested successfully on both synthetic and real datasets but suffered the problem of tight clusters of projected data points around the latent grid centres.

Visualisation of the 'MHC class-I' dataset with PCA, Neuroscale and GTM algorithms have not shown clear separation of the alleles of each HLA loci but instead the alleles of all three loci overlap (as shown in Figure 4.4). But applying non-linear visualisation methods such as GPLVM with MLP as a back constraint have clearly shown better separation between alleles of each gene.

5 Discrete Data Visualisation with Simultaneous Feature Selection

CONTENTS

Both data visualisation and feature selection methods are widely used as two distinct methods for analysing complex large and high-dimensional datasets. A combined data visualisation and feature saliency estimation model was proposed by Maniyar and Nabney (2006a) suitable only for continuous data and here we extend this approach for handling discrete data. We derive a visualisation model based on the latent trait model (LTM) suitable for discrete data to simultaneously estimate feature saliencies as an integrated part of the parameter learning process. This combined approach not only attempts to improve visualisation by reducing the impact of irrelevant (noisy) features but also estimate the saliency of each feature, which is valuable information in its own right. The saliency value lies in a range of 0 to 1 reflecting the importance level of a feature. The effectiveness of the proposed combined approach is demonstrated on synthetic and real datasets.

5.1 Introduction

The datasets we focus on in this chapter have discrete type features. Compared to continuous features datasets, less attention is given in the literature to work with discrete features datasets under the latent variable framework. Analysing these multivariate large discrete features datasets is gaining a lot attention of machine learning experts to develop models that help users to get better understanding of the complexity.

In principle, the machine learning algorithms assume to perform well in cases where we have more information about data instances. This suggests that the use of more features is important for the learning algorithms. However, in practice it is observed that not all the features are important. It is therefore important to select a subset of features which are important thereby ignoring the irrelevant (noisy) features which compromise performance of the learning algorithm. In addition, an understanding of which features are relevant is valuable in its own right. In the exploratory phases of analysis (which is when data visualisation is most used) it is usual to measure as many variables as is feasible, since it is not known which are relevant to the task. Feature selection then plays an important role in simplifying the task and making data collection cheaper and faster.

Feature selection (FS) has been widely used in supervised learning problems where the search is guided by the known target values. FS methods can be categorized into four classes (Silvestre et al., 2013; Alelyani et al., 2013): filters, wrappers, hybrid and embedded.

Filter approaches determine the importance of a feature or a subset of features from the intrinsic characteristics of the data independent of the learning algorithm. Filter approaches are usually fast to compute and can be used as part of a two-step process in conjunction with a learning method: in the first step features are selected independently of the learning algorithm and in the second step the learning algorithm is applied on the selected subset of features.

Wrapper approaches use interaction between subsets of features and the learning algorithm. A subset of features is selected from several candidate feature subsets in a sequential way (either using a forward selection approach where a search starts considering an empty set of features or a backward selection approach where a search starts taking all the features into consideration (Kohavi and John, 1997)) in order to improve the quality of the learning algorithm. Wrapper approaches are slower than filter approaches. *Hybrid* approaches take advantage of both the approaches in a two-stage process by taking a computational efficiency of filter approaches and accuracy of wrapper approaches. Hybrid methods are computationally faster than wrapper methods but slower than filters.

Embedded methods use feature selection as integrated part of a learning algorithm thus ensuring that the selected features are those relevant to the specific learning algorithm. In embedded methods, instead of using a selected subset of features, all features are exploited in the learning process to compute their saliencies where the important features have higher saliency values and less important features have lower saliency values. Performance of embedded methods depends on the learning algorithm but are reported to be faster than wrapper approaches (Vinh et al., 2012).

Feature selection for unsupervised learning algorithms is a difficult and a challenging task as there are no target values known to guide the search. In the literature, not much attention is given to this problem. Very few attempts have been made to estimate the importance of features in the unsupervised learning algorithms. A brief review of feature selection in a clustering perspective is given by Alelyani et al. (2013).

In the model based clustering perspective, Law et al. (2004) proposed an approach where they use Gaussian mixture models for clustering and estimate saliencies of all the features (in a range of 0 to 1) as an integrated part of the clustering algorithm training process. They also use a minimum message length (MML) criterion for model selection. Most of the work on feature selection with simultaneous clustering is done on continuous data using the Gaussian mixture model, whereas work on discrete (binary and multi-categorical) data clustering with simultaneous feature selection is relatively rare. Wang and Kabán (2005) and Bouguila (2010) extend the similar approach proposed by Law et al. (2004) for clustering binary data with simultaneous feature selection using a mixture of Bernoulli distributions. Bouguila (2010) also uses a Bayesian Information Criteria

(BIC) for model selection purpose, while Silvestre et al. (2013) also extends a similar approach for clustering categorical data with simultaneous feature selection using a mixture of multinomial variables. Law et al. (2004) and Bouguila (2010) use a maximum *a posteriori* (MAP) approach for parameter learning, whereas Wang and Kabán (2005) and Silvestre et al. (2013) use a maximum likelihood approach for learning parameters of the model.

Our focus here is on data visualisation which is usually considered to get a new transformed extracted low-dimensional feature space for the representation of high-dimensional data space. However, an approach of extracting a low-dimensional embedded space with simultaneously considering the importance of each feature in the learning process was adapted for GTM by Maniyar and Nabney (2006a) and such an approach can be termed an embedded approach of feature selection. Adapting the approach proposed by Maniyar and Nabney (2006a), we propose here an extension to an LTM (a visualisation model for discrete data) to simultaneously estimate feature saliencies while training a visualisation model.

5.2 Related Work

The most closely related work to address this problem was proposed by Maniyar and Nabney (2006a) which extended a GTM to simultaneously estimate feature saliencies. GTM-FS visualisation model is an extension of the approach proposed by Law et al. (2004) and is explained in chapter 4. Mumtaz et al. (2012) extended GTM-FS to adapt log-transformations at certain steps of EM training to avoid numerical problems that are observed usually while working with datasets of more than 300-dimensions.

Vellido et al. (2006) also proposed a variant of the GTM to handle missing values and outliers in a robust way using a mixture of Student *t*-distributions (also known as t-GTM) and they also extended t-GTM to simultaneously estimate feature saliencies while training a visualisation model. Vellido et al. (2006) also extended the computation of feature saliency using Gaussian-GTM (the standard GTM) as proposed in (Vellido, 2006, 2005). Approaches proposed by Maniyar and Nabney (2006a) and Vellido (2006, 2005) are very similar. A variational GTM was proposed by Caparroso (2008) and it was also extended to estimate feature saliencies while training a visualisation model using a similar principle as proposed by Maniyar and Nabney (2006a); Vellido et al. (2006); Vellido (2006, 2005).

To our knowledge no approach currently exists for multivariate *discrete* data visualisation with simultaneous feature selection. We propose here an extension to the LTM to simultaneously estimate feature saliencies (LTM was developed as a generalisation of GTM to handle different type of data mainly focusing on multivariate discrete data visualisation).

5.3 Discrete Data Visualisation

Kabán and Girolami (2001) proposed a visualisation model, similar in principle to GTM, with the use of the exponential family of distributions to model the noise in the data. The model was proposed to handle different types of data mainly focusing on discrete datasets and is known as the latent trait model (LTM) (because of the multivariate discrete data visualisation on the continuous latent space). LTM uses the Bernoulli distribution for modelling binary data and the multinomial distribution for modelling multi-categorical data. Like GTM, LTM also uses a continuous two-dimensional latent space, $M \times K$ (where $M = 2$), with uniform grid points (representing Bernoulli/Multinomial distribution means) for visualisation. A single data centre $\mathbf{z}_k$ in the latent space, $\mathcal{H}$, can be mapped back to a data space, $\mathcal{D}$, using the corresponding link functions as explained in the following subsections. Detailed derivation of LTM model is available in (Kabán and Girolami, 2001).

5.3.1 Mapping function for binary data

The link function for the binary feature dataset for the kth latent point $\mathbf{z}_k$ is the expected value (i.e. the probability of weighted sum of success) obtained by applying a logistic sigmoid function composed with the mapping function $f(\mathbf{z}_k, \mathbf{W}) = \mathbf{\Phi}(\mathbf{z}_k)\mathbf{W}$,

$$\mathbf{m}_k = g(f(\mathbf{z}_k, \mathbf{W})) = \frac{\exp(\mathbf{\Phi}(\mathbf{z}_k)\mathbf{W})}{1 + \exp(\mathbf{\Phi}(\mathbf{z}_k)\mathbf{W})}, \tag{5.1}$$

where $\mathbf{\Phi}(\mathbf{z}_k) = \left\{\phi_l(\mathbf{z}_k), \ldots, \phi_L(\mathbf{z}_k)\right\}$ is a set of fixed non-linear basis function (we use here a radial basis function), $\mathbf{W}$ is an $L \times D$ matrix of weights.

5.3.2 Mapping function for multi-categorical data

Each dth categorical feature is encoded into a 1-of-S_d binary encoded set (i.e. where S_d represents the number of categories in the dth feature). The link function for the multi-category feature dataset for any kth latent point $\mathbf{z}_k$ is the expected value (i.e. the probability of weighted sum of relating to each category) which is obtained by applying the *softmax* function in composition with the mapping function $f(\mathbf{z}_k, \mathbf{w}_{s_d}) = \mathbf{\Phi}(\mathbf{z}_k)\mathbf{w}_{s_d}$,

$$\mathbf{m}_{ks_d} = g(f(\mathbf{z}_k, \mathbf{w}_{s_d})) = \frac{\exp(\mathbf{\Phi}(\mathbf{z}_k)\mathbf{w}_{s_d})}{\sum_{s'_d=1}^{S_d} \exp(\mathbf{\Phi}(\mathbf{z}_k)\mathbf{w}_{s'_d})}, \tag{5.2}$$

where $\mathbf{\Phi}(\mathbf{z}_k) = \left\{\phi_l(\mathbf{z}_k), \ldots, \phi_L(\mathbf{z}_k)\right\}$ is a set of fixed non-linear basis function (we use here a radial basis function), $\mathbf{w}_{s_d}$ is an $L \times 1$ vector of weights for the sth binary encoding of the dth feature.

5.3.3 Mixture model density function

The Bernoulli and multinomial probability distribution are given in equation (5.3) and (5.4) respectively

$$p(\mathbf{x}_n|\mathbf{m}_k) = \prod_{d=1}^{D} m_{kd}^{x_{nd}}(1 - m_{kd})^{1-x_{nd}}, \tag{5.3}$$

where m_{kd} is the mean of the dth variable of the kth Bernoulli distribution.
Considering a dataset $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, where each data point $\mathbf{x}_n = \{x_{n1}, \cdots, x_{nD}\}$ contains D features with discrete values. More specifically, we consider that each dth feature can have S_d discrete values (i.e. $x_{nd} \in \left\{\gamma_1, \cdots, \gamma_{S_d}\right\}$). We here assume that each feature $\mathbf{x}_d$ can be modelled taking the multinomial distribution. We take $m_{s_d} = p(\mathbf{x}_d = \gamma_{s_d})$ and $\sum_{s_d=1}^{S_d} m_{s_d} = 1$. Now assuming that the features are independent, the probablility density function for a data vector $\mathbf{x}_n$ for a given $\mathbf{m}_k$ is defined as,

$$p(\mathbf{x}_n|\mathbf{m}_k) = \prod_{d=1}^{D} \prod_{s_d=1}^{S_d} m_{ks_d}^{x_{ns_d}}, \tag{5.4}$$

where $x_{ns_d} = I(x_{nd}, s)$ and $I(x_{nd}, s)$ is a binary encoding function which is 1 if $x_{nd} = \gamma_{s_d}$ otherwise it is 0.

The mixture model probability density function for the latent trait model (LTM) can now be represented as a sum of Bernoulli/Multinomial distributions

$$p(\mathbf{x}_n|\pi, \mathbf{m}) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}_n|\mathbf{m}_k), \tag{5.5}$$

where K is a number of mixture components, π_k is the mixing coefficient of the kth component in the mixture with a $p(\mathbf{x}_n|\mathbf{m}_k)$ as a Bernoulli/multinomial distribution.

5.4 Discrete Data Visualisation with Simultaneous Feature Selection (FS)

For the purpose of estimating feature saliencies, features are assumed to be independent of the component label and a mixture model can be represented as,

$$p(\mathbf{x}_n|\pi, \mathbf{m}) = \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} p(x_{nd}|m_{kd}), \tag{5.6}$$

where $p(.|m_{kd})$ is the probability density function for the dth feature of the kth component and π_k is taken as a mixing co-efficent of the kth component.

We assume $\Psi = \{\psi_1, \psi_2, \ldots, \psi_D\}$ to be a set of binary indicators where $\psi_d = 1$ for a relevant feature and $\psi_d = 0$ otherwise. We take an assumption here that if the feature is irrelevant then it follows a common shared density represented as a Bernoulli/multinomial distribution $q(.|\lambda)$ and is also independent of the class label (Bouguila, 2010; Wang and Kabán, 2005; Silvestre et al., 2013). Now the mixture density is represented by

$$p(\mathbf{x}_n|\pi, \Psi, \mathbf{m}, \lambda) = \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} [p(x_{nd}|m_{kd})]^{\psi_d} [q(x_{nd}|\lambda_d)]^{1-\psi_d}. \tag{5.7}$$

We treat ψ_d as a missing (or latent) variable. We then estimate the feature saliency as $\rho_d = p(\psi_d = 1)$, the probability of the dth feature being relevant. The resulting model can now be represented as

$$p(\mathbf{x}_n|\Omega) = \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} [\rho_d p(x_{nd}|m_{kd})] + [(1-\rho_d) q(x_{nd}|\lambda_d)], \tag{5.8}$$

where $\Omega = (\pi_k, \rho_d, m_{kd}, \lambda_d)$ represents the set of all the parameters of the model. ψ_d is a binary indicator variable so we can re-write $[p(x_{nd}|m_{kd})]^{\psi_d}[q(x_{nd}|\lambda_d)]^{1-\psi_d}$ from equation (5.7) as $\psi_d[p(x_{nd}|m_{kd})] + (1-\psi_d)[q(x_{nd}|\lambda_d)]$ in equation (5.8) (see Appendix C.1 for detailed derivation (this is adapted from the derivation given in (Law et al., 2004))).

The log-likelihood can now be written as

$$\mathcal{L}(\Omega) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\Omega). \tag{5.9}$$

5.4.1 An EM algorithm for LTM-FS

We can exploit the latent structure of LTM, in a similar way as for GTM-FS, to simultaneously estimate feature saliency and estimate the parameters of the model using an *expectation-maximization* (EM) algorithm. We flip a biased coin for each feature where the probability of a head for the dth feature is ρ_d; if we get the head then we consider that the feature is generated from the mixture component $p(.|m_{kd})$ otherwise the component $q(.|\lambda_d)$ is used. We assume that the latent component label, y, is a missing variable and then in the E-step, we use the current parameters, $\mathbf{\Omega}$, to compute posterior probabilities, $r_{nk} = p(y_n = k|\mathbf{x}_n)$, for each nth data point for each of the kth Bernoulli/Multinomial components using Bayes' theorem as,

$$r_{nk} = \frac{\pi_k \prod_{d=1}^{D} \left\{\rho_d p(x_{nd}|m_{kd}) + (1-\rho_d) q(x_{nd}|\lambda_d)\right\}}{\sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} \left\{\rho_d p(x_{nd}|m_{kd}) + (1-\rho_d) q(x_{nd}|\lambda_d)\right\}}. \tag{5.10}$$

We use the responsibility matrix, $\mathbf{R}$, to compute $u_{nkd} = p(\psi_d = 1, y_n = k|\mathbf{x}_n)$ which measures the importance of the dth feature for the nth data point belonging to the kth component and $v_{nkd} = p(\psi_d = 0, y_n = k|\mathbf{x}_n)$

$$u_{nkd} = \frac{\rho_d p(x_{nd}|m_{kd})}{\rho_d p(x_{nd}|m_{kd}) + (1-\rho_d) q(x_{nd}|\lambda_d)} r_{nk}, \tag{5.11}$$

$$v_{nkd} = r_{nk} - u_{nkd}. \tag{5.12}$$

M-Step for Bernoulli case:

In the M-step for the Bernoulli case, we use the simple gradient-based approach used in (Kabán and Girolami, 2001) as an inner loop to update the weights (a detailed derivation is given in Appendix C.1)

$$\Delta \mathbf{w}_d \propto \mathbf{\Phi}^T \left[\mathbf{U}_d \mathbf{x}_d - \mathbf{E}_d g(\mathbf{\Phi} \mathbf{w}_d)\right], \tag{5.13}$$

where $\mathbf{\Phi}$ is a $K \times L$ matrix, $\mathbf{w}_d$ is a $L \times 1$ weight vector, $\mathbf{U}_d$ is a $K \times N$ matrix obtained from equation (5.11), $\mathbf{x}_d$ is a $N \times 1$ data vector, and $\mathbf{E}_d$ is a $K \times K$ diagonal matrix with elements

$$e_{kkd} = \sum_{n=1}^{N} u_{nkd}. \tag{5.14}$$

Once we obtain the re-estimated $\widehat{\mathbf{w}}_d$ for each dth feature from equation (5.13), we can then straightforwardly update $\mathbf{m}_k$ for each Bernoulli component using equation (5.1) as follows,

$$\widehat{\mathbf{m}_k} = g(\mathbf{\Phi}(\mathbf{z}_k)\widehat{\mathbf{W}}), \tag{5.15}$$

and the parameter λ_d of the common shared density for each dth feature can be updated using

$$\widehat{\lambda_d} = \frac{\sum_n (\sum_k v_{nkd} x_{nd})}{\sum_{nk} v_{nkd}}. \tag{5.16}$$

Usually the feature saliency parameter is updated by

$$\widehat{\rho_d} = \frac{\sum_{nk} u_{nkd}}{\sum_{nk} u_{nkd} + \sum_{nk} v_{nkd}}. \tag{5.17}$$

If we take the Beta distribution as a prior (a conjugate prior for Bernoulli distribution) for the feature saliency (the same is used by Bouguila (2010) for clustering binary data with simultaneous feature saliency estimation),

$$p(\rho_d) = \frac{\Gamma(\alpha_d + \beta_d)}{\Gamma(\alpha_d)\Gamma(\beta_d)} \rho_d^{\alpha_d - 1} (1 - \rho_d)^{\beta_d - 1}, \tag{5.18}$$

then the feature saliency measure can be updated using

$$\widehat{\rho_d} = \frac{\max(\sum_{nk} u_{nkd} + \alpha_d - 1, 0)}{\max(\sum_{nk} u_{nkd} + \alpha_d - 1, 0) + \max(\sum_{nk} v_{nkd} + \beta_d - 1, 0)}. \tag{5.19}$$

Here we use hyperparameters (i.e. α, β), for the prior distribution over the feature saliency, both set to be 2 (i.e. the same is used by Bouguila (2010) in the binary data clustering perspective where they also estimate saliencies of features).

M-Step for multinomial case:

The weight sub-matrix (where each sub-matrix represents weights for one encoded feature) for the multinomial case can be updated using

$$\Delta \mathbf{W}_{S_d} \propto \mathbf{\Phi}^T \left[\mathbf{U}_d \mathbf{X}_{S_d} - \mathbf{E}_d g(\mathbf{\Phi} \mathbf{W}_{S_d}) \right], \tag{5.20}$$

where $\mathbf{\Phi}$ is a $K \times L$ matrix, $\mathbf{W}_{S_d}$ is a $L \times S_d$ weight sub-matrix, $\mathbf{U}_d$ is a $K \times N$ matrix obtained from equation (5.11), $\mathbf{X}_{S_d}$ is a $N \times S_d$ data matrix (dth feature encded to 1-of-S_d), and $\mathbf{E}_d$ is a $K \times K$ diagonal matrix with elements

$$e_{kkd} = \sum_{n=1}^{N} u_{nkd}. \tag{5.21}$$

Once we obtain the re-estimated $\widehat{\mathbf{W}_{S_d}}$ for each dth feature from equation (5.20), we can then straightforwardly update $\mathbf{m}_{kS_d}$ for each dth feature of the multinomial component using equation (5.2) as follows,

$$\widehat{\mathbf{m}_{kS_d}} = g(\mathbf{\Phi}(\mathbf{z}_k)\widehat{\mathbf{W}_{S_d}}), \tag{5.22}$$

and the parameter λ_d of the common shared density for each dth feature can be updated using

$$\widehat{\lambda_{S_d}} = \frac{\sum_n (\sum_k v_{nkd} \mathbf{x}_{nS_d})}{\sum_{nk} v_{nkd}}. \tag{5.23}$$

Usually the feature saliency parameter is updated using

$$\widehat{\rho_d} = \frac{\sum_{nk} u_{nkd}}{\sum_{nk} u_{nkd} + \sum_{nk} v_{nkd}}. \tag{5.24}$$

If we take the Dirichlet-type prior (which is a natural conjugate prior for the multinomial) for the feature saliencies (the same is used by Silvestre et al. (2013) for clustering categorical data with simultaneous feature saliency)

$$p(\rho_1, \cdots, \rho_D) \propto \prod_{d=1}^{D} \rho_d^{-\frac{Kc_d}{2}} (1-\rho_d)^{\frac{c_d}{2}}. \tag{5.25}$$

then the feature saliency measure can be updated by

$$\widehat{\rho_d} = \frac{\max\left(\sum_{nk} u_{nkd} - \frac{K(c_d-1)}{2}, 0\right)}{\max\left(\sum_{nk}, u_{nkd} - \frac{K(c_d-1)}{2}, 0\right) + \max\left(\sum_{nk} v_{nkd} - \frac{(c_d-1)}{2}, 0\right)}, \tag{5.26}$$

where c_d represents the number of categories for the dth feature. Algorithm 5.4.1 summarises LTM-FS. Like standard LTM model (as discussed in Section 3.3.6.1), parameters update for the LTM-FS model scales as $\mathcal{O}(LND + LK(N + D + K))$. However, it requires to process an extra loop for the D features to update weight vector $\widehat{\mathbf{w}}_d$ in the parameter learning process.

5.5 Experiments

An LTM-FS was tested on synthetic and real datasets (both for binary and multinomial variables). Visualisation results of LTM-FS are compared with standard LTM. We initialised the weight matrix, $\mathbf{W}$, using principal component analysis (PCA). We used labels for better understanding of the distribution of data points of different classes on the visualisation results. Label information is also used for measuring the visualisation quality using KL divergences and nearest neighbour (NN) classification error. We used 8×8 latent and 4×4 RBF grids both for LTM and LTM-FS model training for binary datasets

Algorithm 5.4.1: LTM-FS algorithm summary

```
Input:Training dataset.
OutPut:Trained LTM-FS visualisation model with feature saliency for all the features.
begin
    Generate the latent grid points $\mathbf{z}_k \in \mathcal{H}$, $k = 1, \cdots, K$;
    Generate the basis function grid, $\mathbf{\Phi}(\mathbf{z}_k)$, centres $\{\nu_l\}$, $l = 1, \cdots, L$;
    Select the basis functions, $\mathbf{\Phi}(\mathbf{z}_k)$;
    Compute the design matrix of basis function activations, $\mathbf{\Phi}$;
    Initialise weight matrix ($\mathbf{W}$), randomly or using PCA;
    Apply the mapping function (using equation (5.1) for binary data or equation (5.2) for multi-categorical data) to initialise means of the mixture components;
    Initialise feature saliency, $\rho_d$, for each $d$th feature, to 0.5;
    Initialise the mixing coefficient, $\pi_k$, with $\frac{1}{K}$ for each $k$th component in the grid;
    Set the initial mean for the shared distribution, $q(.|\lambda)$, as the mean of the data;
    repeat
        E-Step:
        Compute $\mathbf{R}$, $\mathbf{U}$ and $\mathbf{V}$ using equation (5.10)), (5.11) and (5.12), using current parameters, $\Omega$;
        M-Step:
        for d=1 to$D$ do
            repeat
                Re-estimate the weight vector, $\mathbf{w}_d$, (for binary case), using
                $\Delta\mathbf{w}_d \propto \mathbf{\Phi}^T \left[\mathbf{U}_d\mathbf{x}_d - \mathbf{E}_d g(\mathbf{\Phi}\mathbf{w}_d)\right]$, using equation (5.13)
                OR
                Re-estimate the weight matrix, $\mathbf{W}_{\mathbf{s}_d}$, (for multinomial case)
                $\Delta\mathbf{W}_{S_d} \propto \mathbf{\Phi}^T \left[\mathbf{U}_d\mathbf{X}_{S_d} - \mathbf{E}_d g(\mathbf{\Phi}\mathbf{W}_{S_d})\right]$, using equation (5.20);
            until convergence;
        end
        Re-estimate the mean, $\mathbf{m}_k$, for each $k$th component of the mixture using equation (5.15) for Bernoulli case or using equation (5.22) for multinomial case;
        Re-estimate the mean of the shared distribution using equation (5.16) for binary data or equation (5.23) for multi-categorical data;
        Re-estimate the feature saliency, $\rho_d$, using equation (5.19) for binary data or equation (5.26) for multi-categorical data;
    until convergence;
end
```

(both for synthetic and an MHC datasets). We also performed experiments with different latent and RBF grid sizes both for standard LTM and LTM-FS (for binary data case) and observed that the results were consistently similar both in terms of visualisation results and estimated feature saliencies (see Appendix C.3). Whereas in the multinomial case, the LTM-FS is observed to be sensitive to the latent and RBF grid sizes: further details are given in Section 5.6.

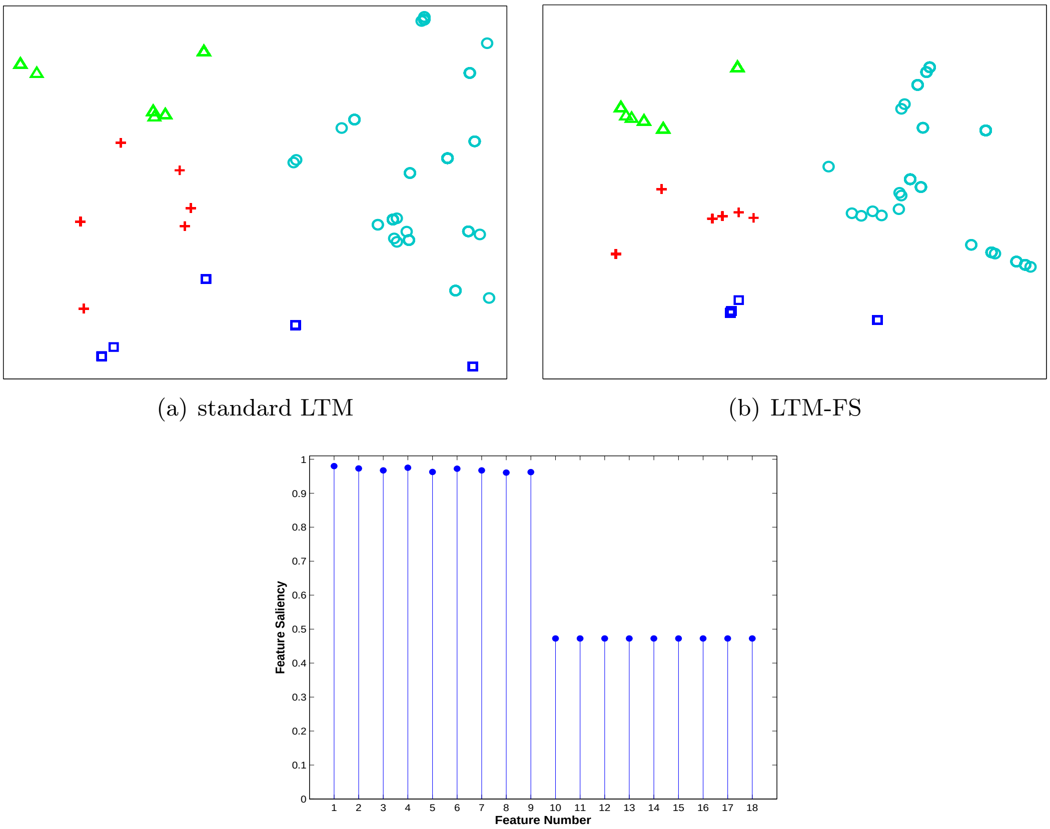
5.5.1 Synthetic binary datasets

We generated synthetic binary datasets with different numbers of feature sets. We use two synthetic binary datasets here to demonstrate LTM-FS: the first dataset is of 18 features with 9 informative features with four true clusters and 9 uninformative features and the second dataset is of 100 features with 40 informative features with four true clusters and 60 uninformative features (a similar synthetic dataset is used by Wang and Kabán (2005)). We randomly added ones of different densities in the uninformative features with different proportions such as no ones, all ones, $0.2, 0.4, 0.6$, and 0.8: i.e. effectively an uncorrelated Bernoulli random variable with $p = 0$, 1, 0.2, 0.4, 0.6, and 0.8 respectively. We also add 1s of 5% of the total number of points (to create a small spread in each class in the data space) in the informative features. We present here one of the results with random ones added in uninformative features whereas results with different densities of randomly added ones are given in Appendix C.4.

Visualisations obtained from standard LTM and LTM-FS are presented in Figures 5.1 and 5.2 for both synthetic binary datasets respectively. Estimated feature saliencies are presented in Figures 5.1(c) and 5.2(c) respectively. Further discussion on results is given in Section 5.6.

5.5.2 Synthetic multi-categorical datasets

We generated a synthetic multi-categorical dataset with three features (where the first feature has 1-to-3 categorical values, the second feature has 1-to-6 categorical values and the third feature has 1-to-9 categorical values). We then generated two sets of synthetic datasets from this dataset by adding different numbers of noisy features (in the first dataset we added two noisy (uninformative) features with a random distribution of 1-to-3 categorical values (for the first noisy feature) and 1-to-6 categorical values (for the second noisy feature)) whereas for the second dataset we added one more noisy feature in the first dataset with 9 randomly distributed categorical values. Similar synthetic datasets have previously been used by Silvestre et al. (2013) to demonstrate estimating feature saliencies



(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

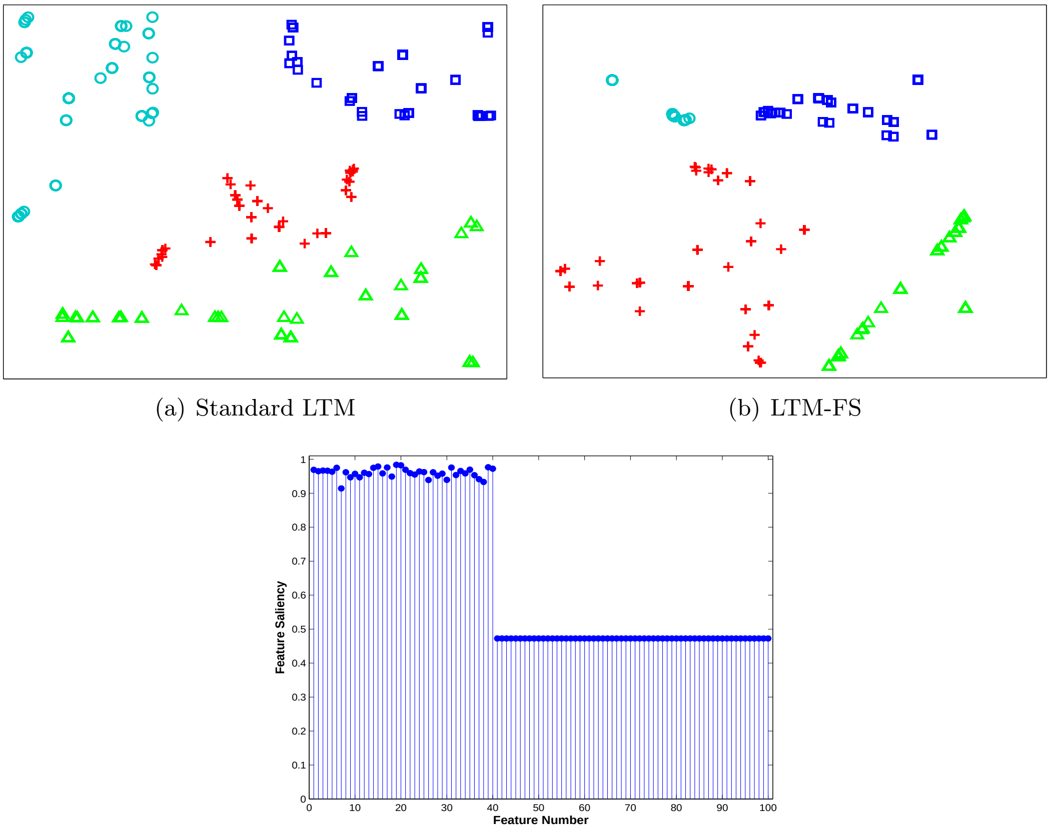
Figure 5.1: The LTM and the LTM-FS visualisations of the binary synthetic dataset-I. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

with simultaneous clustering using mixtures of multinomial distributions.

Visualisations obtained from LTM and LTM-FS, using a mixture of multinomials for modelling the noise, are presented in Figures 5.3 and 5.4 respectively. We also present the comparative evaluation of the visualisations in Table 5.1. Estimated feature saliencies for both datasets are presented in Figures 5.3(c) and 5.4(c) respectively. Further discussion of the results is given in Section 5.6.

5.5.3 Major histocompatibility complexity class-I binary dataset

To demonstrate the effectiveness of our proposed approach, we used a real binary dataset generated from the primary sequences of protein family of Major Histocompatibility Complex (MHC) for class-I of humans (also known as Human Leukocyte Antigen (HLA)). The MHC class-I sequences has already been divided into three classes (e.g. HLA-A, HLA-B and HLA-C) based on a similarity measure of the sequences. As explained in Chapter 2, we used 3840 aligned sequence's (1236 for HLA-A, 1777 for HLA-B and 827 for HLA-C) with



(a) Standard LTM

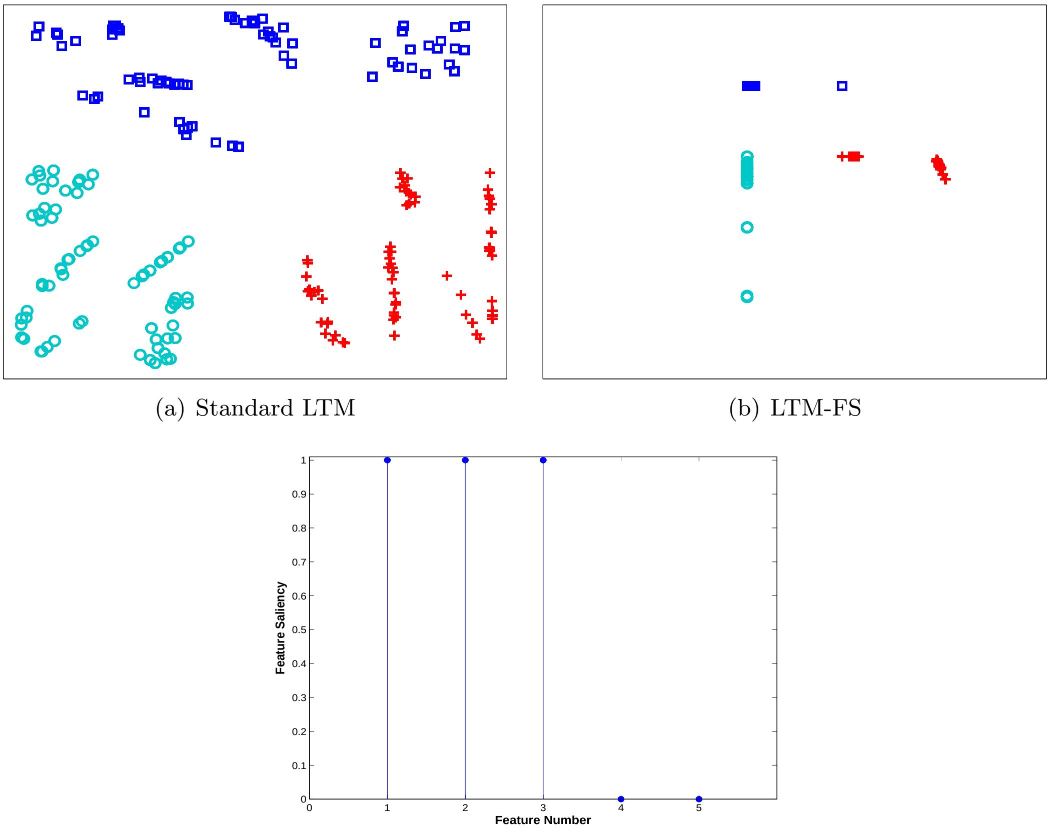
(b) LTM-FS

(c) Estimated saliencies

Figure 5.2: The LTM and the LTM-FS visualisations of the binary synthetic dataset-II. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

most conserved region over 182 length of amino acids for each class to generate a binary dataset. We initially generated a consensus sequence based on the maximum occurrence of amino acids at certain position in all the aligned sequences. Each aligned sequence is then matched with a consensus sequence to generate binary dataset by putting 1 if the amino acid of the target sequence is matched with the corresponding positioned amino acid in the consensus sequence and 0 otherwise.

Visualisations obtained using LTM and LTM-FS, with a mixture of Bernoulli distributions for modelling the noise, are presented in Figure 5.5. Estimated feature saliencies are presented in Figure 5.5(c). To demonstrate the effectiveness of the feature saliency estimation approach, we re-trained the visualisation model with the selected features only (with saliencies $>$ 0.5 in Figure 5.5(c)) both with LTM and LTM-FS (see Figure 5.6). Further discussion of the results is given in Section 5.6.



(a) Standard LTM

(b) LTM-FS

(c) Estimated saliencies

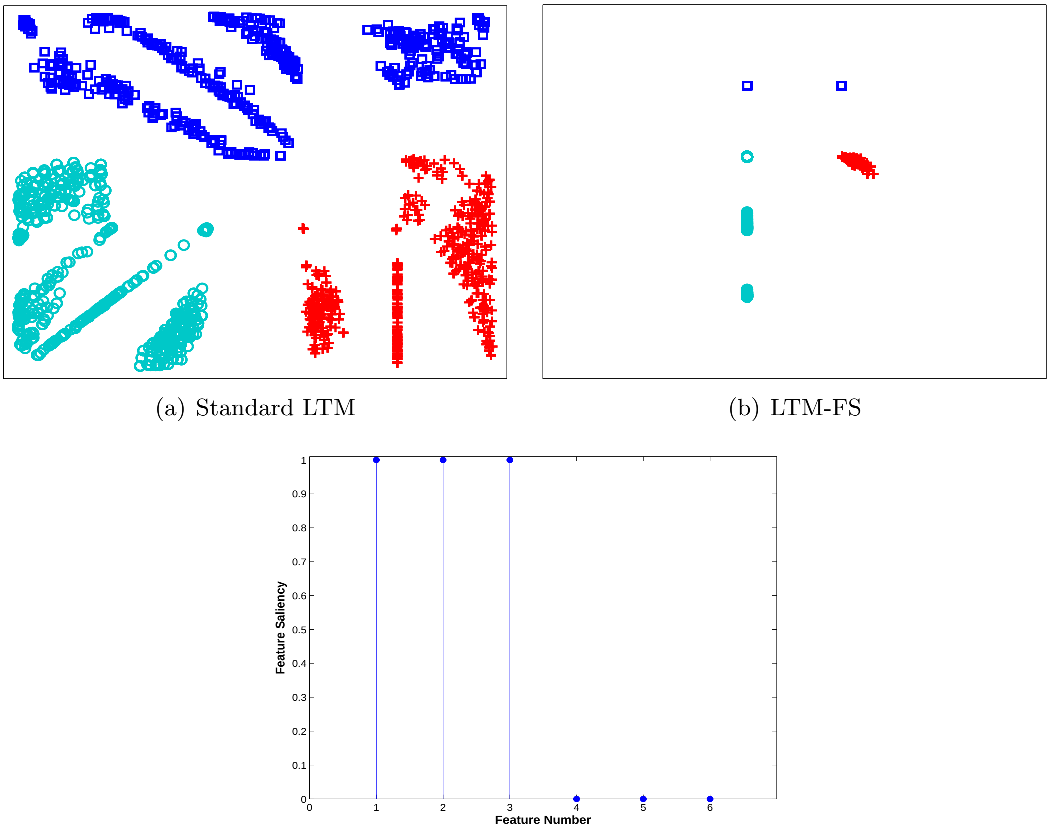
Figure 5.3: The LTM and the LTM-FS visualisations of the multi-category synthetic dataset-I. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

5.5.4 Wisconsin breast cancer categorical dataset

To demonstrate LTM-FS for multi-category data we used the Wisconsin breast cancer dataset downloaded from the UCI data repository (Bache and Lichman, 2013). The dataset contains 699 records, but 16 records have missing values. For simplicity, we ignored those records containing missing values, leaving 683 records consisting of 444 benign patients and 239 malignant patients. The dataset has 10 categorical features[1] where each feature can have up to ten categories. We use 1-of-S_d binary encoded scheme for each of the dth categorical feature.

Visualisations obtained from LTM and LTM-FS, with the assumption of mixture of multinomial for modelling the noise, are presented in Figure 5.7. Estimated feature saliencies are presented in Figure 5.7. Further discussion on results is given in Section 5.6.

[1] List of features for Wisconsin breast cancer dataset: Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli and Mitoses



(a) Standard LTM

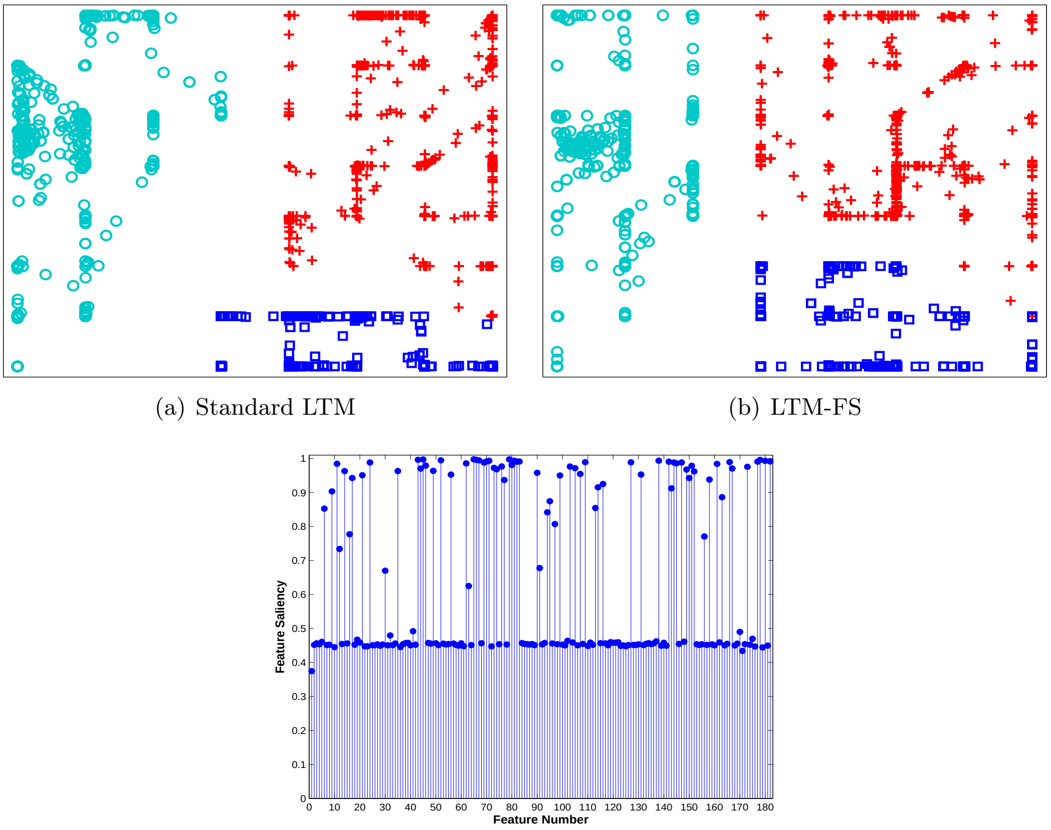
(b) LTM-FS

(c) Estimated saliencies

Figure 5.4: The LTM and the LTM-FS visualisations of the multi-category synthetic dataset-II. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

Datasets		**KL Divergence**	
		LTM	**LTM-FS**
Binary	Synthetic-I	334.1249	**415.2572**
	Synthetic-II	282.9710	**334.8833**
	MHC Class-I	178.6538	**198.0998**
Multinomial	Synthetic-I	96.2732	**144.1746**
	Synthetic-II	105.5500	**143.2366**
	Breast Cancer Wisconsin	**31.9634**	26.4662

Table 5.1: Visualisation quality evaluation metrics of KL divergence to compare the LTM and the LTM-FS models. The higher the value of the KL divergence the better the separation between the classes on the visualisation plots. Generally the LTM-FS model have shown better the KL divergence values except for the breast cancer wisconsin dataset.

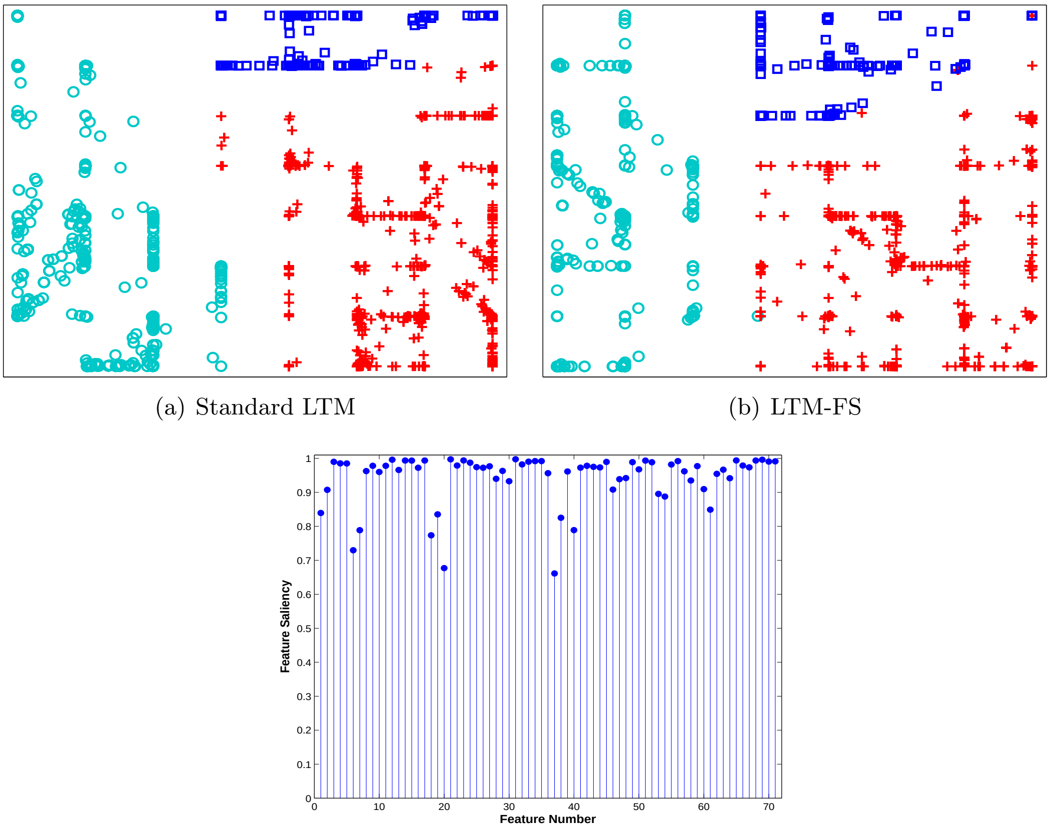


(a) Standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure 5.5: The LTM and the LTM-FS visualisations of the MHC class-I sequence-based binary dataset. The data points shown as cyan circles represent alleles of the HLA-A, red plus signs for the HLA-B and blue squares for the HLA-C. Both the LTM and the LTM-FS visualisations (i.e. (a) and (b) respectively) have shown clear separation between three classes of the MHC class-I hence it is difficult to visually observe better clustering structure from both. Feature saliencies estimated from the LTM-FS are shown in (c) on a scale of 0-to-1.

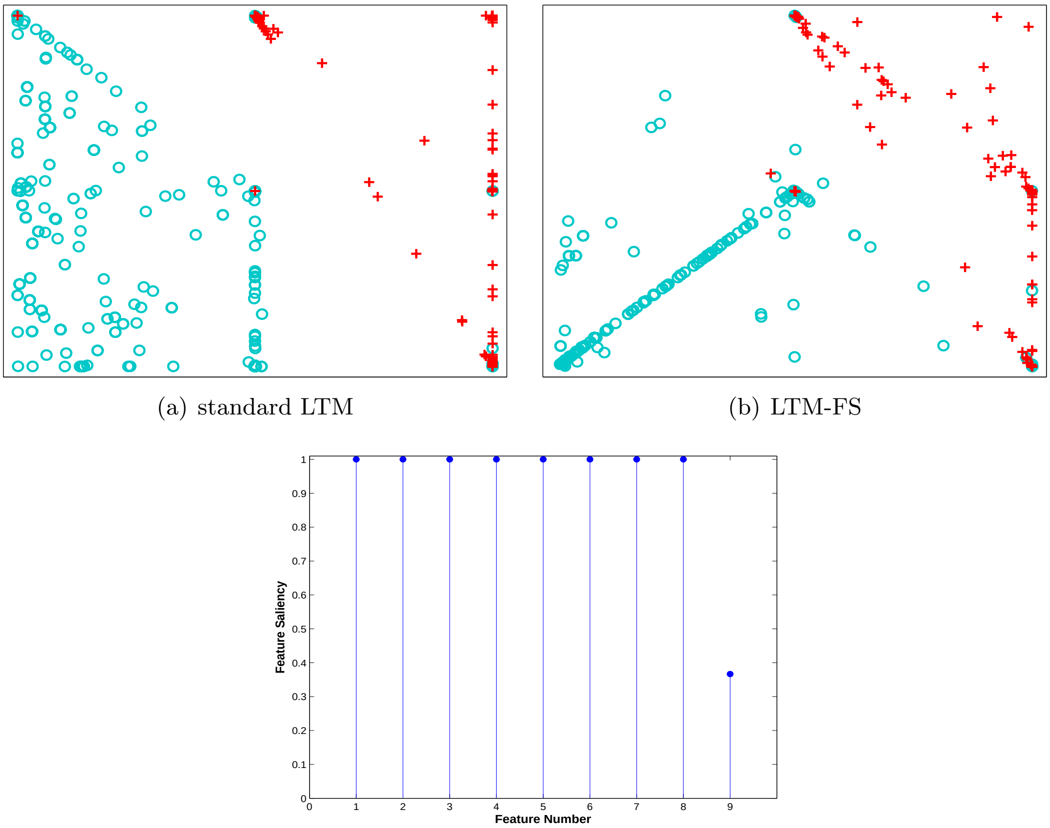


(a) Standard LTM

(b) LTM-FS

(c) Estimated Saliences

Figure 5.6: The LTM and the LTM-FS visualisations for the MHC class-I, sequence-based binary dataset, using selected features with estimated saliencies>= 0.5 from the saliencies shown in Figure 5.5(c)). In (a) and (b) both visualisation methods have also shown clear separation between three classes of the MHC using only 82 selected features instead of 182 features suggesting small set of features is good enough to get better separation between the classes of MHC class-I genes. The (c) show estimated saliencies from the LTM-FS using this small number of selected features. Legend same as in Figure 5.5.



(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure 5.7: The LTM and the LTM-FS visualisations of the breast cancer multi-category dataset. The data points showed as cyan circles represent benign and red plus for malignant. The visualisation results both from the LTM and the LTM-FS are not significantly different because saliencies estimated shown in (c) from LTM-FS for all the features are higher except for one feature (i.e. the 9th feature which is ‘Mitosis’) which has shown lower saliency value.

5.6 Discussion

Visualisations of the synthetic binary and multi-category datasets using both LTM and LTM-FS visualisation algorithms have shown clear separation among the four classes of data. However, LTM-FS based visualisations have shown more compact clusters (for example see sub-figures c and d of Figures 5.1, 5.2, 5.3, 5.4) (with less spread of data points for each class) compared to LTM visualisation (for example see sub-figures a and b of Figures 5.1, 5.2, 5.3, 5.4) (where the data points have more spread for each class). The greater compactness in LTM-FS visualisations is because of the separate shared multivariate distribution, $q(.|\lambda)$, that models the noise in the data. The other potential advantage of using LTM-FS over standard LTM is its ability to estimate feature saliencies while training a visualisation model. This suggests that LTM-FS not only gives more compact clusters appearing on visualisations but also estimates feature saliencies which are valuable for improving our understanding of the data. For both types of synthetic datasets (i.e. binary and multi-categorical), we also observed from the visualisation quality measure of KL divergence (see Table 5.1) that the LTM-FS have shown better results. For synthetic binary and multi-category datasets (where the data is generated from four distinct classes), nearest neighbour classification error was computed and observed to be usually zero both for LTM and LTM-FS.

Visualisation of the MHC class-I binary dataset using LTM-FS was also better than LTM in terms of the KL-divergence quality measure. For the MHC class-I binary dataset, NN classification error, both for LTM and LTM-FS was computed and observed to be 1 usually. The potential advantage of using LTM-FS over LTM is its capability of estimating feature saliencies while training a visualisation model. We re-trained both LTM and LTM-FS with the selected subset of features (with estimated saliencies greater than 0.5 in Figure 5.5(c)). The visualisations using this selected subset of features have shown the fact that we can still get better separation among clusters of three classes (as shown in Figure 5.6(b)). This suggests that small sets of features are good enough to get effective visualisations of the dataset in terms of separation between the classes.

The LTM-FS model for multi-categorical data is observed to be very sensitive to both the latent and RBF grid sizes. For synthetic categorical datasets, for comparison purpose both for LTM and LTM-FS, we used a 6×6 latent and 4×4 RBF grid. LTM-FS results are encouraging for synthetic multi-categorical datasets (with respect to visualisation results and also in terms of visualisation quality measure of KL divergences (see Table 5.1) and NN-classification), whereas for the real dataset it still faces some challenges for the selection of latent and RBF grid sizes. For the breast cancer dataset, we used a latent grid of size 3×3 and an RBF grid of 5×5. However for the real multi-categorical dataset, results were better with the standard LTM compared to LTM-FS in terms of visualisation quality measures of KL divergence (see Table 5.1) and NN classification error (for LTM 29 and for LTM-FS 37). Visualisation results both for LTM and LTM-FS are not very much different and estiamted saliencies using LTM-FS are higher except for one feature (i.e. Mitosis) (see Figure 5.7(c)).

5.7 Conclusion

Analysing multivariate large high-dimensional discrete datasets is a challenging task in the presence of inherent noise. Usually both data visualisation and feature selection are used as two separate approaches in proteomic, genomics and other bioinformatics areas. Joining both the approaches in a single framework is logical and can play an important role as the benefits from one technique can be exploited for the other technique to get more useful understanding.

We were successful in modifying a feature selection approach for the unsupervised task of visualising a multivariate discrete dataset onto a low-dimensional latent space with the assumption that the data is generated from probabilistic mixture models. Our proposed algorithm, LTM-FS, not only attempts to improve the visualisation quality by modelling the irrelevant noisy features as a shared distribution but also estimates saliencies for all the features which help the user to understand the importance of each feature. LTM-FS results for binary datasets (both synthetic and real) and synthetic multi-category results are satisfactory and encouraging whereas for real multi-category datasets the proposed algorithm still faces some challenges.

6 Mixed-Type Data Visualisation and Simultaneous Feature Selection

CONTENTS

6.1 Introduction

Type-specific data analysis has been well studied in the machine learning[1]. In the recent couple of decades, the need for analysing mixed-type data is gaining a lot of attention from machine learning experts because of the fact that real world processes often generate a data of mixed-type. An example of such a mixed-type data could be a hospital's patients' dataset where typical fields include age (discrete or continuous), gender (binary), test results (binary or continuous), height (continuous) etc.

In practice a number of ad-hoc solutions are used to handle mixed-type data in analysis. For example, if there is a mixture of continuous and discrete variables, then either all the discrete variables are converted to some numerical scoring equivalent or on the other hand all the continuous variables are considered as discrete variables adopting some grouping criteria. Alternatively, both types of variables are analysed separately and then the results are combined using some criteria. According to Krzanowski (1983), "All these options involve some element of subjectivity, with possible loss of information, and do not appear very satisfactory in general". The ideal general solution for analysing such heterogeneous data is to specify a model that builds a joint distribution with the assumption of an appropriate noise distribution for each type of feature set (for example a Bernoulli distribution for modelling binary features, a multinomial distribution for modelling multi-category features and a Gaussian distribution for modelling continuous features) and then fitting the model to data where the parameter estimates are used to draw inferences (de Leon and Chough, 2013).

In the literature there is no multivariate distribution that can model random variables of different types. However, one possible way of jointly modelling discrete and continuous features is using a latent variable approach to understand the correlation between features of different type in combination. For example, a dataset consisting of continuous, binary and categorical features can be modelled in a latent variable model using the conditional distribution as a product of Gaussian, Bernoulli and multinomial distributions in the data space. Such a latent variable model follows a conditional independence criterion for each type of the observed variable to find a correlation between observed features and latent variables in a unified framework.

A generative topographic mapping (GTM) is a probabilistic non-linear visualisation model for modelling the relationship between observed continuous features and continuous latent variables. As explained earlier in Section 3.3.6, as a generalisation of GTM, a latent

[1] http://letdataspeak.blogspot.co.uk/2012/07/mixed-type-data-analysis-i-overview.html

variable model called the latent trait model (LTM) was proposed with a goal of handling different types of data by considering the exponential family of distributions (Kabán and Girolami, 2001). However an LTM model was proposed to handle only one variable type in a given dataset with a main focus on discrete type data only whereas we in this chapter propose to extend such a model to handle mixed-type data (i.e. a dataset with variables of more than one type).

Influenced from the generalisation of GTM given by Kabán and Girolami (2001), we propose here a probabilistic non-linear latent variable model by combining a generative topographic mapping (GTM) (appropriate for continuous data) and a latent trait model (LTM) (appropriate for discrete data) in a principled way to visualise mixed-type data on a single continuous latent space under a unified proposed framework of conditional independence criteria: we shall refer to this model a generalised GTM (GGTM).

We also propose an extension of GGTM to simultaneously estimate feature saliency (we call it as GGTM-FS). The potential advantage of GGTM-FS is to improve the latent space visualisation of mixed-type datasets by reducing the impact of noisy features using a probabilistic approach as discussed in chapters 3, 4 and 5 (where we apply a similar approach to estimate feature saliency while visualising one type of data (i.e. continuous or binary or multi-category)): this extension also gives feature saliency values which can be used to understand the importance of each feature in the dataset (for more informative clusters).

6.2 Related Work

Bishop and Svensen (1998); Bishop et al. (1998); Tipping (1999) discuss the idea that if the multivariate dataset contains mixed-types then it can be modelled with GTM-like (Bishop and Svensen, 1998; Bishop et al., 1998) and Probabilistic PCA-like latent variable models. However, these papers did not implement any models and argued that this can be achieved by just taking the product of likelihoods based on the conditional independence assumption for the latent variable model formalism. Our interest here is mainly to use a GTM-like visualisation non-linear model for this purpose. We briefly explain the evolution of GTM-like models in the following paragraph.

Initially GTM was developed with the assumption of a mixture of Gaussian distributions as a generative model of continuous data. Later on, Kabán and Girolami (2001) proposed a generalisation of GTM in order to make it more general by using an appropriate noise model based on the type of the dataset. They derived a general EM algorithm based on the fact that the noise model is taken from the exponential family of distributions. For discrete datasets they used a Bernoulli distribution for binary features and a multinomial distribution for multi-category features with a one-of-N encoding. All these variants of GTM are only able to model one type of feature at a time in the latent variable model framework. Kabán and Girolami (2001) also discussed a possible way of handling mixed-type data using the approach discussed and suggested by Bishop and Svensen (1998); Bishop et al. (1998); Tipping (1999). However, this idea of handling mixed-type dataset with GTM like non-linear models has been only discussed but not implemented and tested.

However, about a decade ago this idea was implemented with a linear model by Yu and Tresp (2004) to visualise a mix of continuous and binary features data on a single continuous latent space by extending probabilistic principal component analysis (PPCA). He called this model a generalised PPCA (GPPCA): it was influenced by the generalisation of PPCA to binary data (Tipping, 1999). GPPCA is a linear probabilistic model and uses a variational EM algorithm for parameter estimation, whereas our proposed probabilistic latent variable model is a non-linear variant and we use a variant of the EM for parameter estimation (similar to that proposed by Kabán and Girolami (2001)). In the literature there are few other latent variable models for mixed-type datasets but as of our knowledge they are all linear models (like GPPCA) (Moustaki, 1996; Sammel et al., 1997; Dunson, 2000) and they either use numerical integration or a sampling approach to handle the intractable integration for fitting a latent variable model of this type.

As discussed in the related work section of Chapter 5, feature saliency estimation is usually applied to clustering algorithms (for example, Bouguila (2010) discusses the possibility of computing feature saliency while clustering data vectors with a mix of continuous and discrete variables but did not implement it) and only a few attempts have been made in general to apply simultaneous feature saliency to the latent variable model framework and they are usually with the Gaussian assumption to hanlde only continuous type datasets. We proposed, in Chapter 5, a combined parameter-learning and feature-saliency estimation algorithm for discrete type datasets (where we used a Bernoulli noise model for binary features and a multinomial noise model for multi-category features) and have shown its effectiveness on synthetic and real datasets.

To the best of our knowledge, there is no similar approach in the literature for estimating feature saliency when modelling the data with joint probability distribution both for clustering and latent variable model framework (and we have also not found any discussion of latent variables models for mixed-type data with simultaneous feature saliency estimation).

6.3 A Generalised Generative Topographic Mapping (GGTM) Model

The main goal of a latent variable model is to find a low dimensional manifold, $\mathcal{H}$, with M-dimensions (usually $M = 2$) for the distribution $p(\mathbf{x})$ of high-dimensional data space, $\mathcal{D}$, with D-dimensions. Latent variable models like GTM (appropriate for continuous datasets) or LTM (appropriate for discrete datasets), have been developed to handle a dataset where all the features are of the same type (either continuous or binary or multi-categorical).

We consider here the task of modelling a D-dimensional data space defined by $|\mathcal{R}|$ continuous, $|\mathcal{B}|$ binary and/or $|\mathcal{C}|$ multi-categorical (where for each $d \in |\mathcal{C}|$, we use 1-of-S_d encoded binary features) number of features respectively. The link functions defined for mapping between the latent space and the data space for continuous, binary and multi-category features are defined in equations (6.1), (6.2) and (6.3) respectively

$$\mathbf{m}^{\mathcal{R}} = \mathbf{\Phi}(\mathbf{z})\mathbf{W}^{\mathcal{R}}. \tag{6.1}$$

$$\begin{aligned}\mathbf{m}^{\mathcal{B}} &= g^{\mathcal{B}}(\mathbf{\Phi}(\mathbf{z})\mathbf{W}^{\mathcal{B}}) \\ &= \frac{\exp(\mathbf{\Phi}(\mathbf{z})\mathbf{W}^{\mathcal{B}})}{1+\exp(\mathbf{\Phi}(\mathbf{z})\mathbf{W}^{\mathcal{B}})}.\end{aligned} \tag{6.2}$$

$$\begin{aligned}m^{\mathcal{C}}_{s_d} &= g^{\mathcal{C}}(\mathbf{\Phi}(\mathbf{z})\mathbf{w}^{\mathcal{C}}_{s_d}) \\ &= \frac{\exp(\mathbf{\Phi}(\mathbf{z})\mathbf{w}^{\mathcal{C}}_{s_d})}{\sum_{s'_d=1}^{S_d}\exp(\mathbf{\Phi}(\mathbf{z})\mathbf{w}_{s'_d})}.\end{aligned} \tag{6.3}$$

These link functions map a latent point $\mathbf{z} \in \mathcal{H}$ to the point in the data space (i.e. $\mathbf{m}^{\mathcal{R}}$ is a centre of Gaussian (appropriate for continuous features), $\mathbf{m}^{\mathcal{B}}$ is a centre of Bernoulli (which is the probability of weighted sum of success in case of binary-typed data), and $\mathbf{m}^{\mathcal{C}}$ is centre of multinomial (which is the probability of weighted sum of relating to each category in case of multi-category type data) in the data space respectively). We write each observation vector, $\mathbf{x}_n$ in terms of sub-vectors $\mathbf{x}_n^{\mathcal{R}}$, $\mathbf{x}_n^{\mathcal{B}}$ and $\mathbf{x}_n^{\mathcal{C}}$ for continuous, binary and multi-category features respectively. In the rest of this chapter we use superscript $\mathcal{R}$ for continuous features, superscript $\mathcal{B}$ for binary features and superscript $\mathcal{C}$ for categorical features representation. The likelihood of each type subset of features (where each subset take one type of feature) in an observation given the latent variables and the model parameters, based on the corresponding distributional assumption, are given in equations (6.4), (6.5) and (6.6) respectively as,

$$\begin{aligned} p(\mathbf{x}_n^{\mathcal{R}}|\mathbf{z}, \mathbf{W}^{\mathcal{R}}, \beta) &= p(\mathbf{x}_{\mathbf{n}}^{\mathcal{R}}|\mathbf{m}^{\mathcal{R}}, \beta) \\ &= \left(\frac{\beta}{2\pi}\right)^{\frac{|\mathcal{R}|}{2}} \exp\left(-\frac{\beta}{2}||\mathbf{m}^{\mathcal{R}} - \mathbf{x}_n^{\mathcal{R}}||^2\right). \end{aligned} \tag{6.4}$$

$$\begin{aligned} p(\mathbf{x}_n^{\mathcal{B}}|\mathbf{z}, \mathbf{W}^{\mathcal{B}}) &= p(\mathbf{x}_{\mathbf{n}}^{\mathcal{B}}|\mathbf{m}^{\mathcal{B}}) \\ &= \prod_{d=1}^{|\mathcal{B}|} \left(m_d^{\mathcal{B}}\right)^{x_{nd}^{\mathcal{B}}} \left(1 - m_d^{\mathcal{B}}\right)^{(1-x_{nd}^{\mathcal{B}})}. \end{aligned} \tag{6.5}$$

$$\begin{aligned} p(\mathbf{x}_n^{\mathcal{C}}|\mathbf{z}, \mathbf{W}^{\mathcal{C}}) &= p(\mathbf{x}_{\mathbf{n}}^{\mathcal{C}}|\mathbf{m}^{\mathcal{C}}) \\ &= \prod_{d=1}^{|\mathcal{C}|} \prod_{s_d=1}^{S_d} \left(m_{s_d}^{\mathcal{C}}\right)^{x_{ns_d}^{\mathcal{C}}}. \end{aligned} \tag{6.6}$$

For modelling the mixed-type datasets under the latent variable formulism, we compute the product of the likelihoods of Gaussian (equation (6.4)), Bernoulli (equation (6.5)) and multinomial (equation (6.6)) distributions and then the data distribution of, $\mathbf{x}$, with the given weight matrix, $\mathbf{W}$, can be achieved taking an integration over the distribution of latent variables, $\mathbf{z}$, as

$$\begin{aligned} p(\mathbf{x}|\Omega) = \int & p(\mathbf{x}_{\mathbf{n}}^{\mathcal{R}}|\mathbf{z}, \mathbf{W}^{\mathcal{R}}, \beta) \\ & p(\mathbf{x}_{\mathbf{n}}^{\mathcal{B}}|\mathbf{z}, \mathbf{W}^{\mathcal{B}}) p(\mathbf{x}_{\mathbf{n}}^{\mathcal{C}}|\mathbf{z}, \mathbf{W}^{\mathcal{C}}) p(\mathbf{z})\, d\mathbf{z}. \end{aligned} \tag{6.7}$$

where $\Omega = (\mathbf{W}^{\mathcal{R}}, \beta, \mathbf{W}^{\mathcal{B}}, \mathbf{W}^{\mathcal{C}})$ is a set containing all the parameters of the model. We consider prior distribution, $p(\mathbf{z})$, as a sum of delta functions on the nodes of regular grid in latent space (the same prior distribution was used for standard GTM (Bishop and Svensen, 1998) and LTM (Kabán and Girolami, 2001)) as

$$p(\mathbf{z}) = \frac{1}{K} \sum_{k=1}^{K} \delta(\mathbf{z} - \mathbf{z}_k). \tag{6.8}$$

The data distribution can now be defined from equations (6.7) and (6.8) (where mixing co-efficient are taken as fixed for all components (i.e. $\pi_k = \frac{1}{K}$)),

$$p(\mathbf{x}|\Omega) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\mathbf{z}_k, \Omega). \tag{6.9}$$

The log-likelihood now takes the form

$$\mathcal{L}(\Omega) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k p(\mathbf{x}_n | \mathbf{z}_k, \Omega). \tag{6.10}$$

6.3.1 Distributional Assumptions over the Noise Model

The distributional assumption over the noise model is important for the type of the data the model can handle and also for the link function required for modelling different types of data (Kabán and Girolami, 2001). It has been argued in (Kabán and Girolami, 2001; Bishop et al., 1998) that modelling continuous data can be achieved with the assumption of noise as independent and identically distributed (i.i.d.) Gaussian, that gives a tractable analytical solution, which is considered not to be suitable for the discrete type datasets. For example for the Gaussian case that is appropriate for the continuous feature set, the link function is considered as a linear regression function of the latent vectors with weight matrix (see Equation (6.1)). However for the purpose of simplicity and generality, an exponential family of distributions is assumed for noise modelling purpose to handle different type of features in a dataset during the derivation of the LTM algorithm (a model developed with a main focus on datasets with discrete type features). The similar idea of using exponential family of distributions is adopted here for mixed-type data modelling under the latent variable framework (where we apply the same for each type subset of features (i.e. $\mathbf{x}^{\mathcal{R}}$ or $\mathbf{x}^{\mathcal{B}}$ or $\mathbf{x}^{\mathcal{C}}$)). For the purpose of simplicity we use $\mathbf{x}^{\mathcal{M}}$ where superscript $\mathcal{M}$ can be replaced with either $\mathcal{R}$ or $\mathcal{B}$ or $\mathcal{C}$ to indicate type of subset of features for a data point $\mathbf{x}$. The functional form of the exponential family of distributions can be defined by

$$p_{\mathcal{G}}(\mathbf{x}^{\mathcal{M}} | \theta^{\mathcal{M}}) = \exp\left\{ \mathbf{x}^{\mathcal{M}} \theta^{\mathcal{M}} - \mathcal{G}\left(\theta^{\mathcal{M}}\right) \right\} p_0(\mathbf{x}^{\mathcal{M}}). \tag{6.11}$$

In our case, the conditional probability distribution of a data point $\mathbf{x}_n^{\mathcal{M}}$ given latent point $\mathbf{z}_k$ and a weight matrix $\mathbf{W}^{\mathcal{M}}$ can be defined as,

$$\begin{aligned} p_{\mathcal{G}}(\mathbf{x}_n^{\mathcal{M}} | \mathbf{z}_k, \mathbf{W}^{\mathcal{M}}) = \ & \exp\left\{ \mathbf{x}_n^{\mathcal{M}} \mathbf{\Phi}(\mathbf{z}_k) \mathbf{W}^{\mathcal{M}} - \mathcal{G}\left(\mathbf{\Phi}(\mathbf{z}_k) \mathbf{W}^{\mathcal{M}} \right) \right\} \\ & p_0(\mathbf{x}_n^{\mathcal{M}}), \end{aligned} \tag{6.12}$$

where $\mathcal{G}(.)$ is the cumulant function and is defined as

$$\mathcal{G}\left(\mathbf{\Phi}(\mathbf{z}_k) \mathbf{W}^{\mathcal{M}} \right) = \ln\left(\int \exp(\mathbf{x}^{\mathcal{M}} \mathbf{\Phi}(\mathbf{z}_k) \mathbf{W}^{\mathcal{M}}) p_0(\mathbf{x}^{\mathcal{M}}) \, d\mathbf{x}^{\mathcal{M}} \right). \tag{6.13}$$

The *natural parameter* $\theta^{\mathcal{M}}$ of the exponential family of the distribution is taken to be a linear mixing of the latent vectors with respect to the weight matrix $\mathbf{W}^{\mathcal{M}}$,

$$\theta_k^{\mathcal{M}} = \mathbf{\Phi}(\mathbf{z}_k)\mathbf{W}^{\mathcal{M}}, \tag{6.14}$$

where $\mathbf{W}^{\mathcal{M}}$ is the weight matrix of size $L \times |\mathcal{M}|$. The gradient of the cumulant function with respect to the natural parameter (i.e. $\mathbf{\Phi}(\mathbf{z}_k)\mathbf{W}^{\mathcal{M}}$) is

$$\mathbf{m}_k^{\mathcal{M}} = g^{\mathcal{M}}(\mathbf{\Phi}(\mathbf{z}_k)\mathbf{W}^{\mathcal{M}}) = \nabla_{\theta_k^{\mathcal{M}}}\mathcal{G}(\mathbf{\Phi}(\mathbf{z}_k)\mathbf{W}^{\mathcal{M}}), \tag{6.15}$$

where ∇ represents the gradient operation and the function $g(.)$ is the link function (Kabán and Girolami, 2001).

With the first moment identity for the log-likelihood functions applied to the exponential family of distribution (Barndorff-Nielsen, 1978), we know that $\mathbf{m}_k^{\mathcal{M}}$ represents the mean of the kth class under the distribution $p(\mathbf{x}_n^{\mathcal{M}}|\mathbf{z}_k)$,

$$\mathbf{m}_k^{\mathcal{M}} = E\{\mathbf{x}^{\mathcal{M}}|\mathbf{z}_k\}, \tag{6.16}$$

where $\mathbf{x}^{\mathcal{M}}$ represent the observation vectors and $E\{.\}$ represent the expectation operator. We also know that the second moment identity functions for the log-likelihood functions applied to the exponential family distributions explains that the expected value of the Hessian of the cummulant function with respect to the natural parameters (i.e. $\theta_k^{\mathcal{M}} = \mathbf{\Phi}(\mathbf{z}_k)\mathbf{W}^{\mathcal{M}}$) represents the covariance matrix of the kth class for the distribution (i.e. $p(\mathbf{x}^{\mathcal{M}}|\mathbf{z}_k)$), (i.e. the Fisher information matrix). Such a matrix is represented as $\mathbf{G}_k^{\mathcal{M}}$,

$$\mathbf{G}_k^{\mathcal{M}} = \nabla_{\theta_k^{\mathcal{M}}}\mathbf{m}_k^{\mathcal{M}} = Var\{\mathbf{x}^{\mathcal{M}}|\mathbf{z}_k\}, \tag{6.17}$$

6.3.2 An expectation maximization (EM) algorithm for GGTM

Our proposed model is based on mixture distributions where each component is a product of Gaussian, Bernoulli and/or multinomial distribution. Parameters of the mixture model can be determined using an *expectation-maximization* (EM) algorithm.

An EM algorithm can be formulated as: in the **E-step**, we use the current parameter set, Ω, to compute the posterior probabilities (i.e. responsibilities) of each latent space component for each of the nth data point using Bayes' theorem as,

$$r_{kn} = p(\mathbf{z}_k|\mathbf{x}_n, \mathbf{W}) = \frac{\pi_k p(\mathbf{x}_n|\mathbf{z}_k, \mathbf{W})}{\sum_{k'}^{K} \pi_{k'} p(\mathbf{x}_n|\mathbf{z}_{k'}, \mathbf{W})}, \tag{6.18}$$

where

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{z}_k, \mathbf{W}) =& p(\mathbf{x}_\mathbf{n}^{\mathcal{R}}|\mathbf{z}_k, \mathbf{W}^{\mathcal{R}}, \beta) \\ & p(\mathbf{x}_\mathbf{n}^{\mathcal{B}}|\mathbf{z}_k, \mathbf{W}^{\mathcal{B}}) p(\mathbf{x}_\mathbf{n}^{\mathcal{C}}|\mathbf{z}_k, \mathbf{W}^{\mathcal{C}}). \end{aligned} \tag{6.19}$$

In the context of EM algorithm methodology, we can use the maximization of the relative likelihood (McLachlan and Krishnan, 1997; Kabán and Girolami, 2001; Bishop, 1995b; Dempster et al., 1977) (instead of maximizing the log-likelihood) which does not require the computation of the log of a sum. The relative likelihood between old and new set of parameters can be calculated as,

$$\begin{aligned} Q &= \sum_{n=1}^{N}\sum_{k=1}^{K} r_{kn} \log \left\{ p(\mathbf{x}_n|\mathbf{z}_k, \mathbf{W}) p(\mathbf{z}_k) \right\} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} r_{kn} \left\{ \begin{aligned} & \left\{ \mathbf{x}_n^{\mathcal{R}} \theta_k^{\mathcal{R}} - \mathcal{G}\left(\theta_k^{\mathcal{R}}\right) + \log(p_0(\mathbf{x}_n^{\mathcal{R}})) \right\} \\ & + \left\{ \mathbf{x}_n^{\mathcal{B}} \theta_k^{\mathcal{B}} - \mathcal{G}\left(\theta_k^{\mathcal{B}}\right) + \log(p_0(\mathbf{x}_n^{\mathcal{B}})) \right\} \\ & + \left\{ \mathbf{x}_n^{\mathcal{C}} \theta_k^{\mathcal{C}} - \mathcal{G}\left(\theta_k^{\mathcal{C}}\right) + \log(p_0(\mathbf{x}_n^{\mathcal{C}})) \right\} \\ & + \left\{ \log(p(\mathbf{z}_k)) \right\} \end{aligned} \right\} \end{aligned} \tag{6.20}$$

where $\theta_k^{\mathcal{M}} = \mathbf{\Phi}(\mathbf{z}_k)\mathbf{W}^{\mathcal{M}}$.

M-step for Gaussian noise model parameters

The posterior probabilities, $\mathbf{R}$ (computed at the E-step), are then used to re-estimate parameters of the weight matrix, $\mathbf{W}^{\mathcal{R}}$, using the following set of linear equations (the detailed derivations for this formula are available in (Bishop and Svensen, 1998))

$$\widehat{W^{\mathcal{R}}} = (\mathbf{\Phi}^T \mathbf{E} \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{R} \mathbf{X}^{\mathcal{R}}, \tag{6.21}$$

where $\mathbf{\Phi}$ is a $K \times L$ matrix with elements $\phi_l(z_k)$, $\mathbf{R}$ is a $K \times N$ matrix with elements r_{kn}, $\mathbf{X^R}$ is an $N \times |\mathcal{R}|$ data matrix of real values and the diagonal matrix $\mathbf{E}$ contains the values

$$e_{kk} = \sum_{n=1}^{N} r_{kn}. \tag{6.22}$$

Equation (6.21) can now be solved to determine the update weight matrix, $\widehat{\mathbf{W}^{\mathcal{R}}}$, and $\mathbf{\Phi}$ remains constant (and can be computed before the optimization starts). The re-estimation formula for the β can now be defined as

$$\frac{1}{\widehat{\beta^{\mathcal{R}}}} = \frac{1}{N|\mathcal{R}|} \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} ||\mathbf{\Phi}(\mathbf{z}_k)\widehat{\mathbf{W}}^{\mathcal{R}} - \mathbf{x}_n^{\mathcal{R}}||^2. \tag{6.23}$$

M-step for Bernoulli and multinomial noise model parameters

The parameters of a Bernoulli distribution weight parameter matrix, $\mathbf{W}^{\mathcal{B}}$, can be updated using a gradient-based approach (detailed derivation of this approach is given in (Kabán and Girolami, 2001)),

$$\Delta \mathbf{W}^{\mathcal{B}} \propto \mathbf{\Phi}^T \left[\mathbf{R}\mathbf{X}^{\mathcal{B}} - \mathbf{E} g^{\mathcal{B}}(\mathbf{\Phi} W^{\mathcal{B}}) \right], \tag{6.24}$$

where $X^{\mathcal{B}}$ is an $N \times |\mathcal{B}|$ data matrix containing binary values. A similar approach can also be used to update the parameter submatrix for each dth binary encoded multi-categorical feature using

$$\Delta \mathbf{W}^{\mathcal{C}}_{S_d} \propto \mathbf{\Phi}^T \left[\mathbf{R}\mathbf{X}^{\mathcal{C}}_{S_d} - \mathbf{E} g^{\mathcal{C}}(\mathbf{\Phi} W^{\mathcal{C}}_{S_d}) \right], \tag{6.25}$$

where $X^{\mathcal{C}}_{S_d}$ is an $N \times S^{\mathcal{C}}_d$ binary-encoded data submatrix of dth multi-category feature.

6.3.3 Experiments

The generalised GTM was evaluated using both synthetic and real datasets. The weight matrix, $\mathbf{W}$, was initialised using principal component analysis (PCA). The results of GGTM are compared with the standard GTM visualisation. For evaluating the quality of visualisations, we use the same quality measures as in other experiments in this thesis as: the data space is assumed to be of mixed-type, therefore we compute pair-wise distances using Hamming distances for the binary features and Euclidean distances for the continuous features and we divide each column in the distance matrix by its standard deviation in order to make both distance matrices on an equivalent scale.

Experiments were repeated with different latent and RBF grid sizes as explained in Table 6.3.3. We present here the visualisation results with a latent grid of size 8×8 and an RBF grid of size 4×4 both for training and test datasets. Visualisation quality evaluations metrics are shown with all latent grid and RBF grid sizes (in 1-to-6 settings) as explained in Table 6.3.3.

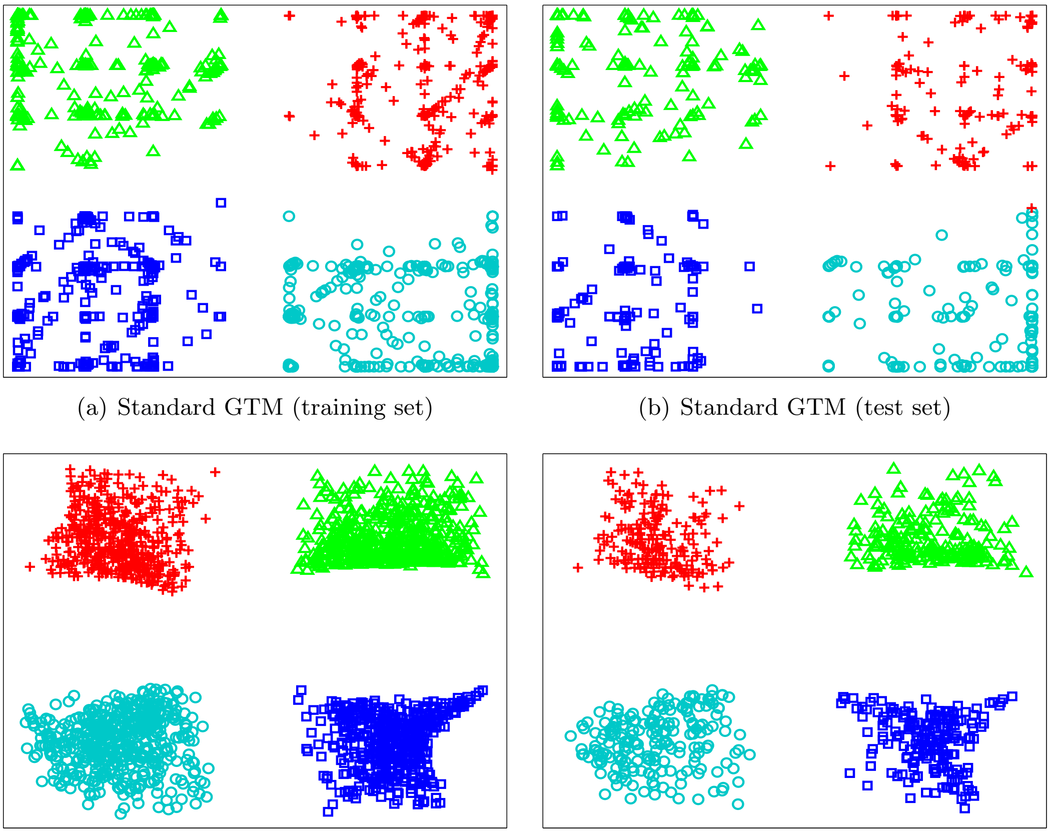
	Latent Grid Size	**RBF Grid Size**
1	12×12	8×8
2		4×4
3		2×2
4	8×8	16
5		2×2
6	4×4	2×2

Table 6.1: Latent and RBF grid sizes.

6.3.3.1 Synthetic dataset-I (continuous and binary features)

We generated a training dataset of $2,000$ data points from an equiprobable mixture of two Gaussians, $\mathcal{N}(\mathbf{m}_k, I)$ (with $k = 1, 2$) with means $\mathbf{m}_1 = \begin{pmatrix} 2.0 \\ 3.5 \\ 3.5 \end{pmatrix}$, and $\mathbf{m}_2 = \begin{pmatrix} 3.5 \\ 4.5 \\ 4.5 \end{pmatrix}$. We then generated a binary dataset of four classes with 9 binary features. The label variable we used indicates the four binary classes. We combined both datasets to make a dataset with $2,000$ data points and a total of 12 features. The test dataset of 800 data points with 12 features was also generated from the same distributions.

Visualisation results of standard GTM and the GGTM are shown in Figure 6.1. The visualisation quality evaluation metrics are given in Tables 6.2, 6.3 and 6.4.



(a) Standard GTM (training set)

(b) Standard GTM (test set)

(c) GGTM (training set)

(d) GGTM (test set)

Figure 6.1: Demonstration of the mixed-type data visualisations using the standard GTM and the GGTM models for synthetic dataset-I (i.e. consisting of continuous and binary features). GGTM visualisations in (c) and (d) has shown more compact blob like a separate cluster for each class compared to the standard GTM visualisations as shown in (a) and (b). Whereas in (b) few data points between red pluses and cyan circles classes overlap in case of the standard GTM.

		Trustworthiness		**Continuity**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	**0.9706**	0.9431	0.9568	0.9491
	GGTM	0.9623	**0.9575**	**0.9753**	**0.9664**
2	**GTM**	**0.9805**	0.9680	0.9691	0.9615
	GGTM	0.9687	**0.9708**	**0.9834**	**0.9777**
3	**GTM**	0.9524	0.9555	**0.9855**	**0.9791**
	GGTM	**0.9617**	**0.9677**	0.9836	0.9782
4	**GTM**	**0.9762**	0.9674	0.9705	0.9644
	GGTM	0.9715	**0.9757**	**0.9868**	**0.9809**
5	**GTM**	0.9523	0.9554	**0.9846**	**0.9787**
	GGTM	**0.9616**	**0.9673**	0.9831	0.9778
6	**GTM**	0.9462	0.9499	0.9691	0.9682
	GGTM	**0.9617**	**0.9672**	**0.9831**	**0.9775**

Table 6.2: The standard GTM and the GGTM comparison using quality evaluation metrics of trustworthiness and continuity (synthetic data I).

		MRREd		**MRREl**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	**0.0258**	0.1113	0.0303	**0.1097**
	GGTM	0.0301	**0.1073**	**0.0279**	0.1121
2	**GTM**	**0.0292**	0.0957	0.0305	0.1025
	GGTM	0.0298	**0.0955**	**0.0271**	**0.1010**
3	**GTM**	0.0324	**0.0978**	**0.0266**	**0.1028**
	GGTM	**0.0316**	0.0995	0.0269	0.1066
4	**GTM**	0.0317	0.1094	0.0319	0.1108
	GGTM	**0.0293**	**0.0886**	**0.0267**	**0.0980**
5	**GTM**	0.0335	0.1005	0.0271	**0.1043**
	GGTM	**0.0317**	**0.1004**	**0.0270**	0.1072
6	**GTM**	0.0348	0.1295	0.0306	0.1225
	GGTM	**0.0315**	**0.1001**	**0.0269**	**0.1067**

Table 6.3: The standard GTM and the GGTM comparison using quality evaluation metrics of the mean relative rank errors with respect to the data space and the latent space (synthetic data I).

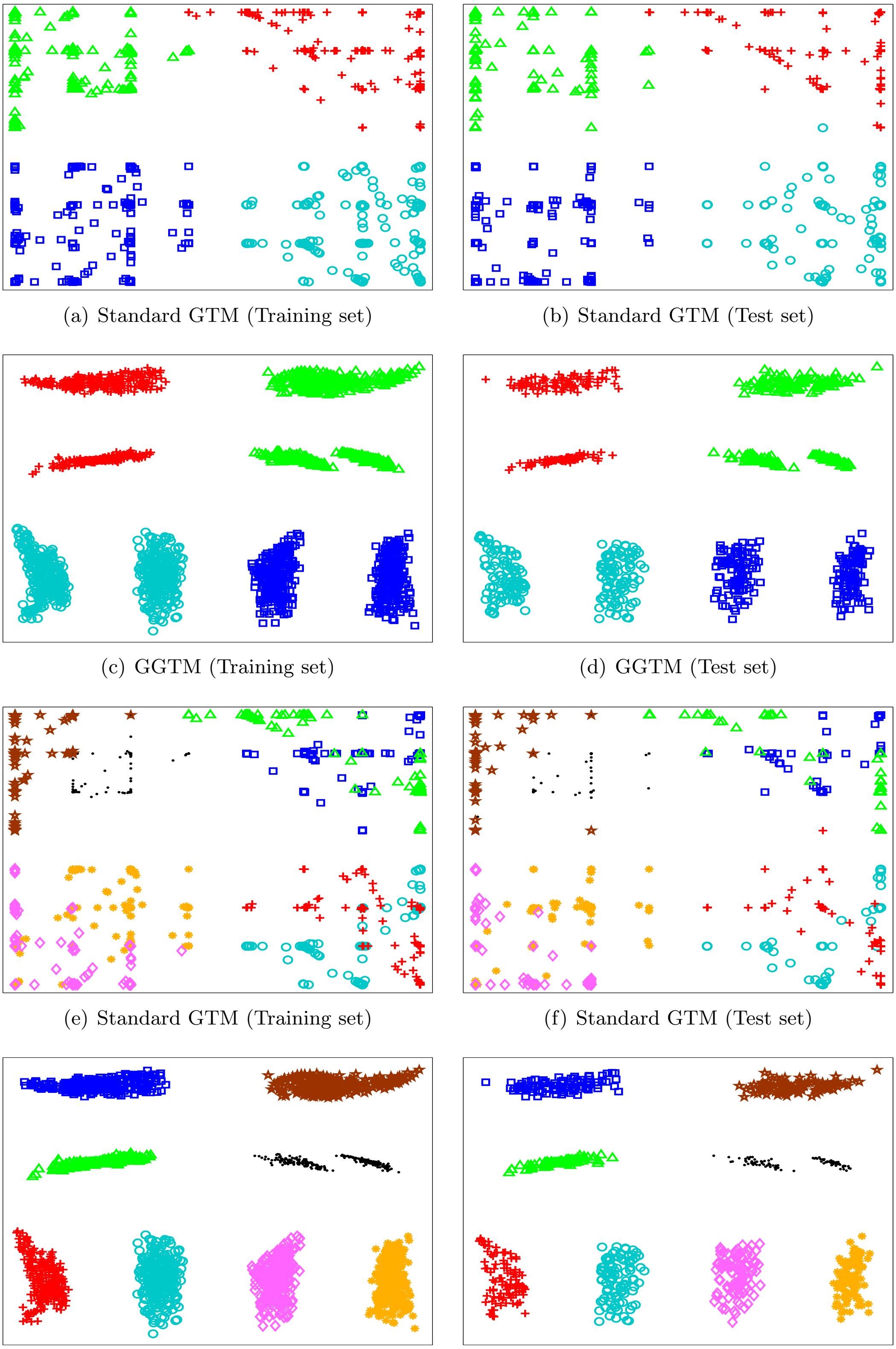
		AVDD		**NLL**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	0.7853	0.7034	**0.4281**	2.4839
	GGTM	**0.6705**	**0.6703**	0.5817	**0.6446**
2	**GTM**	0.7776	0.7077	2.9739	3.5763
	GGTM	**0.5905**	**0.6305**	**0.5970**	**0.6225**
3	**GTM**	0.5111	**0.5380**	**7.7252**	8.0322
	GGTM	**0.5674**	0.6228	0.7656	**0.7929**
4	**GTM**	0.8010	0.7012	3.9768	4.4205
	GGTM	**0.5521**	**0.5842**	**1.4003**	**1.4211**
5	**GTM**	**0.5417**	0.5526	7.7309	8.0449
	GGTM	0.5735	**0.6255**	**1.4976**	**1.5255**
6	**GTM**	0.8295	0.7415	7.9432	8.2174
	GGTM	**0.5781**	**0.6296**	**2.8040**	**2.8320**

Table 6.4: The standard GTM and the GGTM comparison using the quality evaluation metrics of the distance distortion and the negative log-likelihood per point (synthetic data I).

6.3.3.2 Synthetic dataset-II (continuous, binary and multi-category)

We used the same set of binary and real features dataset as generated in Section 6.3.3.1. Here we added two multi-category features both for training and test datasets with 8 and 16 categories in the first and second multi-category features respectively.

Visualisation results of standard GTM and GGTM are shown in Figure 6.2. The visualisation quality evaluation metrics are given in Tables 6.5, 6.6 and 6.7.



(a) Standard GTM (Training set)

(b) Standard GTM (Test set)

(c) GGTM (Training set)

(d) GGTM (Test set)

(e) Standard GTM (Training set)

(f) Standard GTM (Test set)

(g) GGTM (Training set)

(h) GGTM (Test set)

Figure 6.2: The standard GTM and the GGTM visualisations for the mixed-type synthetic dataset-II (i.e. continuous, binary and multinomial features). Subfigures (a), (b), (c) and (d) are assigned colours from 4 classes defined in the binary features whereas subfigures (e), (f), (g) and (h) are assigned colours from 8 classes (categories) based on the first multi-categorical feature.

		Trustworthiness		**Continuity**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	0.9652	0.9460	0.9254	0.9163
	GGTM	**0.9757**	**0.9762**	**0.9753**	**0.9778**
2	**GTM**	0.9713	0.9756	0.9638	0.9676
	GGTM	**0.9804**	**0.9873**	**0.9870**	**0.9903**
3	**GTM**	0.9204	0.9410	0.9468	0.9470
	GGTM	**0.9689**	**0.9837**	**0.9766**	**0.9867**
4	**GTM**	0.9626	0.9611	0.9389	0.9396
	GGTM	**0.9769**	**0.9867**	**0.9840**	**0.9892**
5	**GTM**	0.9194	0.9401	0.9440	0.9459
	GGTM	**0.9676**	**0.9834**	**0.9762**	**0.9867**
6	**GTM**	0.9162	0.9352	0.9320	0.9387
	GGTM	**0.9665**	**0.9831**	**0.9762**	**0.9867**

Table 6.5: The standard GTM and the GGTM comparison using quality evaluation metrics of the trustworthiness and the continuity (synthetic data II).

		MRREd		**MRREl**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	**0.0258**	**0.0655**	0.0309	**0.0840**
	GGTM	0.0306	0.0849	**0.0266**	0.0992
2	**GTM**	0.0332	0.0841	**0.0246**	0.1122
	GGTM	**0.0304**	**0.0757**	0.0248	**0.0932**
3	**GTM**	0.0417	0.1094	**0.0253**	**0.0958**
	GGTM	**0.0403**	**0.1054**	0.0297	0.1132
4	**GTM**	0.0363	0.0877	0.0268	0.1144
	GGTM	**0.0340**	**0.0863**	**0.0264**	**0.1001**
5	**GTM**	**0.0406**	0.1109	**0.0253**	**0.0935**
	GGTM	0.0408	**0.1065**	0.0298	0.1155
6	**GTM**	**0.0376**	0.1096	**0.0259**	**0.0930**
	GGTM	0.0419	**0.1073**	0.0294	0.1168

Table 6.6: The standard GTM and the GGTM comparison using quality evaluation metrics of the mean relative rank errors with respect to the data space and the latent space (synthetic data II).

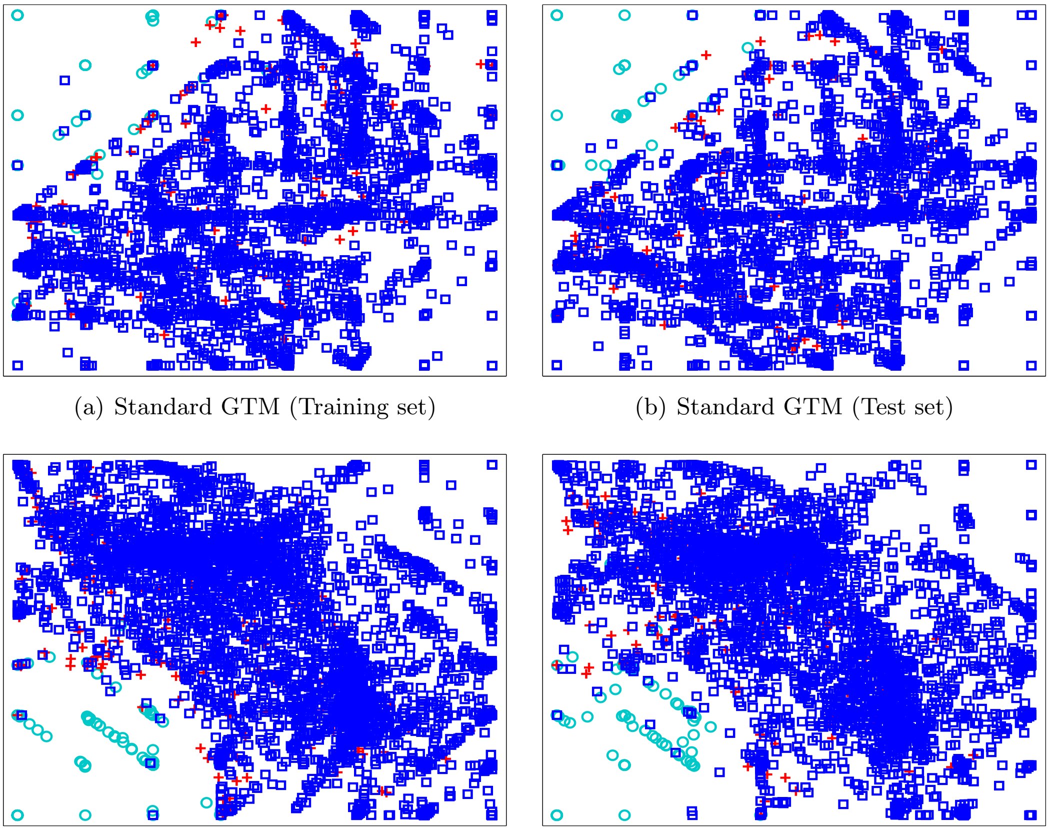
		AVDD		**NLL**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	0.7906	0.7744	**-10.6870**	**-6.6810**
	GGTM	**0.6720**	0.7444	-1.5930	-1.5877
2	**GTM**	0.7223	0.7492	**-1.8250**	-0.6415
	GGTM	**0.6036**	**0.6740**	-1.5643	**-1.5815**
3	**GTM**	**0.5442**	**0.6629**	8.1947	7.9889
	GGTM	0.7159	0.8178	**-1.0488**	**-1.0645**
4	**GTM**	0.7790	0.7969	0.3964	1.2724
	GGTM	**0.6513**	**0.7267**	**0.0284**	**0.0087**
5	**GTM**	**0.5950**	**0.6774**	8.3146	8.1828
	GGTM	0.7121	0.8127	**0.4604**	**0.4427**
6	**GTM**	0.7812	**0.7844**	9.3161	9.1896
	GGTM	**0.7151**	0.8052	**2.8838**	**2.8653**

Table 6.7: The standard GTM and the GGTM comparison using quality evaluation metrics of the distance distortion and the negative log-likelihood per point (synthetic data II).

6.3.3.3 Hypothyroid dataset

The first real dataset we used to demonstrate the effectiveness of the proposed GGTM is the hypothyroid disease dataset downloaded from the UCI data repository (Bache and Lichman, 2013). The dataset consists of mixed types, with 15 binary features and 6 continuous features, and contains three classes: primary thyroid, compensated thyroid, and normal. The dataset is divided into a training set of $3,772$ data points (93 with primary hypothyroid, 191 with compensated hypothyroid and 3488 normal) and a test set of $3,428$ data points (73 with primary hypothyroid, 177 with compensated hypothyroid and 3178 normal).

Visualisation results both for training and test datasets using standard GTM and the GGTM are shown in Figure 6.3. The visualisation quality evaluation metrics are given in Tables 6.8, 6.9 and 6.10.



(a) Standard GTM (Training set)

(b) Standard GTM (Test set)

(c) GGTM (Training set)

(d) GGTM (Test set)

Figure 6.3: The standard GTM and the GGTM visualisations of the thyroid disease mixed-type dataset. Cyan circles (‘o’) for primary hypothyroid, red plus sign (‘+’) for compensated hypothyroid and blue squares (‘□’) for normal.

		Trustworthiness		**Continuity**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	**0.7625**	**0.7331**	0.8307	0.8102
	GGTM	0.7143	0.6819	**0.8585**	**0.8405**
2	**GTM**	**0.7360**	**0.7172**	0.8122	0.7935
	GGTM	0.6869	0.6636	**0.8265**	**0.8073**
3	**GTM**	0.6789	0.6625	0.8175	0.8037
	GGTM	**0.7235**	**0.7105**	**0.8897**	**0.8774**
4	**GTM**	**0.7355**	**0.7195**	0.8315	0.8160
	GGTM	0.7039	0.6822	**0.8467**	**0.8296**
5	**GTM**	0.6724	0.6583	0.8145	0.8003
	GGTM	**0.7244**	**0.7115**	**0.8898**	**0.8778**
6	**GTM**	0.6530	0.6404	0.7904	0.7761
	GGTM	**0.7224**	**0.7115**	**0.8875**	**0.8763**

Table 6.8: The standard GTM and the GGTM comparison using quality evaluation metrics of the trustworthiness and the continuity (for the thyroid disease dataset).

		MRREd		**MRREl**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	**0.0166**	**0.0184**	0.0151	0.0167
	GGTM	0.0181	0.0201	**0.0142**	**0.0156**
2	**GTM**	**0.0177**	**0.0198**	0.0143	0.0159
	GGTM	0.0182	0.0200	**0.0140**	**0.0156**
3	**GTM**	**0.0180**	**0.0200**	0.0141	0.0155
	GGTM	0.0194	0.0216	**0.0140**	0.0155
4	**GTM**	**0.0182**	**0.0199**	0.0146	0.0162
	GGTM	0.0187	0.0205	**0.0141**	**0.0156**
5	**GTM**	**0.0174**	**0.0195**	0.0143	0.0158
	GGTM	0.0194	0.0216	**0.0140**	**0.0155**
6	**GTM**	**0.0167**	**0.0184**	0.0143	0.0157
	GGTM	0.0192	0.0214	**0.0140**	**0.0156**

Table 6.9: The standard GTM and the GGTM comparison using quality evaluation metrics of the mean relative rank errors with respect to the data space and the latent space (for the thyroid disease dataset).

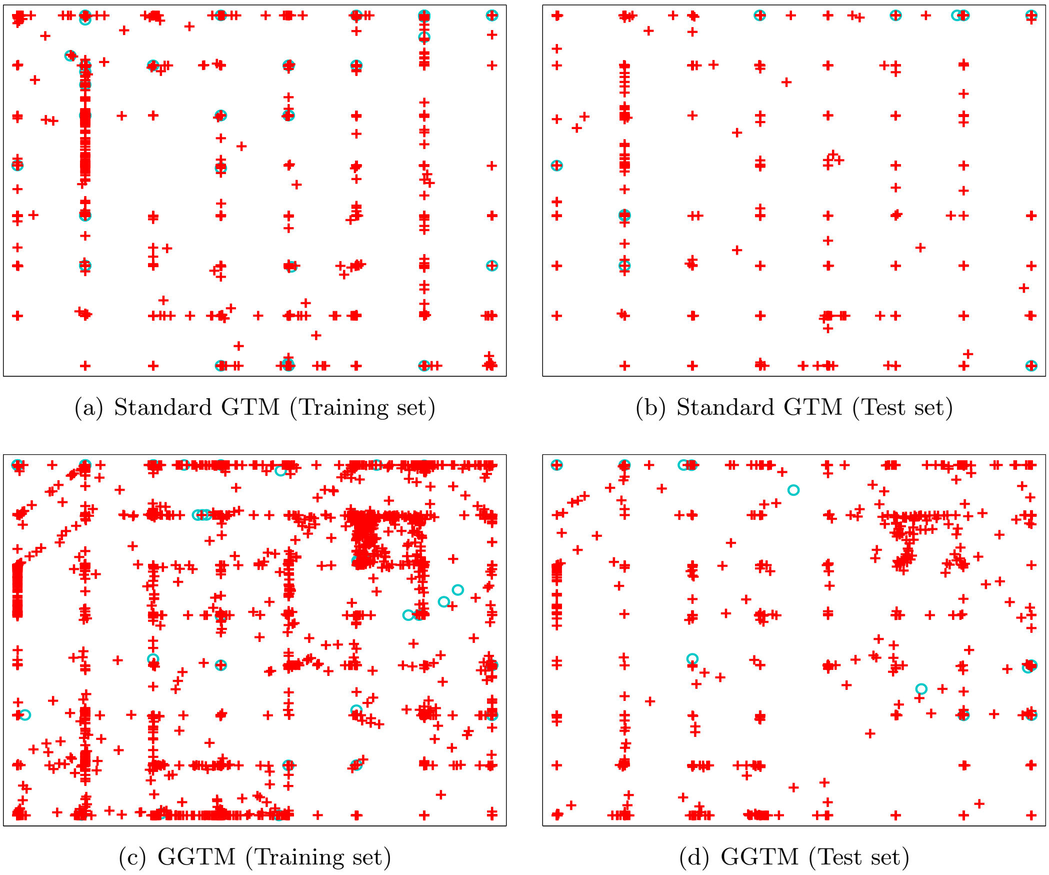
		AVDD		**NLL**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	0.7856	0.7839	3.8824	6.8544
	GGTM	**0.7298**	**0.7287**	**3.0192**	**3.5480**
2	**GTM**	0.8156	0.8178	5.8714	7.3545
	GGTM	**0.7566**	**0.7685**	**3.5234**	**3.8907**
3	**GTM**	0.8601	0.8640	9.5835	11.0325
	GGTM	**0.6677**	**0.6801**	**4.8464**	**5.1038**
4	**GTM**	0.8055	0.8089	6.7331	7.9179
	GGTM	**0.7548**	**0.7561**	**4.4990**	**4.8337**
5	**GTM**	0.8628	0.8639	10.2171	11.5110
	GGTM	**0.6696**	**0.6820**	**5.6612**	**5.9187**
6	**GTM**	0.9304	0.9306	11.6134	12.5628
	GGTM	**0.6850**	**0.6987**	**7.0969**	**7.3406**

Table 6.10: The standard GTM and the GGTM comparison using quality evaluation metrics of the distance distortion and the negative log-likelihood per point (for the thyroid disease dataset).

6.3.3.4 Bioassay dataset

The second real dataset we used to demonstrate the effectiveness of the GGTM is one of the bioassay datasets downloaded from the UCI data respository (Bache and Lichman, 2013). The dataset consists of mixed types, with 113 binary features and 31 continuous features, and contains two classes: active and inactive. The dataset is divided into a training set and test set with 3423 and 856 data points respectively.

Visualisation results for training and test datasets using standard GTM and the GGTM are shown in Figure 6.4. The visualisation quality evaluation metrics are given in Tables 6.11, 6.12 and 6.13. Results for two similar datasets (i.e. bioassays 'AID1608' and 'AID456') are given in Appendix D.4.



(a) Standard GTM (Training set)

(b) Standard GTM (Test set)

(c) GGTM (Training set)

(d) GGTM (Test set)

Figure 6.4: The standard GTM and the GGTM visualisations of bioassays dataset 'AID362'. Cyan circles ('o') for active compounds and red plus ('+') for inactive compounds.

		Trustworthiness		**Continuity**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	0.9044	0.8532	**0.8973**	**0.8574**
	GGTM	**0.9073**	**0.8572**	0.8875	0.8517
2	**GTM**	0.8812	0.8552	0.8806	0.8504
	GGTM	**0.8913**	**0.8642**	**0.8918**	**0.8534**
3	**GTM**	0.8430	0.8406	0.8959	0.8719
	GGTM	**0.8552**	**0.8535**	**0.9146**	**0.8929**
4	**GTM**	0.8779	0.8565	0.8864	**0.8583**
	GGTM	**0.8837**	**0.8576**	**0.8881**	0.8502
5	**GTM**	**0.8391**	0.8338	0.8956	0.8727
	GGTM	0.8390	**0.8390**	**0.9070**	**0.8873**
6	**GTM**	0.8005	0.8191	0.8773	0.8614
	GGTM	**0.8338**	**0.8371**	**0.9004**	**0.8815**

Table 6.11: The standard GTM and the GGTM comparison using quality evaluation metrics of the trustworthiness and the continuity for the bioassy dataset 'AID362'.

		MRREd		**MRREl**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	0.0204	0.0758	0.0178	0.0784
	GGTM	**0.0191**	**0.0731**	**0.0175**	**0.0766**
2	**GTM**	0.0198	0.0817	0.0170	0.0806
	GGTM	**0.0191**	**0.0798**	**0.0168**	**0.0802**
3	**GTM**	0.0194	0.0897	0.0167	0.0802
	GGTM	**0.0192**	**0.0882**	**0.0161**	**0.0778**
4	**GTM**	0.0204	0.0858	0.0171	0.0832
	GGTM	**0.0197**	**0.0814**	**0.0168**	**0.0799**
5	**GTM**	0.0195	0.0915	0.0165	0.0808
	GGTM	**0.0193**	**0.0908**	**0.0161**	**0.0781**
6	**GTM**	**0.0187**	0.0962	**0.0158**	0.0815
	GGTM	0.0192	**0.0911**	0.0162	**0.0811**

Table 6.12: The standard GTM and the GGTM comparison using quality evaluation metrics of the mean relative rank errors with respect to the data space and the latent space for the bioassy dataset 'AID362'.

		AVDD		NLL	
		Training	**Test**	**Training**	**Test**
1	**GTM**	0.8620	0.7840	**11.4491**	**16.0178**
	GGTM	**0.8409**	**0.7723**	30.3105	31.3519
2	**GTM**	0.8757	0.8302	**33.4452**	**34.6705**
	GGTM	**0.8526**	**0.8096**	37.5442	37.8452
3	**GTM**	0.8303	0.8236	55.9538	57.5050
	GGTM	**0.8007**	**0.7998**	**43.5324**	**43.6197**
4	**GTM**	0.8940	0.8292	**36.0574**	**37.2367**
	GGTM	**0.8610**	**0.7985**	38.7942	39.0690
5	**GTM**	0.8412	0.8215	57.0091	58.7741
	GGTM	**0.8126**	**0.7992**	**44.5537**	**44.6649**
6	**GTM**	**0.9110**	**0.8545**	62.1582	63.9599
	GGTM	0.9242	0.8767	**46.5073**	**46.5737**

Table 6.13: The standard GTM and the GGTM comparison using quality evaluation metrics of the distance distortion and the negative log-likelihood per point for the bioassy dataset ‘AID362’.

6.3.3.5 MHC class-I dataset

The MHC class-I dataset we consider here consists of 182 binary features derived from the primary sequences (from the $\alpha - 1$ and $\alpha - 2$ regions only) of MHC class-I by matching each sequence with the consensus sequence (as previously explained in Section 5.5.3). We combined this binary dataset with the electrostatic potential values dataset of MHC class-I with 2418 continuous features (as previously explained in Section 2.1.7) yielding a dataset with a total of 2600 features of mixed type for a total of 3944 proteins. We divided the dataset of 3944 proteins into training and test sets with 3157 and 787 data points respectively. As discussed earlier in chapter 4, the GTM type model had problems of tight clusters of points around the centres of the latent grid, implying that it is not suitable for visualising such high-dimensional datasets. However, our focus here is to compare the standard GTM and proposed GGTM model to see whether the GGTM model improves the visualisation results.

Visualisation results both for training and test datasets using standard GTM and the GGTM are shown in Figure 6.5. The visualisation quality evaluation metrics are given in Tables 6.14, 6.15 and 6.16.

		Trustworthiness		**Continuity**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	0.9307	0.8825	0.9222	0.8671
	GGTM	**0.9463**	**0.9200**	**0.9248**	**0.8941**
2	**GTM**	0.9311	0.9134	0.9071	0.8397
	GGTM	**0.9332**	**0.9047**	**0.9192**	**0.8665**
3	**GTM**	0.8862	0.8684	0.8740	0.8231
	GGTM	**0.8950**	**0.8806**	**0.9122**	**0.8790**
4	**GTM**	0.9131	0.8923	0.9032	0.8438
	GGTM	**0.9156**	**0.8938**	**0.9056**	**0.8625**
5	**GTM**	0.8606	0.8649	0.8717	0.8210
	GGTM	**0.8869**	**0.8833**	**0.8942**	**0.8703**
6	**GTM**	0.8087	0.8417	0.8534	0.8051
	GGTM	**0.8376**	**0.8558**	**0.8600**	**0.8382**

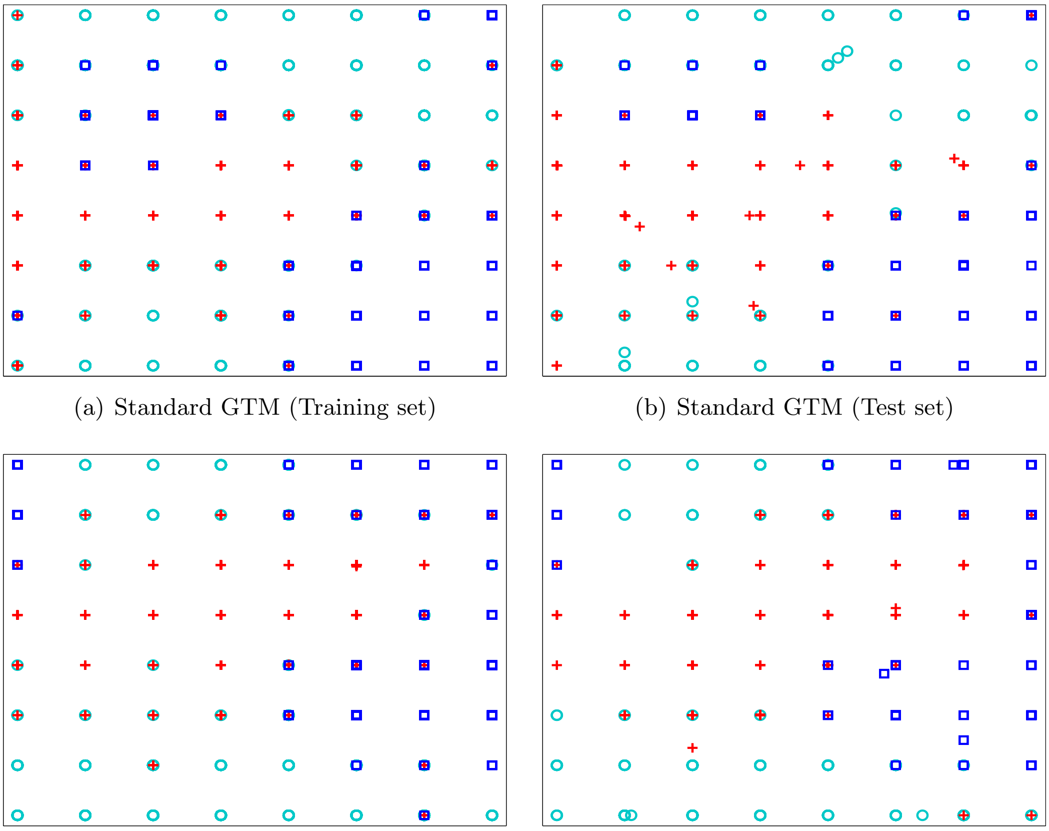
Table 6.14: The standard GTM and the GGTM comparison using quality evaluation metrics of the trustworthiness and the continuity for the MHC class-I dataset.

		MRREd		**MRREl**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	0.0214	0.0847	0.0209	0.0929
	GGTM	0.0214	**0.0811**	**0.0207**	**0.0911**
2	**GTM**	0.0219	0.0897	**0.0195**	0.0925
	GGTM	0.0219	**0.0884**	0.0198	**0.0931**
3	**GTM**	0.0209	**0.0912**	**0.0186**	0.0947
	GGTM	**0.0211**	0.0970	0.0186	**0.0904**
4	**GTM**	**0.0224**	0.0953	**0.0189**	0.0955
	GGTM	0.0223	**0.0937**	0.0193	**0.0942**
5	**GTM**	**0.0218**	**0.0955**	**0.0179**	**0.0925**
	GGTM	0.0210	0.0954	0.0183	0.0973
6	**GTM**	**0.0204**	0.1023	**0.0167**	**0.0877**
	GGTM	0.0200	**0.1034**	0.0174	0.0922

Table 6.15: The standard GTM and the GGTM comparison using quality evaluation metrics of the mean relative rank errors with respect to the data space and the latent space for the MHC class-I dataset).

		AVDD		**NLL**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	**0.8197**	**0.7477**	1432.1519	1524.2093
	GGTM	0.8407	0.7607	**47.2628**	**47.8952**
2	**GTM**	0.8728	0.8305	1649.0974	1669.0851
	GGTM	**0.8451**	**0.7633**	**52.9972**	**52.5008**
3	**GTM**	0.8397	0.8259	1887.3501	1908.1627
	GGTM	**0.7840**	**0.7854**	**65.7375**	**62.0124**
4	**GTM**	0.8511	0.7850	1748.2107	1766.4025
	GGTM	**0.8389**	**0.7746**	**55.0430**	**54.1448**
5	**GTM**	0.8225	0.7913	1958.4099	1972.6974
	GGTM	**0.8047**	**0.7764**	**64.0500**	**61.1906**
6	**GTM**	**0.8630**	**0.7769**	2142.9356	2155.5489
	GGTM	0.8730	0.7798	**64.9483**	**62.4839**

Table 6.16: The standard GTM and the GGTM comparison using quality evaluation metrics of the distance distortion and the negative log-likelihood per point for the MHC class-I dataset.



(a) Standard GTM (Training set)

(b) Standard GTM (Test set)

(c) GGTM (Training set)

(d) GGTM (Test set)

Figure 6.5: The standard GTM and the GGTM visualisations of the MHC class-I dataset (i.e. mixed of continuous electrostatic potential and binary sequence based features). Legend same as in Figure 4.4

6.3.4 Discussion

Visualisation of the mixed-type synthetic dataset-I (i.e. with only continuous and binary features) has shown good separation between classes using both standard GTM and GGTM visualisation algorithms both for training and test sets (with the exception of a few points from two classes that overlapped in case of standard GTM for the test dataset). As expected, the visualisation results suggested that GGTM performs better (for example see Figure 6.1) for all the considered latent and RBF grid sizes and the quality metrics were usually better for the GGTM than standard GTM (see Tables 6.2, 6.3 and 6.4). Experiments for synthetic dataset-II (i.e. with continuous, binary and multi-category features) have shown that the GGTM outperformed standard GTM in terms of visualisation considering the class separation (with labels assigned using binary classes and classes defined by multi-category features): see Figure 6.2. We also observed that GGTM outperformed GTM in terms of quality measures such as trustworthiness and continuity (see Table 6.5) with all considered latent and RBF grid sizes settings whereas the results are more mixed for other quality measures (see Tables 6.6 and 6.7).

For a real dataset of hypothyroid disease, visual inspection of results revealed that the GGTM visualisations were better than those of standard GTM in terms of separation between classes (see Figure 6.3). Like the synthetic datasets, we repeated the experiments with different latent and RBF grid sizes and observed that GGTM outperformed standard GTM both for training and test sets in terms of quality measures such as continuity, mean relative rank errors with respect to latent space, distance distortion (per point) and negative log-likelihood (per point) (see Tables 6.8, 6.9 and 6.10) whereas mean relative rank errors with respect to data always appeared slightly better for standard GTM and trustworthiness varied with different latent and RBF grid settings (i.e. were not consistently better for one model).

In the case of a bioassay dataset, GGTM visualisations revealed better structure compared to GTM visualisations (see Figure 6.4) and in terms of quality metrics with different latent and RBF grid settings quite often results for GGTM are again better compared to those for standard GTM (see Tables 6.11, 6.12 and 6.13).

We also observed that, for the MHC class-I dataset (with a mix of binary and continuous features), visualisations uing GGTM model appeared to be better compared to standard GTM (see Figure 6.5(c)) in separating three gene classes but the problem of tight-clusters around the latent grid centres was observed (the same is discussed in chapter 4). GGTM always performed better in terms of trustworthiness, continuity and negative log-likelihood whereas in terms of other measures (such as mean relative rank errors with respect to latent and data space and distance distortion) GGTM usually performed better than standard GTM with different latent and RBF grid sizes.

Overall, the GGTM generally gave better results than GTM on mixed data, considering both visual inspection and objective quality measures.

In practice it has been observed that many high-dimensional datasets contain some irrelevant ('noisy') features and removing those features or reducing their impact could play an important role in improving the visualisation results. In the next section we therefore propose an approach to determine the importance of feature as an integrated part of the model's parameter learning process.

6.4 A GGTM with Simultaneous Feature Saliency (GGTM-FS)

We extend GGTM visualisation model proposed in Section 6.3 to simultaneously estimate feature saliencies (we call this extension a GGTM-FS). For estimating feature saliency values under the GGTM visualisation model, we assume that each feature is independent of the component label under the appropriate noise model distribution. As a special case for the Gaussian noise model (appropriate for continuous features subset in the GGTM model settings), the feature independence assumption is modelled by adopting diagonal covariance matrices (as used in (Law et al., 2004; Maniyar and Nabney, 2006a)) instead of spherical covariance (as used in (Bishop and Svensen, 1998) and GGTM). Now the probability density function of the GGTM-FS model takes the form,

$$p(\mathbf{x}_n|\pi,\Theta) = \sum_{k=1}^{K} \pi_k \left[\prod_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}} \left[\prod_{d=1}^{|\mathcal{M}|} p(x_{nd}^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}}) \right] \right], \tag{6.26}$$

where $p(.|\Theta_{kd}^{\mathcal{M}})$ is the probability density functions of the dth feature for the kth component and π_k is the mixing coefficient of the kth component and is taken to be fixed to $\frac{1}{K}$ for all the components in the mixture model.

We take $\Psi = \{\Psi^{\mathcal{R}}, \Psi^{\mathcal{B}}, \Psi^{\mathcal{C}}\}$ and take the assumption that $\Psi^{\mathcal{M}} = (\psi_1^{\mathcal{M}}, \cdots, \psi_{|\mathcal{M}|}^{\mathcal{M}})$ (where $\mathcal{M} \in \{\mathcal{R},\mathcal{B},\mathcal{C}\}$), is the set of binary indicators $\psi_d^{\mathcal{M}} = 1$ for a relevant feature and $\psi_d^{\mathcal{M}} = 0$ otherwise. Now the probability density of our mixture model takes the form as

$$p(\mathbf{x}_n|\pi,\Theta,\lambda,\Psi) = \sum_{k=1}^{K} \pi_k \left[\prod_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}} \left[\prod_{d=1}^{|\mathcal{M}|} [p(x_{nd}^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})]^{\psi_d^{\mathcal{M}}} [q(x_{nd}^{\mathcal{M}}|\lambda_d^{\mathcal{M}})]^{(1-\psi_d^{\mathcal{M}})} \right] \right]. \tag{6.27}$$

The notion of feature saliency is modelled as: we first treat $\psi_d^{\mathcal{M}}$ as a missing variable in the EM algorithm and as a second step we compute the feature saliency, $\rho_d^{\mathcal{M}} = p(\psi_d^{\mathcal{M}} = 1)$, which is a probability of the dth feature relevance. The resulting model can now take the form,

$$p(\mathbf{x}_n|\Omega) = \sum_{k=1}^{K} \pi_k \left[\prod_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}} \left[\prod_{d=1}^{|\mathcal{M}|} [\rho_d^{\mathcal{M}} p(x_{nd}^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})] + [(1-\rho_d^{\mathcal{M}}) q(x_{nd}^{\mathcal{M}}|\lambda_d^{\mathcal{M}})] \right] \right], \qquad (6.28)$$

where $\Omega = \left\{\pi_k, \left\{\Theta_{kd}^{\mathcal{M}}\right\}, \left\{\lambda_d^{\mathcal{M}}\right\}, \left\{\rho_d^{\mathcal{M}}\right\}\right\}$ is the set of all the parameters of the model. A simple way to understand how equation (6.28) is obtained is to observe that

$$[p(x_{nd}^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})]^{\psi_d^{\mathcal{M}}} [q(x_{nd}^{\mathcal{M}}|\lambda_d^{\mathcal{M}})]^{1-\psi_d^{\mathcal{M}}} \qquad (6.29)$$

can be re-written as

$$\psi_d^{\mathcal{M}}[p(x_{nd}^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})] + (1-\psi_d^{\mathcal{M}})[q(x_{nd}^{\mathcal{M}}|\lambda_d^{\mathcal{M}})] \qquad (6.30)$$

as $\psi_d^{\mathcal{M}}$ is a binary indicator variable (for details see the proof in Appendix D.1). The log-likelihood can now take the form

$$\mathcal{L}(\Omega) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\Omega). \qquad (6.31)$$

6.4.1 An EM algorithm for GGTM-FS

The latent structure of GGTM model can be exploited to estimate feature saliencies, in a similar way as previously exploited for the standard GTM to estimate feature saliency (Maniyar and Nabney, 2006a). For this purpose, we consider flipping of a biased coin with probability $\rho_d^{\mathcal{M}}$; if the coin is a head then the feature is generated from the mixture component, $p(.|\Theta_{kd}^{\mathcal{M}})$, otherwise the component, $q(.|\lambda_d^{\mathcal{M}})$, is responsible.

We take $\mathbf{Y}$ (i.e. compnent labels) and Ψ as missing variables and we can derive an EM algorithm for estimating model parameters (see details in Appendix D.2). In the **E-step**, we use the current set of parameters, Ω, to compute the posterior probability (i.e. responsibility), $r_{nk} = p(y_n = k|\mathbf{x}_n)$, that the nth data point belongs to the kth mixture component using Bayes' theorem,

$$r_{nk} = \frac{\pi_k \left[\prod_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}} \left[\prod_{d=1}^{|\mathcal{M}|}[\rho_d^{\mathcal{M}} p(x_{nd}^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})] + [(1-\rho_d^{\mathcal{M}})q(x_{nd}^{\mathcal{M}}|\lambda_d^{\mathcal{M}})]\right]\right]}{\sum_{k=1}^{K} \pi_k \left[\prod_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}} \left[\prod_{d=1}^{|\mathcal{M}|}[\rho_d^{\mathcal{M}} p(x_{nd}^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})] + [(1-\rho_d^{\mathcal{M}})q(x_{nd}^{\mathcal{M}}|\lambda_d^{\mathcal{M}})]\right]\right]}. \tag{6.32}$$

The responsibility matrix, $\mathbf{R}$, is used to compute $u_{nkd}^{\mathcal{M}} = p(\psi_d^{\mathcal{M}} = 1, y_n == k|\mathbf{x}_n^{\mathcal{M}})$, which is a measure of the importance of the nth data point relating to the kth component using the dth feature and $v_{nkd}^{\mathcal{M}} = p(\psi_d^{\mathcal{M}} = 0, y_n = k|\mathbf{x}_n^{\mathcal{M}})$.

$$u_{nkd}^{\mathcal{M}} = \frac{\rho_d^{\mathcal{M}} p(x_{nd}^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})}{\rho_d^{\mathcal{M}} p(x_{nd}^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})] + [(1-\rho_d^{\mathcal{M}})q(x_{nd}|\lambda_d^{\mathcal{M}})} r_{nk}, \tag{6.33}$$

$$v_{nkd}^{\mathcal{M}} = r_{nk} - u_{nkd}^{\mathcal{M}}. \tag{6.34}$$

In the M-Step of the feature saliency parameter update, we take prior distributions for each type of variables separately as explained in Appendix D.3.

M-step for Gaussian noise model parameters:

We can use $\mathbf{U}^{\mathcal{R}}$ to re-estimate the weight matrix $\mathbf{W}^{\mathcal{R}}$ using a set of linear equations. Weight vector $\mathbf{w}_d$ of each dth feature can be updated using

$$\widehat{\mathbf{w}_d^{\mathcal{R}}} = (\mathbf{\Phi}^T \mathbf{E}_d^{\mathcal{R}} \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{U}_d^{\mathcal{R}} \mathbf{x}_d^{\mathcal{R}}, \tag{6.35}$$

where $\mathbf{\Phi}$ is a $K \times L$ matrix, $\mathbf{U}_d^{\mathcal{R}}$ is a $K \times N$ matrix calculated using equation (6.33), $\mathbf{x}_d^{\mathcal{R}}$ is a $N \times 1$ data vector of real values and a diagonal matrix $E_d^{\mathcal{R}}$ containing the values

$$e_{kkd}^{\mathcal{R}} = \sum_{n=1}^{N} u_{nkd}^{\mathcal{R}}. \tag{6.36}$$

Now we can straightforwardly re-estimate parameters of the mixture model using the re-estimated weight matrix, $\widehat{\mathbf{W}}^{\mathcal{R}}$: first we re-estimate the centres (for continuous features) of the mixture model in the data space (see equation (6.37)) and then we use these re-estimated centres to update the diagonal Gaussian width in each direction (for each continuous feature): see equation (6.38) (these are the same as for standard GTM-FS (Maniyar and Nabney, 2006a))

$$\widehat{Mean\Theta_k^{\mathcal{R}}} = \widehat{\mathbf{m}_k^{\mathcal{R}}} = \mathbf{\Phi}(\mathbf{z}_k)\widehat{W^{\mathcal{R}}}, \tag{6.37}$$

where $\widehat{\mathbf{m}_k^{\mathcal{R}}}$ is a $1 \times |\mathcal{R}|$ vector.

$$\frac{1}{\widehat{\beta_d^{\mathcal{R}}}} = \frac{\sum_k \sum_n u_{nkd}^{\mathcal{R}} (x_{nd}^{\mathcal{R}} - \widehat{m_{kd}^{\mathcal{R}}})^2}{\sum_k \sum_n u_{nkd}^{\mathcal{R}}}. \tag{6.38}$$

Common density parameters, $\lambda_d^{\mathcal{R}}$, can be updated using

$$\widehat{Mean\lambda_d^{\mathcal{R}}} = \frac{\sum_n (\sum_k v_{nkd}^{\mathcal{R}}) x_{nd}^{\mathcal{R}}}{\sum_{nk} v_{nkd}^{\mathcal{R}}} \tag{6.39}$$

$$\widehat{Var\lambda_d^{\mathcal{R}}} = \frac{\sum_n (\sum_k v_{nkd}^{\mathcal{R}} (x_{nd}^{\mathcal{R}} - \widehat{Mean\lambda_d^{\mathcal{R}}})^2}{\sum_{nk} v_{nkd}^{\mathcal{R}}}. \tag{6.40}$$

The feature saliency parameters for the continuous features set can be updated using

$$\widehat{\rho_d^{\mathcal{R}}} = \frac{\max(\sum_{nk} u_{nkd}^{\mathcal{R}} - \frac{KP}{2}, 0)}{\max(\sum_{nk} u_{nkd}^{\mathcal{R}} - \frac{KP}{2}, 0) + \max(\sum_{nk} v_{nkd}^{\mathcal{R}} - \frac{T}{2}, 0)}, \tag{6.41}$$

where P and T are the number of parameters in $\Theta_{kd}^{\mathcal{R}}$ and $\lambda_d^{\mathcal{R}}$ respectively.

M-step for Bernoulli noise model parameters:

In the M-step of the Bernoulli case, we use a simple gradient-based approach (Kabán and Girolami, 2001) to update the weights as,

$$\Delta \mathbf{w}_d^{\mathcal{B}} \propto \mathbf{\Phi}^T \left[\mathbf{U}_d^{\mathcal{B}} \mathbf{x}_d^{\mathcal{B}} - \mathbf{E}_d^{\mathcal{B}} g^{\mathcal{B}}(\mathbf{\Phi} \mathbf{w}_d^{\mathcal{B}}) \right], \tag{6.42}$$

$$e_{kkd}^{\mathcal{B}} = \sum_{n=1}^{N} u_{nkd}^{\mathcal{B}}. \tag{6.43}$$

Once we obtain the re-estimated vector $\widehat{\mathbf{w}_d^{\mathcal{B}}}$ of each dth feature from equation (6.42), we can then straightforwardly re-estimate centres of the Bernoulli distributions using equation (6.2) as follows,

$$\widehat{\mathbf{m}_k^{\mathcal{B}}} = g^{\mathcal{B}}(\mathbf{\Phi}(\mathbf{z}_k) \widehat{\mathbf{W}^{\mathcal{B}}}), \tag{6.44}$$

and the mean of the common density can be updated as,

$$\widehat{\lambda_d^{\mathcal{B}}} = \frac{\sum_n (\sum_k v_{nkd}^{\mathcal{B}}) x_{nd}^{\mathcal{B}}}{\sum_{nk} v_{nkd}^{\mathcal{B}}}. \tag{6.45}$$

The feature saliency parameter is updated as follows,

$$\widehat{\rho_d^{\mathcal{B}}} = \frac{\max(\sum_{nk} u_{nkd}^{\mathcal{B}} + \alpha_d - 1, 0)}{\max(\sum_{nk} u_{nkd}^{\mathcal{B}} + \alpha_d - 1, 0) + \max(\sum_{nk} v_{nkd}^{\mathcal{B}} + \beta_d - 1, 0)}. \tag{6.46}$$

M-step for multinomial noise model parameters:

The weights sub-matrix (where each sub-matrix represent weights for one encoded feature) for the multinomial case can be updated as,

$$\Delta \mathbf{W}^{\mathcal{C}}_{\mathbf{S}_d} \propto \mathbf{\Phi}^T \left[\mathbf{U}^{\mathcal{C}}_d \mathbf{X}^{\mathcal{C}}_{\mathbf{S}_d} - \mathbf{E}^{\mathcal{C}}_d g^{\mathcal{C}}(\mathbf{\Phi} \mathbf{W}^{\mathcal{C}}_{\mathbf{S}_d}) \right], \tag{6.47}$$

where $\mathbf{\Phi}$ is a $K \times L$ matrix, $\mathbf{W}_{S_d}$ is an $L \times S_d$ weight sub-matrix, $\mathbf{U}^{\mathcal{C}}_d$ is a $K \times N$ matrix obtained from equation (6.33), $\mathbf{X}_{\mathbf{S}_d}$ is a $N \times S_d$ data matrix (dth feature encoded to 1-of-S_d binary numbers), and $\mathbf{E}^{\mathcal{C}}_d$ is a $K \times K$ diagonal matrix with elements

$$e^{\mathcal{C}}_{kkd} = \sum_{n=1}^{N} u^{\mathcal{C}}_{nkd}. \tag{6.48}$$

Once we obtain the re-estimated matrix $\widehat{\mathbf{W}^{\mathcal{C}}_{S_d}}$ from equation (6.47), we can then straightforwardly calculate the mean of each feature of the multinomial distributions using equation (6.3) as

$$\widehat{\mathbf{m}^{\mathcal{C}}_{kS_d}} = g^{\mathcal{C}}(\mathbf{\Phi}(\mathbf{z}_k)\mathbf{w}^{\mathcal{C}}_{S_d}), \tag{6.49}$$

and the mean of the common density can be updated as follows,

$$\widehat{\lambda^{\mathcal{C}}_{S_d}} = \frac{\sum_n (\sum_k v^{\mathcal{C}}_{nkd}) x^{\mathcal{C}}_{nS_d}}{\sum_{nk} v^{\mathcal{C}}_{nkd}}. \tag{6.50}$$

The feature saliency parameter is updated with

$$\widehat{\rho^{\mathcal{C}}_d} = \frac{\max\left(\sum_{nk} u^{\mathcal{C}}_{nkd} - \frac{K(c_d-1)}{2}, 0\right)}{\max\left(\sum_{nk}, u^{\mathcal{C}}_{nkd} - \frac{K(c_d-1)}{2}, 0\right) + \max\left(\sum_{nk} v^{\mathcal{C}}_{nkd} - \frac{(c_d-1)}{2}, 0\right)}, \tag{6.51}$$

where c_d represents number of categories for the dth feature.

6.4.2 Experiments

A series of experiments were performed to demonstrate the effectiveness of the proposed GGTM-FS model both for synthetic and real datasets. Each weight sub-matrix (i.e. $\mathbf{W}^{\mathcal{R}}$, $\mathbf{W}^{\mathcal{B}}$ and $\mathbf{W}^{\mathcal{C}}$) was initialised using principal components analysis (PCA).

Algorithm 6.4.1: GGTM-FS algorithm summary

```
Input:Training dataset.
OutPut:Trained GGTM-FS visualisation model with feature saliency for
mixed-type features. begin
    Generate the latent grid points $\mathbf{z}_k \in \mathcal{H}$, $k = 1, \cdots, K$;
    Generate the basis function grid, $\mathbf{\Phi}(\mathbf{z}_k)$, centres $\{\nu_l\}$, $l = 1, \cdots, L$;
    Select the basis functions, $\mathbf{\Phi}(\mathbf{z}_k)$;
    Compute the design matrix of basis function activations, $\mathbf{\Phi}$ (as in LTM (Kabán
    and Girolami, 2001) and GTM (Bishop and Svensen, 1998));
    Initialise weight matrix ($\mathbf{W}$), randomly or using PCA;
    Apply the link functions (use equation (6.1) for continuous features,
    equation (6.2) for binary features and/or equation (6.3) for multi-categorical
    features) to initialise means of the mixture components;
    Initialise feature saliencies, $\rho_d^{\mathcal{R}}, \rho_d^{\mathcal{B}}, \rho_d^{\mathcal{C}}$, for each type $d$th feature, to 0.5;
    Initialise the mixing coefficient, $\pi_k$, with $\frac{1}{K}$ for each $k$th component in the grid;
    Set the initial means for the shared distributions, $q(.^{\mathcal{R}}|\lambda^{\mathcal{R}}), q(.^{\mathcal{B}}|\lambda^{\mathcal{B}}), q(.^{\mathcal{C}}|\lambda^{\mathcal{C}})$, as
    the mean of the data;
    repeat
        E-Step:
        Compute $\mathbf{R}$ (equation (6.32)) and $\mathbf{U}^{\mathcal{M}}$ (using equation (6.33)) and $\mathbf{V}^{\mathcal{M}}$
        (using equation (6.34)), using current parameters, $\Omega$;
        M-Step:
        for $d$=1 to|$\mathcal{R}$| do
            repeat
                Re-estimate the weight vector, $\mathbf{w}_d^{\mathcal{R}}$, (for continuous case), using
                $\widehat{\mathbf{w}_d^{\mathcal{R}}} = (\mathbf{\Phi}^T \mathbf{E}_d^{\mathcal{R}} \mathbf{\Phi})^{-1} \mathbf{\Phi}^T U_d^{\mathcal{R}} \mathbf{x}_d^{\mathcal{R}}$, from equation (6.35)
            until convergence;
        end
        for $d$=1 to|$\mathcal{B}$| do
            repeat
                Re-estimate the weight vector, $\mathbf{w}_d$, (for binary case), using
                $\Delta \mathbf{w}_d^{\mathcal{B}} \propto \mathbf{\Phi}^T \left[ \mathbf{U}_d^{\mathcal{B}} \mathbf{x}_d^{\mathcal{B}} - \mathbf{E}_d^{\mathcal{B}} g^{\mathcal{B}}(\mathbf{\Phi} \mathbf{w}_d^{\mathcal{B}}) \right]$, from equation (6.42)
            until convergence;
        end
        for $d$=1 to|$\mathcal{C}$| do
            repeat
                Re-estimate the weight matrix, $\mathbf{W}_{\mathbf{S}_d}$, (for multinomial case), using
                $\Delta \mathbf{W}_{\mathbf{S}_d}^{\mathcal{C}} \propto \mathbf{\Phi}^T \left[ \mathbf{U}_d^{\mathcal{C}} \mathbf{X}_{\mathbf{S}_d}^{\mathcal{C}} - \mathbf{E}_d^{\mathcal{C}} g^{\mathcal{C}}(\mathbf{\Phi} \mathbf{W}_{\mathbf{S}_d}^{\mathcal{C}}) \right]$, from equation (6.47);
            until convergence;
        end
        Re-estimate the means for each feature type, $\mathbf{m}_k^{\mathcal{R}}$ (using equation (6.37)),
        $\mathbf{m}_k^{\mathcal{B}}$ (using equation (6.44)), $\mathbf{m}_k^{\mathcal{C}}$ (using equation (6.49)), for each $k$th
        component of the mixture in the data space and also diagonal Gaussian
        width, $\frac{1}{\beta_d}$, for the all the continuous features using (6.38);
        Re-estimate the mean of the shared distribution using equation (6.39) for
        continuous features, equation (6.45) for binary data and/or equation (6.50)
        for multi-categorical data and also the shared distribution($q(.^{\mathcal{R}}|\lambda^{\mathcal{R}})$)
        variance for continuous feature set using equation (6.40);
        Re-estimate the saliencies of features, $\rho_d^{\mathcal{M}}$, using equation (6.41) for
        continuous data, using equation (6.46) for binary data or equation (6.51) for
        multi-categorical data;
    until convergence;
end
```

6.4.2.1 Synthetic dataset-I (continuous and binary features)

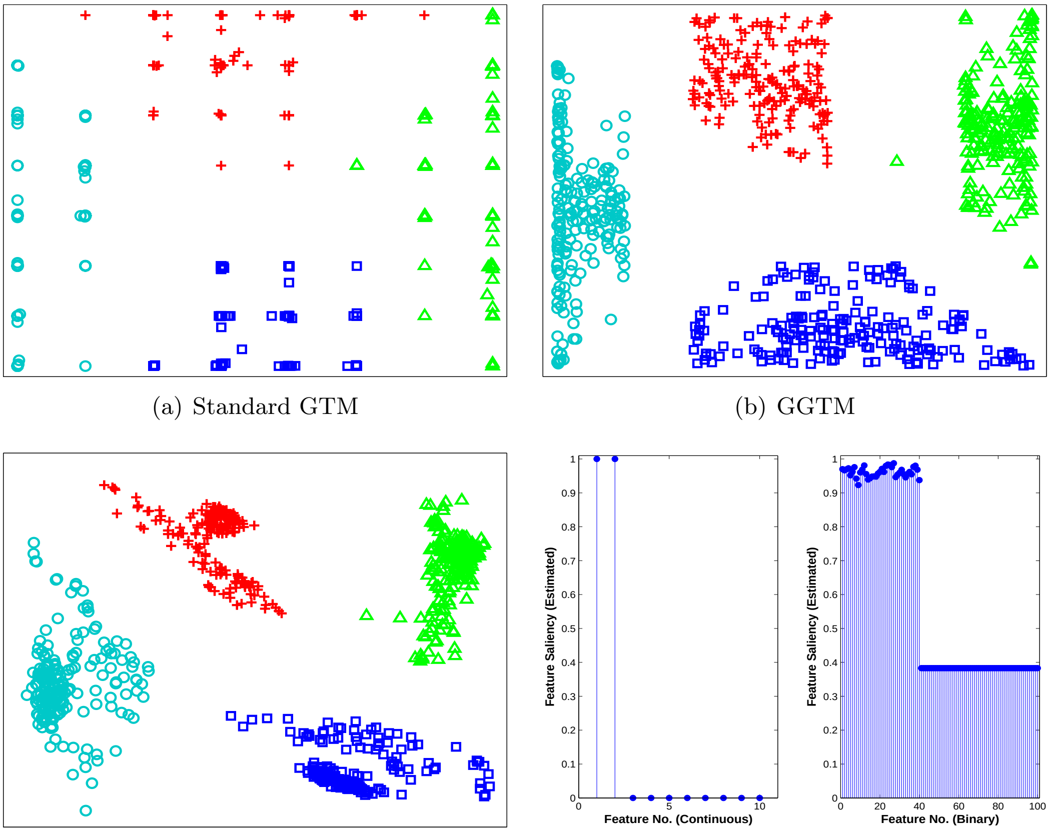
The synthetic dataset we generated contains 800 data points from an equiprobable mixture of four Gaussians, $\mathcal{N}(\mathbf{m}_k, I)$ with $k = 1, \cdots, 4$ with means $\mathbf{m}_1 = \binom{0}{3}$, $\mathbf{m}_2 = \binom{1}{9}$, $\mathbf{m}_3 = \binom{6}{4}$ and $\mathbf{m}_4 = \binom{7}{10}$ (similar as in Chapter 4). We then generated 8 noisy features (where each feature was sampled independently from $\mathcal{N}(0, I)$ distribution) and combined these with the original dataset yielding a 10-feature dataset. We then generated a binary dataset of 100 features where the first 40 features are drawn from four equiprobable clusters and the remaining 60 features are noisy. A small amount of noise (5%) was added by inserting random zeros in the informative features. For the uninformative features we added noise with different densities from no noise (by adding all zeros or all ones) or by $0.2, 0.4, 0.6, 0.8$. We combined both continuous and binary features yielding a dataset with 110 features with 800 data points. To model the data, we used a latent grid of size 8×8 and an RBF grid of size 4×4.

Visualisation results for standard GTM, GGTM and GGTM-FS are presented in Figure 6.6 where we also show feature saliencies estimated using the GGTM-FS.

6.4.2.2 Synthetic dataset-II (continuous, binary and multi-category)

We generated a two-feature multi-category dataset of 800 data points where first the feature has 8 equiprobable categories) and the second has 16 (giving a further two clusters for each category in the first feature) categories. Another set of two features were generated with 8 and 16 randomly distributed categorical values. We combined both two informative (non-noisy) and two uninformative (noisy) features yielding a multi-category dataset of 4 features. We then appended this 4 features multi-category dataset with the dataset of 110 continuous and binary features (as described in Section 6.4.2.1) yielding a mixed dataset of 114 features. We used a latent grid of size 8×8 and an RBF grid of size 4×4.

Visualisation plots from the standard GTM, proposed GGTM and GGTM-FS are shown in Figure 6.7 where we also show saliencies of features estimated using GGTM-FS.



(a) Standard GTM

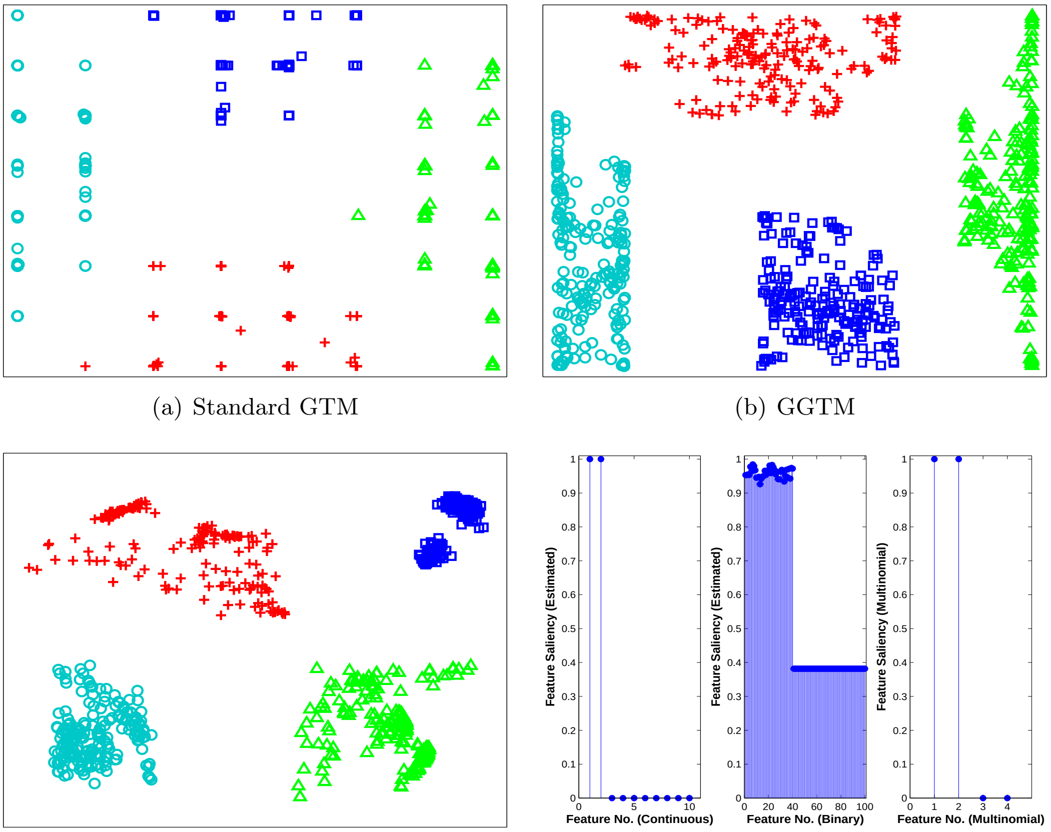
(b) GGTM

(c) GGTM-FS

(d) Estimated saliencies)

Figure 6.6: The standard GTM, the GGTM and the GGTM-FS visualisaion of the mixed-type synthetic dataset-I (i.e. continuous and binary features).

Demonstration of the mixed-type data visualisation using the standard GTM, the GGTM and the GGTM-FS for the synthetic dataset-I (i.e. continuous and binary features). The GGTM-FS visualisation in (c) show better visualisation with compact cluster for each class compared to the standard GTM and the GGTM visualisation given in (a) and (b) respectively, whereas (d) show estimated saliencies from the GGTM-FS.



(a) Standard GTM

(b) GGTM

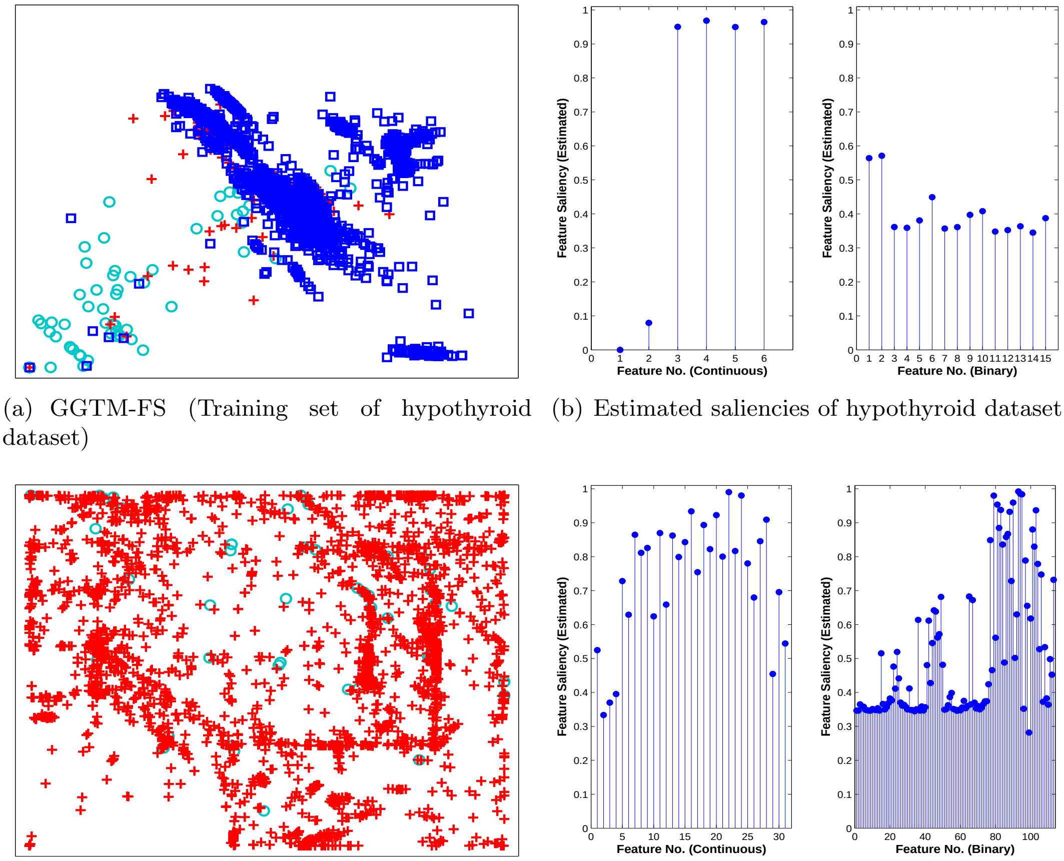
(c) GGTM-FS

(d) Estimated saliencies

Figure 6.7: Demonstration of mixed-type data visualisation using the standard GTM, the GGTM and the GGTM-FS for the synthetic dataset-II (i.e. continuous, binary and multinomial features). GGTM-FS visualisation in (c) show better visualisation with compact cluster for each class compared to standard GTM and GGTM visualisation given in (a) and (b) respectively, whereas (d) show estimated saliencies from GGTM-FS.

6.4.2.3 Real Datasets

The real datasets we used here are the same as in section 6.3.3 to demonstrate the effectiveness of proposed GGTM-FS model. We used a latent grid of size 8 × 8 and an RBF grid of size 4 × 4. Figure 6.8 shows the visualisations and feature saliency plots for the hypothyroid disease and bioassay datasets respectively. For other datasets such as bioassays 'AID1608' and 'AID456', results are given in Appendix D.4.



(a) GGTM-FS (Training set of hypothyroid dataset)

(b) Estimated saliencies of hypothyroid dataset

(c) GGTM-FS (Training set of AID362 dataset)

(d) Estimated Saliencies of AID362 dataset)

Figure 6.8: The GGTM-FS visualisations and the estimated feature saliencies. (a) and (b) relate to the hypothyroid dataset (legend same as in Figure 6.3); (c) and (d) relate to the bioassay dataset 'AID362' (legend same as in Figure 6.4).

6.4.3 Discussion

The visualisation results for synthetic dataset-I and -II using GGTM-FS have outperformed the other visualisation algorithms, with more compact clusters for each class in the dataset (see Figures 6.6 and 6.7). For synthetic dataset-I (with continuous and binary features), as expected, the model successfully determined that from a continuous feature set, 2 features are informative (with saliencies close to 1) whereas the other 8 features are uninformative (with saliencies close to 0) and from a binary feature set 40 features are informative (with saliencies close to 1) whereas the rest of the 60 are less informative (with smaller saliencies) (see Figure 6.6(d)). For the synthetic dataset-II, where we have four multi-category features in addition to the 110 features of synthetic dataset-I, the saliencies for the continuous and binary features were observed to be similar to the results of the synthetic dataset-I. In addition, as expected, the model successfully determined that the first two features from the multi-category feature set are informative (with saliencies close to 1) whereas the other two features are uninformative (with saliencies equal to 0) (see Figure 6.7(d)).

For the real datasets. the visualisation results for GGTM-FS (see Figure 6.8(a)) are better than both standard GTM and GGTM (see Figure 6.3). In addition GGTM-FS revealed that from the continuous feature set, the first two features have very low saliencies whereas the other features have higher saliency values and from the binary feature set the first two features have slightly better saliencies compared to other features (see Figure 6.8(b)).

The visualisation results for GGTM-FS on the bioassay dataset ‘AID362’ also revealed some interesting structures (see Figure 6.8(c)) compared to the results of standard GTM and GGTM (see Figure 6.4). The saliency values for both continuous and binary features suggested not all the features are informative and this can be seen in Figure 6.8(d).

6.5 Conclusion

In the literature not much attention has been given to analysing mixed-type data using the latent variable formalism. In practice quite often all the features in a mixed-type dataset are transformed to a single type (e.g. if there is a mixture of continuous and discrete variables, then either all the discrete variables are converted to some numerical scoring equivalent or all the continuous variables are considered as discrete variables with some grouping criteria) before applying the appropriate latent variable model. Adopting this transformation approach leads to a loss of information, which affects the results. However, considering the types of variables in the modelling process without any transformation should give better results.

Influenced by the latent trait model (LTM) (Kabán and Girolami, 2001) which was mainly developed for visualising datasets of discrete variables, we were successful in deriving a non-linear model for visualising a mixed-type dataset. We called this model a generalised GTM (GGTM). Experimental visualisation results for both synthetic and real mixed-type datasets have shown that this model performed much better than the standard GTM. We were also successful in extending GGTM to estimate feature saliencies. This extension has also shown success both for synthetic and real datasets not only in terms of improved visualisations but also explaining the importance of features which is also very important in its own right.

7 Conclusions and Future Directions

CONTENTS

In this last chapter of the thesis, we first summarise the outcomes from each chapter and then give some possible future extensions to the work presented here.

7.1 Chapter summary

The research presented in this thesis mainly focuses on developing data exploration algorithms for effective modelling of large and high-dimensional heterogeneous biological datasets. The main motivation behind this project was to determine similarities among proteomic and genomic variations that are important to develop effective medicines and in understanding biological functions such as transplantation rejection, smell recognition etc. We adapted latent-variable model-based techniques to explore patterns in biological datasets but the techniques developed are equally applicable to datasets from other domains. The application of the models developed in this thesis have revealed some useful results to our collaborator from the school of Life and Health Sciences. We now summarise each chapter in this thesis.

Chapter 2

In this chapter we reviewed the basics of bioinformatics mainly focusing on proteins and their structural variants. We also reviewed the major histocompatibiltity complex (MHC) protein family, reasons for studying MHCs and also discussed some of the previous analysis of this protein family. We then reviewed methods of computing electrostatic potential energy for the protein's three-dimensional structure: Electrostatic potentials are important to understand interactions with other proteins or antigens. At the end of the chapter, we explained that experimentally it is costly and time consuming to determine three-dimensional protein structures and therefore biologists are using *in-silico* methods for predicting a protein's three-dimensional structures from known primary sequences. This process generates a high-dimensional dataset which must be analysed to understand and predict better protein structures and properties. In this thesis we opted to analyse such a dataset with data projection methods.

Chapter 3

In this chapter we first reviewed some general purpose visual data mining systems. We also explain briefly the software engineering work that we caried out on the Data Visualisation and Modelling system (DVMS) in order to make it easier to integrate new data projection algorithms. We then reviewed the projection algorithms that are supported by this system: principal component analysis (PCA), Neuroscale

(NSC), generative topographic mapping (GTM) and its variants such as GTM with simultaneous feature saliency, hierarchical GTM, latent trait model (generalisation of GTM that was developed for discrete data) and the Gaussian process latent variable model (GPLVM). Demonstrations of these algorithms were given in the remaining chapters. Data projection methods are considered as unsupervised learning methods and therefore measuring and comparing their performance is difficult. We reviewed some of the methods that we used in rest of the chapters to evaluate the visualisation quality.

Chapter 4

Standard GTM and its extensions such as GTM with feature saliency and hierarchical GTM were observed not to be computationally tractable while training a model for a high-dimensional dataset (usually with dimensions greater than a few hundred). In this chapter we proposed variants of these algorithms where we adopt log-transformations at various steps of the parameter learning process in order to make them tractable for high-dimensional datasets. We tested these variants successfully both for a synthetic (with 500 dimensions) and a real dataset of MHC class-I electrostatic potential values. In order to compare the results for the MHC dataset, other projection algorithms such as PCA, Neuroscale and GPLVM were used. Our proposed variants gave better results, in terms of visual inspection and quality metrics, compared to PCA and Neuroscale but not better than GPLVM. GPLVM in general outperformed all these algorithms and our discussions with the biologists also confirmed that the GPLVM results were more useful and informative.

Chapter 5

The LTM was proposed as a generalisation of the GTM visualisation model in order to use different noise models based on the type of features. In this chapter we derived a latent trait (LTM) based data visualisation models to simultaneously estimate feature saliencies while learning the parameters of the model. This approach has not only improved visualisations (with more compact clusters) by modelling the irrelevant (noisy) features with a shared distribution but also gave feature saliencies which are valuable to understand the importance of each feature. Experimental results both for synthetic and real datasets were encouraging.

Chapter 6

In this chapter we derived a generalised GTM (GGTM) model for visualising a mixed-type dataset under the latent variable framework. The proposed model considers appropriate noise models (e.g. Gaussian for continuous features, Bernoulli for binary features and multinomial for multi-category features) for each type of feature(s) in a mixed-typed dataset to give a single visualisation plot. Experimental results both for synthetic and real datasets were encouraging both in terms of visualisation and visualisation quality evaluation metrics. We also extended this model (i.e. GGTM) to simultaneously estimate feature saliencies while learning the parameters of the model and call this as GGTM-FS. GGTM-FS results were also encouraging both for synthetic and real datasets.

7.2 Future Directions and open questions

In the near future we plan to extend the work presented in this thesis in the following contexts:

- In this thesis we considered *in-silico* methods of three-dimensional protein-structure modelling using homology modelling and the continuum Poisson-Boltzmann electrostatic potential for the region covering the top surface of the structure. We can extend these approaches and apply them to the classification of MHC alleles in terms of peptide specificity, TCR specificity, and antibody interaction and also use it to investigate practical problems in terms of epitope prediction, solid organ and bone-marrow transplantation, mate choice and MHC-mediated adverse drug reactions. We also plan to extend other grid-based properties such as hydrophobicity, polarity, mutability etc. We expect the analysis techniques presented in this thesis could also be useful for other structural systems such as G-Protein Coupled Receptors (Bjarnadóttir et al., 2006) and Kinases (Endicott and Noble, 2013; Vanderstraete et al., 2013).

- In chapter 4 we discussed the computational intractability of GTM-like models for visualising high-dimensional datasets and proposed variants where they can be made tractable using log-transformations. However, even when the parameter learning process was more effective, we observed tight clusters around the nodes of the regular latent grid. GTM-like algorithms often have this problem in cases of high-dimensional datasets. However, this tight-clustering effect was not observed while training a log-based GTM variant model which estimate feature saliencies as an integrated part of the parameters' learning process because this can either remove or reduce the impact of noisy features. The same tight-clustering effect was also observed in the hierarchical GTM and we therefore propose that extension to the hierarchical GTM to simultaneously estimate compute feature saliencies will aslso be useful. The challenging part of this extension will be deriving a strategy that shows how the feature significance obtained at higher level will carry forward in the hierarchy.

- Influenced by the hierarchical GTM model, our future intension is to extend Gaussian process latent variable model to generate hierarchical visualisation models. But the limitation we observed is that there is no simple way of modifying the GPLVM to take account of 'soft' cluster membership, as would be needed for a probabilistic hierarchy (as adopted in hierarchical GTM). It will be interesting if we extend this approach using a probabilistic mixture based ('soft') cluster memberships (which is considered to be a more principled methodology) in order to generate a hierarchy of visualisations.

- One of the avenues of future work is to extend the generalised GTM (GGTM) to a probabilistic mixture-based hierarchical visualisation model (like hierarchical GTM). Another possible extension will be to further extend such a hierarchial visualisation of mixed-type data to estimate feature saliencies while training a hierarchy of visualisation under the probabilistic mixture framework.

- The software tool we developed during this period of time is already freely accessible from our research group webside [1]. We have already added both GGTM and GGTM-FS models to this tool. In future it will be a continuing process to update this tool by adding more projection algorithms and also adding more interactive features and making it available mainly for non-statisticial users to get better understanding of their datasets.

[1] http://www.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/

A Proteins' 3-D structure modelling and electrostatic potential

The three-dimensional structures of proteins are important for understanding protein functions like electrostatic interactions of protein-ligand and protein-protein bindings (Price and Nairn, 2009). We generated our dataset in a two-step process: the first is the prediction of three-dimensional structure of proteins and the second step is the calculation of the continum electrostatic potential values in a multigrid environment.

- Homology Modelling Process;
- Electrostatic Potential Map Calculation Process;

A.1 Homology Modelling Process

Homology modelling is a multi-step process for predicting the three-dimensional protein structures. Figure A.1 explains the steps involved in the process of homology modelling. We used two software tools, one called Modeller (Sali, 2010) and the other called SCWRL4 (Krivov et al., 2009). Modeller is a software tool that is involved to perform the first six steps as shown in Figure A.1 to predict the three-dimensional protein struc-

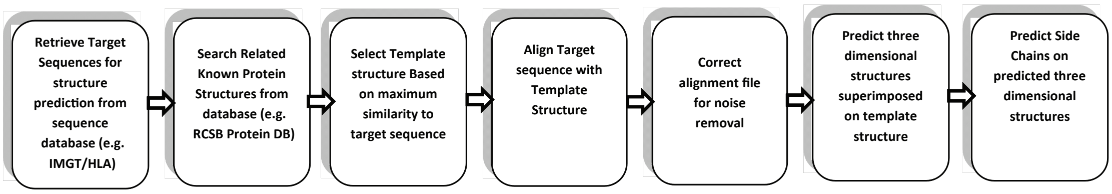
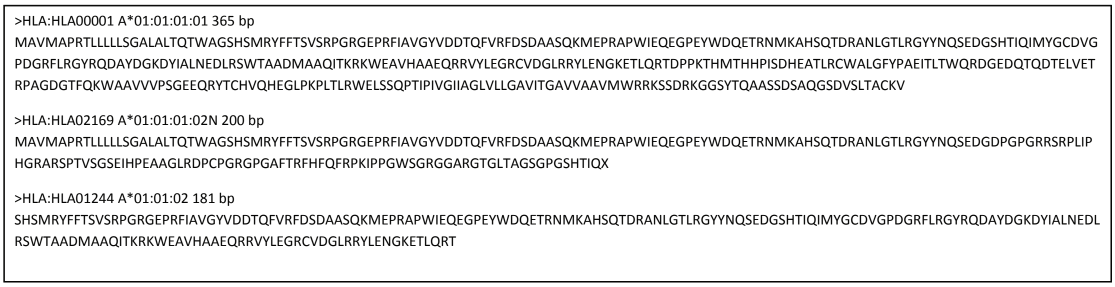

Retrieve Target Sequences for structure prediction from sequence database (e.g. IMGT/HLA)
Search Related Known Protein Structures from database (e.g. RCSB Protein DB)
Select Template structure Based on maximum similarity to target sequence
Align Target sequence with Template Structure
Correct alignment file for noise removal
Predict three dimensional structures superimposed on template structure
Predict Side Chains on predicted three dimensional structures

Figure A.1: Homology modelling process.

tures. Modeller provides a facility of writing ordinary Python [1] computer programming language scripts to perform different tasks using different functions at different stages of the three-dimensional structure prediction. SCWRL4 software tool is used only at step seven as shown in Figure A.1 to predict side chains for the predicted three-dimensional protein structures. The details of the steps for using these software tools are given in the following sub-sections.

A.1.1 The Target Sequence Retrieval from The Sequence Database

The first step in the homology modelling process is to download an amino acid sequence file/files from the target sequence database. In our experiments, we are interested in HLA class-I proteins. Sequence database files of target HLA alleles for target genes were downloaded from the IMGT/HLA database (Robinson et al., 2003). Here in Figure A.2, part of the sequence database file for an HLA-A gene sequence is shown in FASTA format.



```
>HLA:HLA00001 A*01:01:01:01 365 bp
MAVMAPRTLLLLLSGALALTQTWAGSHSMRYFFTSVSRPGRGEPRFIAVGYVDDTQFVRFDSDAASQKMEPRAPWIEQEGPEYWDQETRNMKAHSQTDRANLGTLRGYYNQSEDGSHTIQIMYGCDVG
PDGRFLRGYRQDAYDGKDYIALNEDLRSWTAADMAAQITKRKWEAVHAAEQRRVYLEGRCVDGLRRYLENGKETLQRTDPPKTHMTHHPISDHEATLRCWALGFYPAEITLTWQRDGEDQTQDTELVET
RPAGDGTFQKWAAVVVPSGEEQRYTCHVQHEGLPKPLTLRWELSSQPTIPIVGIIAGLVLLGAVITGAVVAAVMWRRKSSDRKGGSYTQAASSDSAQGSDVSLTACKV

>HLA:HLA02169 A*01:01:01:02N 200 bp
MAVMAPRTLLLLLSGALALTQTWAGSHSMRYFFTSVSRPGRGEPRFIAVGYVDDTQFVRFDSDAASQKMEPRAPWIEQEGPEYWDQETRNMKAHSQTDRANLGTLRGYYNQSEDGDPGPGRRSRPLIP
HGRARSPTVSGSEIHPEAAGLRDPCPGRGPGAFTRFHFQFRPKIPPGWSGRGGARGTGLTAGSGPGSHTIQX

>HLA:HLA01244 A*01:01:02 181 bp
SHSMRYFFTSVSRPGRGEPRFIAVGYVDDTQFVRFDSDAASQKMEPRAPWIEQEGPEYWDQETRNMKAHSQTDRANLGTLRGYYNQSEDGSHTIQIMYGCDVGPDGRFLRGYRQDAYDGKDYIALNEDL
RSWTAADMAAQITKRKWEAVHAAEQRRVYLEGRCVDGLRRYLENGKETLQRT
```

Figure A.2: A sample sequence database file in the FASTA format showing three sequences of HLA-A gene. The sequence headers are represented in lines starting with '>' and the amino acid sequences are represented using short letters in the rest of the lines before the next sequence header.

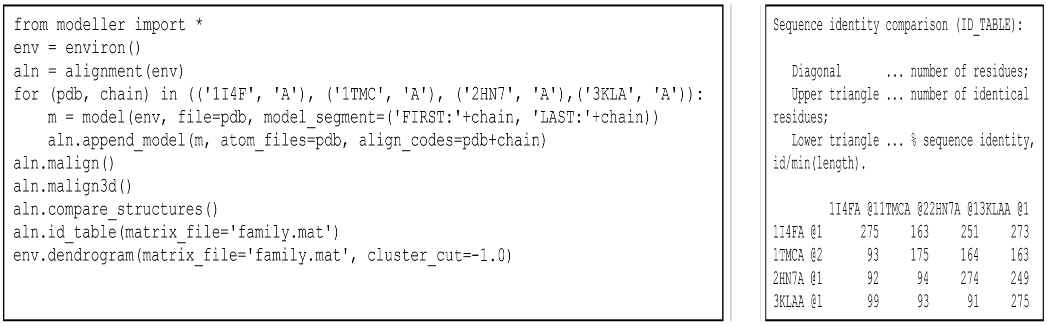
[1] http://www.python.org/

A.1.2 The Related Known Structure Search from the Structure Database

The second step in the homology modelling process is to search and download known structure files that are most related to the target sequence gene type from a protein data bank (Bernsten et al., 1977). For this purpose we were interested to search the structure files whose primary sequences have at least 30% similarity with each of the alleles of HLA-A, HLA-B and HLA-C and for example for the HLA-A we found four protein structures (i.e. protein data bank codes '1I4F','ITMC','2HN7' and '3KLA') with minimum required similarity. The Modeller program also provides the functionality of searching a similar structure to a target sequence from the structure file provided with the program. However, this contains a limited number of protein structures, so we searched a structure file from the up-to-date protein data bank available online.

A.1.3 The Template Structure Selection

The third step in the homology modelling process is to select the structure that has the maximum length with the maximum identity on the specific positions of amino acid residues for sequences of all the related known structures. In our experiments, we downloaded pre-aligned files, for sequences of HLA-A, HLA-B and HLA-C from IMGT/HLA database. These pre-aligned files represent that all the sequences are at least 50% similar in each type of HLA. Therefore, a single known structure for each for HLA-A gene will be good enough for predicting structures of the sequences of HLA-A and the same was observed to be true for HLA-B and HLA-C. For example in case of HLA-A, we downloaded four known structures as explained in Section A.1.2 and compared these structures of HLA-A gene with downloaded sequences of HLA-A. The Modeller software provide function called 'Compare_struture' to compare a set of structures. A Modeller script using this function and part of the output file (.log extension file) showing a sequence identity table are shown in Figure A.3. A sequence identity table assists user in selecting the best structure from a set of structures for the case of HLA-A. The sequence identity table shows that out of these four structures two structures with protein data bank code '1I4F' and '3KLA' have the same length of residues with 273 numbers of residues on the same positions. However, anyone can be selected to predict the structures. From these two, we selected the structure with protein data bank code '1I4F'. This selected structure will be termed as a template structure in this thesis for the HLA-A. For the other two HLAs (i.e. HLA-B and HLA-C) we adopted similar criteria and observed that '1AGD' for HLA-B and '1IM9' for HLA-C retrieved from the protein data bank were the most similar ones with the target sequences of the corresponding HLA type. The three reference protein structures we selected are the same already used by (Doytchinova et al., 2004) for predicting the structures of HLAs.



```
from modeller import *
env = environ()
aln = alignment(env)
for (pdb, chain) in (('1I4F', 'A'), ('1TMC', 'A'), ('2HN7', 'A'),('3KLA', 'A')):
    m = model(env, file=pdb, model_segment=('FIRST:'+chain, 'LAST:'+chain))
    aln.append_model(m, atom_files=pdb, align_codes=pdb+chain)
aln.malign()
aln.malign3d()
aln.compare_structures()
aln.id_table(matrix_file='family.mat')
env.dendrogram(matrix_file='family.mat', cluster_cut=-1.0)
```

```
Sequence identity comparison (ID_TABLE):

   Diagonal       ... number of residues;
   Upper triangle ... number of identical
residues;
   Lower triangle ... % sequence identity,
id/min(length).

         1I4FA @11TMCA @22HN7A @13KLAA @1
1I4FA @1      275     163     251     273
1TMCA @2       93     175     164     163
2HN7A @1       92      94     274     249
3KLAA @1       99      93      91     275
```

(a) (b)

Figure A.3: Demonstration of template structure selection. (a) A modeller script for comparing sequence of known structures. (b) A sequence identity table.

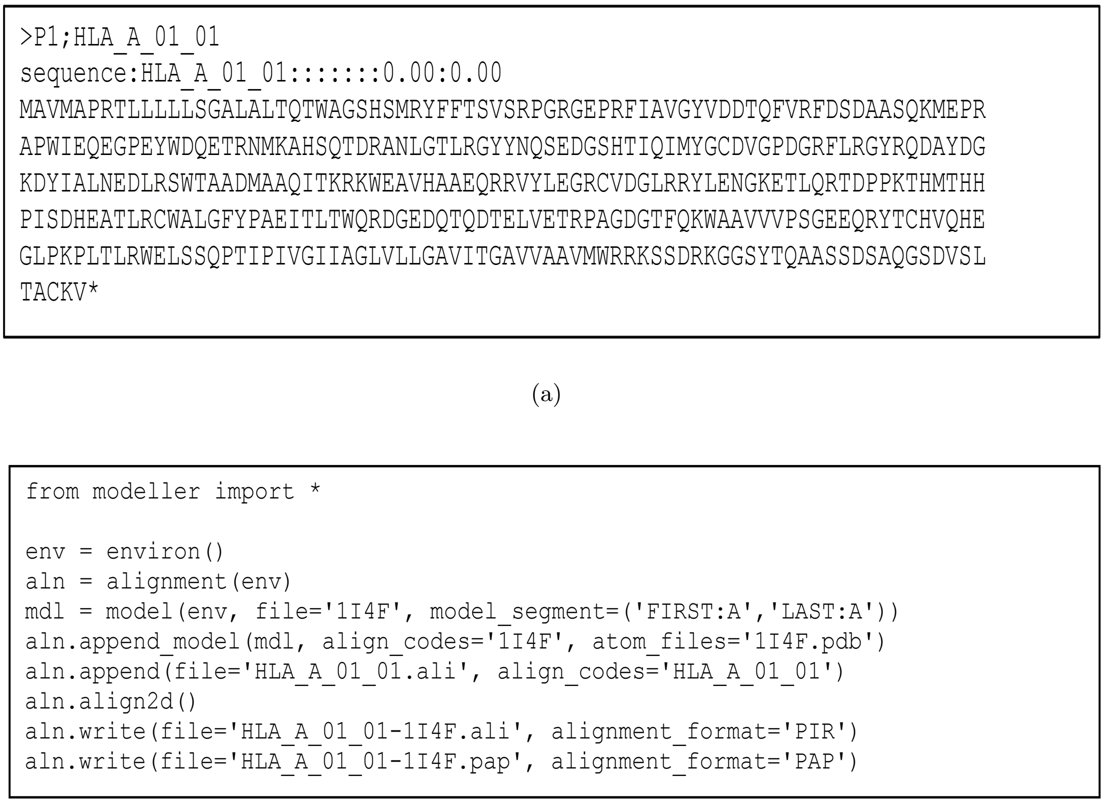
A.1.4 The Sequence to the Structure Alignment

The fourth step of the homology modelling process is to align target amino acid sequences with the sequence of template structure. This task requires three files: the target sequence file (in PIR format as shown in Figure A.4(a)), the selected template structure file (in PDB format) and a Modeller script that uses 'align2d' function[2] to perform sequence-to-structure alignment (as shown in Figure A.4(b)). Modeller is a command line program. It can process only one alignment script at a time. Therefore for aligning more than one sequence with the selected template structure, a set of programs were written in Java to automate the process of preparing the required alignment files.

A.1.5 The Removal of Unwanted Residues from Alignment File

Before generating and executing scripts for predicting structures, it is required to check that if after sequence-to-structure alignment any gaps that were added in the sequence of amino acids of template structure at the start and/or at the end in the alignment file are removed from template structure sequence. The same numbers of residues are removed from the start and/or at the end of the target sequence. This is done to predict the most similar structures for a number of sequences by ignoring all the residues in

[2] This function uses a variable gap penalty function which tends to align in better structural context gaps in the sequence-to-structure alignment file. Details of the variable gap penalty function are provided at http://salilab.org/modeller/manual/node288.html.



```
>P1;HLA_A_01_01
sequence:HLA_A_01_01::::::0.00:0.00
MAVMAPRTLLLLLSGALALTQTWAGSHSMRYFFTSVSRPGRGEPRFIAVGYVDDTQFVRFDSDAASQKMEPR
APWIEQEGPEYWDQETRNMKAHSQTDRANLGTLRGYYNQSEDGSHTIQIMYGCDVGPDGRFLRGYRQDAYDG
KDYIALNEDLRSWTAADMAAQITKRKWEAVHAAEQRRVYLEGRCVDGLRRYLENGKETLQRTDPPKTHMTHH
PISDHEATLRCWALGFYPAEITLTWQRDGEDQTQDTELVETRPAGDGTFQKWAAVVVPSGEEQRYTCHVQHE
GLPKPLTLRWELSSQPTIPIVGIIAGLVLLGAVITGAVVAAVMWRRKSSDRKGGSYTQAASSDSAQGSDVSL
TACKV*
```

(a)

```
from modeller import *

env = environ()
aln = alignment(env)
mdl = model(env, file='1I4F', model_segment=('FIRST:A','LAST:A'))
aln.append_model(mdl, align_codes='1I4F', atom_files='1I4F.pdb')
aln.append(file='HLA_A_01_01.ali', align_codes='HLA_A_01_01')
aln.align2d()
aln.write(file='HLA_A_01_01-1I4F.ali', alignment_format='PIR')
aln.write(file='HLA_A_01_01-1I4F.pap', alignment_format='PAP')
```

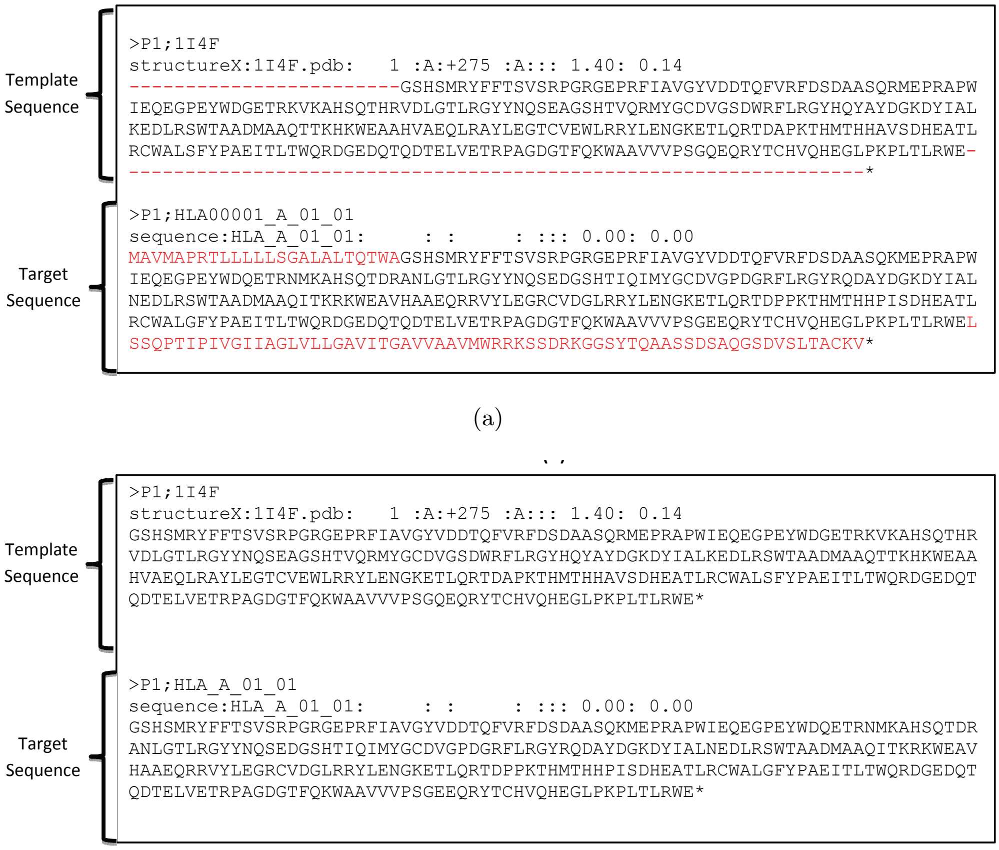
(b)

Figure A.4: (a) An input target sequence file in the PIR format. (b) The modeller script for aligning the target sequence with the template structure sequence.

the target sequence that are not known in the sequence of the given template structure. Figure A.5(a) shows such gaps marked as red in start and end of the alignment file for target and template sequence generated as a result of using alignment method provided by the Modeller software whereas Figure A.5(b) shows the alignment file after removing such marked gaps as red (as shown in Figure A.5(a)) in the template sequence and residues in the target sequence. A program named 'ImproveAlign' is written in Java that does this job for selected file(s) generated in result of sequence to structure alignment process.

A.1.6 The Three-Dimensional Structure Prediction

This step of homology modelling requires a sequence-to-structure alignment file (.ali file), template structure file (.pdb file) and a script with some parameters to generate a three-dimensional structure for the target sequence. The Modeller tool takes this single script as input and generates a three-dimensional structure based on the parameters set in the script. To automate the process for predicting structure for multiple target sequences, a program was written in Java that generates scripts for three-dimensional structure gen-


Template Sequence
Target Sequence
Template Sequence
Target Sequence

```
>P1;1I4F
structureX:1I4F.pdb:   1 :A:+275 :A::: 1.40: 0.14
------------------------GSHSMRYFFTSVSRPGRGEPRFIAVGYVDDTQFVRFDSDAASQRMEPRAPW
IEQEGPEYWDGETRKVKAHSQTHRVDLGTLRGYYNQSEAGSHTVQRMYGCDVGSDWRFLRGYHQYAYDGKDYIAL
KEDLRSWTAADMAAQTTKHKWEAAHVAEQLRAYLEGTCVEWLRRYLENGKETLQRTDAPKTHMTHHAVSDHEATL
RCWALSFYPAEITLTWQRDGEDQTQDTELVETRPAGDGTFQKWAAVVVPSGQEQRYTCHVQHEGLPKPLTLRWE-
-----------------------------------------------------------------*

>P1;HLA00001_A_01_01
sequence:HLA_A_01_01:     : :     : ::: 0.00: 0.00
MAVMAPRTLLLLLSGALALTQTWAGSHSMRYFFTSVSRPGRGEPRFIAVGYVDDTQFVRFDSDAASQKMEPRAPW
IEQEGPEYWDQETRNMKAHSQTDRANLGTLRGYYNQSEDGSHTIQIMYGCDVGPDGRFLRGYRQDAYDGKDYIAL
NEDLRSWTAADMAAQITKRKWEAVHAAEQRRVYLEGRCVDGLRRYLENGKETLQRTDPPKTHMTHHPISDHEATL
RCWALGFYPAEITLTWQRDGEDQTQDTELVETRPAGDGTFQKWAAVVVPSGEEQRYTCHVQHEGLPKPLTLRWEL
SSQPTIPIVGIIAGLVLLGAVITGAVVAAVMWRRKSSDRKGGSYTQAASSDSAQGSDVSLTACKV*
```

(a)

```
>P1;1I4F
structureX:1I4F.pdb:   1 :A:+275 :A::: 1.40: 0.14
GSHSMRYFFTSVSRPGRGEPRFIAVGYVDDTQFVRFDSDAASQRMEPRAPWIEQEGPEYWDGETRKVKAHSQTHR
VDLGTLRGYYNQSEAGSHTVQRMYGCDVGSDWRFLRGYHQYAYDGKDYIALKEDLRSWTAADMAAQTTKHKWEAA
HVAEQLRAYLEGTCVEWLRRYLENGKETLQRTDAPKTHMTHHAVSDHEATLRCWALSFYPAEITLTWQRDGEDQT
QDTELVETRPAGDGTFQKWAAVVVPSGQEQRYTCHVQHEGLPKPLTLRWE*

>P1;HLA_A_01_01
sequence:HLA_A_01_01:     : :     : ::: 0.00: 0.00
GSHSMRYFFTSVSRPGRGEPRFIAVGYVDDTQFVRFDSDAASQKMEPRAPWIEQEGPEYWDQETRNMKAHSQTDR
ANLGTLRGYYNQSEDGSHTIQIMYGCDVGPDGRFLRGYRQDAYDGKDYIALNEDLRSWTAADMAAQITKRKWEAV
HAAEQRRVYLEGRCVDGLRRYLENGKETLQRTDPPKTHMTHHPISDHEATLRCWALGFYPAEITLTWQRDGEDQT
QDTELVETRPAGDGTFQKWAAVVVPSGEEQRYTCHVQHEGLPKPLTLRWE*
```

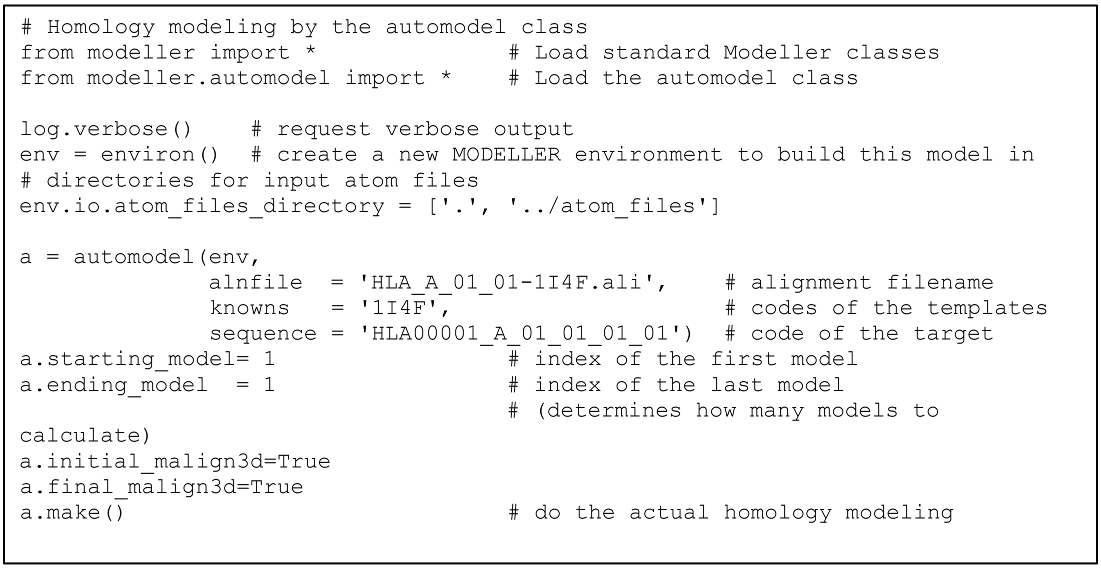
(b)

Figure A.5: (a) An example alignment file (gaps in the start and end of the template sequence marked as red and the corresponding amino acid residues are marked as red in the start and end of the target sequence.) (b) Alignment file after removing gaps marked as red in (a).

eration for selected alignment sequence file(s). This program also generates a batch file which on execution generates three-dimensional structures for the target sequences with well aligned backbone for all the sequences whose structure was unknown. A sample script to predict the three-dimensional structure of one of the target sequence of HLA-A is shown in Figure A.6.

A.1.7 The Prediction of Side Chain Conformation

Prediction of side chains is an important step in protein structure prediction. For closely related protein structures, very little change is often observed in the backbone. Therefore, prediction of side chain conformations can accomplish the process of structure prediction (Veenstra and Kollman, 1997). We used a program called SCWRL4 (Krivov et al., 2009) that predicts side chains of protein structures based on statistical rotamer



```
# Homology modeling by the automodel class
from modeller import *              # Load standard Modeller classes
from modeller.automodel import *    # Load the automodel class

log.verbose()    # request verbose output
env = environ()  # create a new MODELLER environment to build this model in
# directories for input atom files
env.io.atom_files_directory = ['.', '../atom_files']

a = automodel(env,
              alnfile  = 'HLA_A_01_01-1I4F.ali',    # alignment filename
              knowns   = '1I4F',                    # codes of the templates
              sequence = 'HLA00001_A_01_01_01_01')  # code of the target
a.starting_model= 1                 # index of the first model
a.ending_model  = 1                 # index of the last model
                                    # (determines how many models to
calculate)
a.initial_malign3d=True
a.final_malign3d=True
a.make()                            # do the actual homology modeling
```

Figure A.6: A sample script for predicting structure of the target sequence.

library[3] (Dunbrack, 2002). Figure A.7 demonstrates a predicted protein three-dimensional structures before and after predicting side chain conformations using this tool. SCWRL4 is also a command line program to process a single structure at a time. Therefore again a program was written in Java to generate a batch file for using SCWRL4 for processing more than one file at one time.

A.2 The Electrostatic Potential Map Calculation Process

Computing electrostatic potential of a protein structure using a software tool, called the Adaptive Poisson Boltzmann Solver (APBS) (Baker et al., 2001), is a two-step process as explained in Figure A.8.

A.2.1 The PDB Structure to the PQR Structure Conversion

Protein data bank structures in PDB[4] format file or the predicted structures from these three-dimensional structure often misses hydrogen atoms and also misses a fraction of some of the heavy atom coordinates. The process of adding missing hydrogen and predicting missing heavy atoms can be achieved using a software tool called PDB2PQR (Dolinsky et al., 2007). PDB2PQR also replaces some of the parameters in the PDB format file

[3] A statistical rotamer library uses conformer libraries which are samples of side chains of known protein three-dimensional structures quite often in the form of Cartesian coordinates.

[4] A PDB format file is a standard method of representation of three-dimensional protein structure data obtained from X-ray crystallography or NMR methods and details are provided at http://www.rcsb.org/pdb/static.do?p=file_formats/pdb/index.html

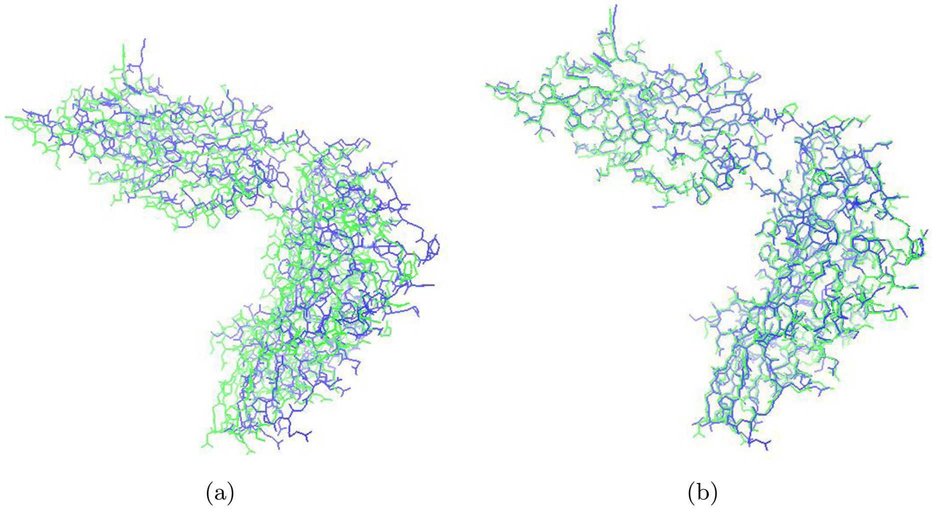


Figure A.7: Demonstration of prediction of side chain conformations for two three-dimensional protein structures (i.e. $HLA\_A\_01\_01$ as blue and $HLA\_A\_01\_02$ as green). (a) Superimpose structures before predicting side chain conformations. (b) Superimpose structures after predicting side chain conformations.

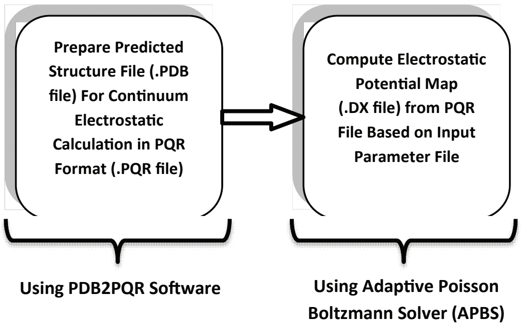

Prepare Predicted Structure File (.PDB file) For Continuum Electrostatic Calculation in PQR Format (.PQR file)
Compute Electrostatic Potential Map (.DX file) from PQR File Based on Input Parameter File
Using PDB2PQR Software
Using Adaptive Poisson Boltzmann Solver (APBS)

Figure A.8: Electrostatic potential map calculation process.

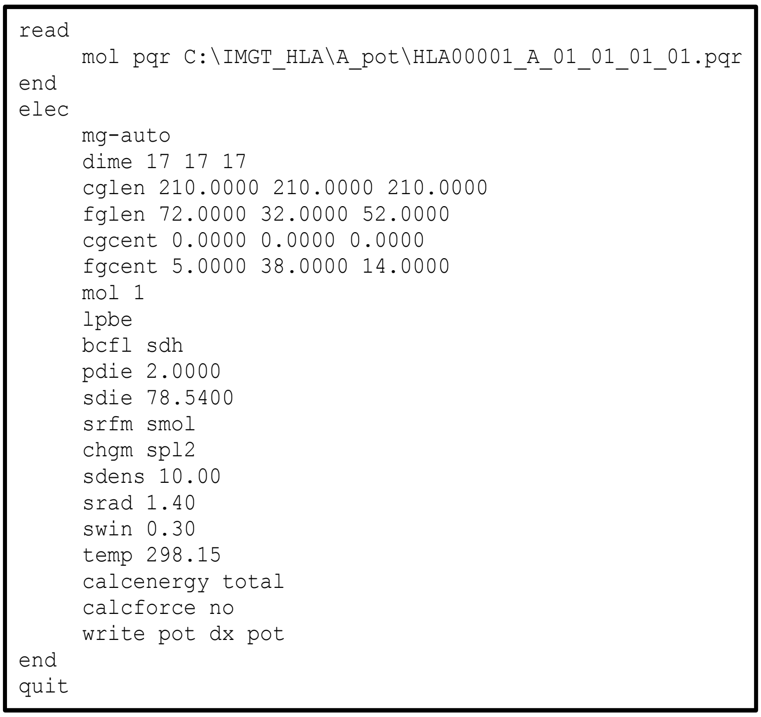
while converting PDB to PQR[5] format file. These parameters are required parameters for APBS software to calculate electrostatic potential energies. PDB2PQR reads these force fields from already available data files such as AMBER99 (Wang et al., 2000). Both web based and command line version of PDB2PQR are available and both can process only one structure file at a time. We used a command-line version.

A.2.2 The Electrostatic Potential Map Calculation

Adaptive Poisson Boltzmann Solver (APBS) is based on the FEtk (Finite Element Toolkit), an adaptive finite-element based modelling C++ class library for solving non-linear partial differential equations. The APBS tool requires an input script for calculating electrostatic

[5] A PQR format file is a most popular method to include parameters in a PDB format file by replacing some of parameters such as occupancy field ('P') is replaced with atomic charge ('Q') and temperature factor column is replaced with the radius ('R').

potential energy between molecular solutes and solvents such as water. In the input script, grid size (three-dimensional lattice) and position are set based on the region of interest of the three-dimensional structure. Both web-based and command-line versions are available but we used the command line version. A sample script with parameters of APBS which we used in experiments is shown Figure A.9. Executing this script generates a file that contains the electrostatic potential value at each grid position. The electrostatic potential value at each grid position represents a single variable. Therefore, assuming a grid of 17 in each direction for a protein structure will have 17^3 variables to represent a single protein.



```
read
     mol pqr C:\IMGT_HLA\A_pot\HLA00001_A_01_01_01_01.pqr
end
elec
     mg-auto
     dime 17 17 17
     cglen 210.0000 210.0000 210.0000
     fglen 72.0000 32.0000 52.0000
     cgcent 0.0000 0.0000 0.0000
     fgcent 5.0000 38.0000 14.0000
     mol 1
     lpbe
     bcfl sdh
     pdie 2.0000
     sdie 78.5400
     srfm smol
     chgm spl2
     sdens 10.00
     srad 1.40
     swin 0.30
     temp 298.15
     calcenergy total
     calcforce no
     write pot dx pot
end
quit
```

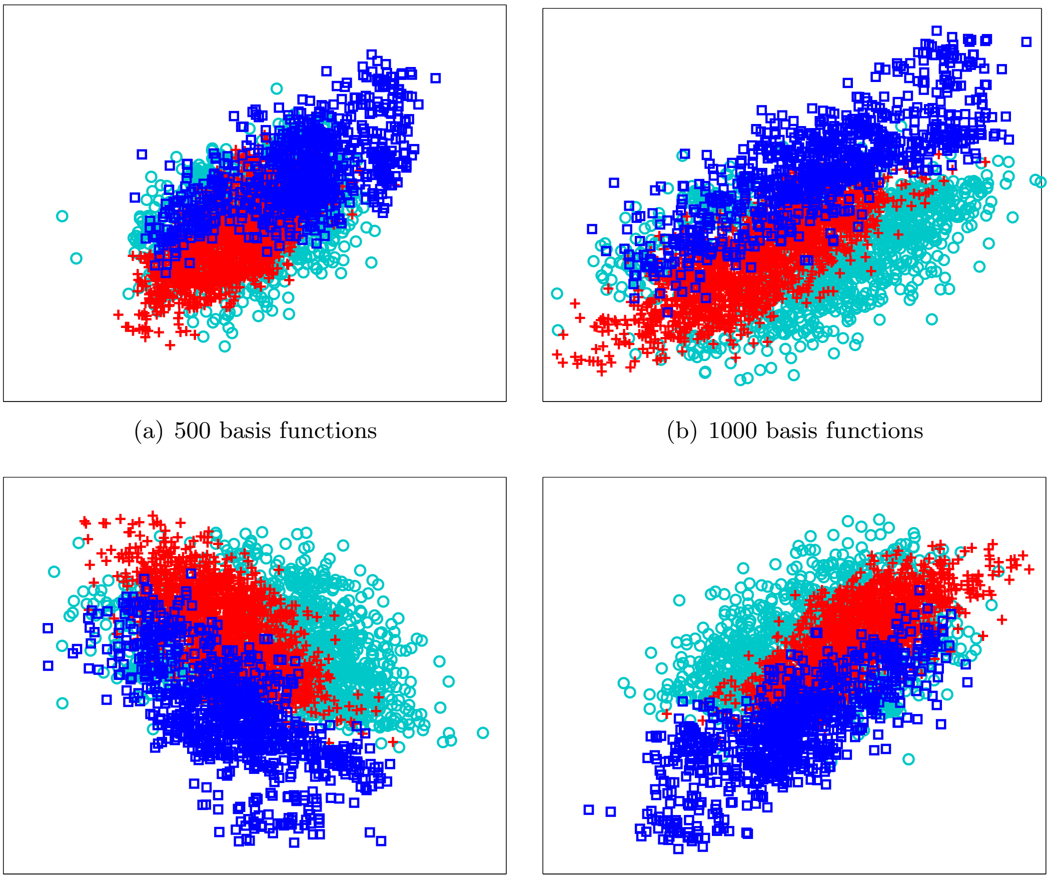
Figure A.9: A sample APBS input script.

A.2.3 An Automation of Electrostatic Potential Map Calculation

PDB2PQR and APBS software tools are available both as web-based and command line version. We used a command line version of both the tools and both can process one file at a time. Therefore, a Graphical User Interface (GUI) was designed in Java to automate the process for generating batch files for both the processes.

B The Neuroscale visualisations of MHC class-I dataset

This appendix show some visualisation of MHC class-I dataset using Neuroscale with different number of basis functions to observe any significant impact on the visualisation. As explained in section 3.3.2, usually it is more appropriate to use number of basis functions equal to the number of data points in the dataset. However we observe that for the MHC class-I data repeating the experiments with different number of basis function have not made much improvement in the results (see Figure B.1).



(a) 500 basis functions

(b) 1000 basis functions

(c) 2000 basis functions

(d) 3000 basis functions

Figure B.1: The Neuroscale visualisations of the MHC class-I dataset with different number of basis functions.

C The LTM-FS Visualisation model, EM derivation of the LTM-FS, and additional visualisation results of the LTM-FS model.

C.1 The Mixture Model for the GTM-FS/LTM-FS

The conditional density of $\mathbf{x}_n$ given the $\Psi = \{\psi_d, \cdots, \psi_D\}$ is defined as (this is same as equation (3.27)),

$$p(\mathbf{x}_n|\Psi) = \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} [p(x_{nd}|\theta_{kd})]^{\psi_d} [q(x_{nd}|\lambda_d)]^{(1-\psi_d)}. \tag{C.1}$$

We take Ψ as a set of missing variables and take feature saliency as $\rho_d = p(\psi_d = 1)$, $d = 1, \ldots, D$, as a set of parameters to be estimated. We consider that the ψ_ds are mutually independent and also independent of the hidden component label y for any data pattern $\mathbf{x}$. Considering

$$\begin{aligned}
p(\mathbf{x}_n, \Psi) &= p(\mathbf{x}_n|\Psi)p(\Psi) \\
&= \left(\sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} [p(x_{nd}|\theta_{kd})]^{\psi_d}[q(x_{nd}|\lambda_d)]^{(1-\psi_d)}\right) \prod_{d=1}^{D} \rho_d^{\psi_d}(1-\rho_d)^{1-\psi_d} \\
&= \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} [\rho_d p(x_{nd}|\theta_{kd})]^{\psi_d}[(1-\rho_d)q(x_{nd}|\lambda_d)]^{(1-\psi_d)}.
\end{aligned} \tag{C.2}$$

The marginal density of $\mathbf{x}_n$ takes the form

$$\begin{aligned}
p(\mathbf{x}_n) &= \sum_{psi=0}^{1} p(\mathbf{x}_n, \Psi) \\
&= \sum_{k=1}^{K} \pi_k \sum_{\psi=0}^{1} \prod_{d=1}^{D} [\rho_d p(x_{nd}|\theta_{kd})]^{\psi_d}[(1-\rho_d)q(x_{nd}|\lambda_d)]^{(1-\psi_d)} \\
&= \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} \sum_{\psi_d=0}^{1} [\rho_d p(x_{nd}|\theta_{kd})]^{\psi_d}[(1-\rho_d)q(x_{nd}|\lambda_d)]^{(1-\psi_d)}.
\end{aligned} \tag{C.3}$$

This is equation (3.28) and it is important to note that the features are independent of the component label y.

C.2 Deriving the EM algorithm of the GTM/LTM -FS

From equation (3.28), the complete-data log-likelihood is defined as,

$$p(\mathbf{x}_n|\Omega) = \pi_k \left[\left[\prod_{d=1}^{D} [\rho_d p(x_{nd}|\Theta_{kd})] + [(1-\rho_d)q(x_{nd}|\lambda_d)]\right]\right]. \tag{C.4}$$

By defining the following terms

$$r_{nk} = p(y_n = k|\mathbf{x}_n). \tag{C.5}$$

$$u_{nkd} = p(\psi_d = 1, y_n == k|\mathbf{x}_n). \tag{C.6}$$

$$v_{nkd} = p(\psi_d = 0, y_n == k|\mathbf{x}_n). \tag{C.7}$$

All these quantities are calculated using the current set of parameters $\Omega^{current}$. It is important to note that $u_{nkd} + v_{nkd} = r_{nk}$ and $\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} = N$. The expected log-likelihood using the current set of parameters $\Omega^{current}$ we get

$$\begin{aligned}
&E_{\Omega^{current}}[\ln P(\mathbf{X},\mathbf{y},\Psi)] \\
&= \sum_{n,k,\Psi} p(y_n = k, \Psi|\mathbf{x}_n) \left(\ln \pi_k + \sum_d \Big(\psi_d \left(\ln p(x_{nd}|\theta_{kd}) + \ln \rho_d\right) \right. \\
&\quad \left. + (1-\psi_d)(\ln q(x_{nd}|\lambda_d) + \ln(1-\rho_d))\Big) \right) \\
&= \sum_{n,k} p(y_n = k|\mathbf{x}_i) \ln \pi_k + \sum_{n,k}\sum_d \sum_{\psi_d=0}^{1} p(y_n = k, \psi_d|\mathbf{x}_n) \\
&\quad \Big(\psi_d \left(\ln p(x_{nd}|\theta_{kd}) + \ln \rho_d\right) + (1-\psi_d)(\ln q(x_{nd}|\lambda_d) + \ln(1-\rho_d))\Big) \qquad \text{(C.8)} \\
&= \underbrace{\sum_k (\sum_n r_{nk}) \ln \pi_k}_{\text{part 1}} + \left(\sum_{k,d} \underbrace{\sum_n u_{nkd} \ln p(x_{nd}|\theta_{kd})}_{\text{part 2}} + \sum_d \underbrace{\sum_{n,k} v_{nkd} \ln q(x_{nd}|\lambda_d)}_{\text{part 3}} \right. \\
&\quad \left. + \sum_d \left(\underbrace{\ln \rho_d \sum_{n,k} u_{nkd} + \ln(1-\rho_d) \sum_{n,k} v_{nkd}}_{\text{part 4}} \right) \right)
\end{aligned}$$

where ρ_d is the saliency of the dth feature, $p(x_{nd}|m_{kd})$ represent the probability density function of the dth feature for the kth component, and $q(x_{nd}|\lambda_d)$ is the common background density. Each part of the equation (C.8) can be maximised separately.

M-step for GTM-FS case:

The 2nd term of the equation (C.8) is represented as

$$\begin{aligned}
\mathcal{L}_{2ndpart} &= \sum_{kd} \sum_n u_{nkd} \ln p(x_{nd}|\theta_{kd}) \\
\mathcal{L}_{2ndpart} &= \sum_{kd} \sum_n u_{nkd} \ln \left\{ \sqrt{\frac{\beta}{2\pi}} \exp\left\{ -\frac{\beta||x_{nd} - \boldsymbol{\phi}(\mathbf{z}_k)\mathbf{w}_d||^2}{2} \right\} \right\} \qquad \text{(C.9)} \\
\mathcal{L}_{2ndpart} &= \sum_{kd} \sum_n u_{nkd} \left\{ \frac{1}{2} \ln\left(\frac{\beta}{2\pi}\right) - \frac{\beta||x_{nd} - \boldsymbol{\phi}(\mathbf{z}_k)\mathbf{w}_d||^2}{2} \right\}
\end{aligned}$$

Now differentiating equation (C.9) with respect to w_{ld}, (i.e. $l = 1, \cdots, L$), we get

$$\frac{\partial \mathcal{L}_{2ndpart}}{\partial w_{ld}} = \sum_k \sum_n u_{nkd} \left[\beta \left(x_{nd} - \boldsymbol{\Phi}(\mathbf{z}_k)\mathbf{w}_d\right) \boldsymbol{\phi}_l(\mathbf{z}_k) \right] \qquad \text{(C.10)}$$

and setting above equation equal to 0 and solving it we get

$$\sum_k \sum_n u_{nkd} \Big[\beta \left(x_{nd} - \boldsymbol{\Phi}(\mathbf{z}_k)\mathbf{w}_d \right) \phi_l(\mathbf{z}_k) \Big] = 0. \tag{C.11}$$

This can be represented in matrix notation form as

$$\boldsymbol{\Phi}^T \mathbf{E}_d \boldsymbol{\Phi} \hat{\mathbf{w}}_d = \boldsymbol{\Phi}^T \mathbf{U}_d \mathbf{x}_d, \tag{C.12}$$

where $\boldsymbol{\Phi}$ is a $K \times L$ matrix, $\mathbf{U}_d$ is a $K \times N$ matrix computed using equation (3.31), $\mathbf{x}_d$ is a $N \times 1$ data vector and a diagonal matrix $\mathbf{E}_d$ can take the values

$$e_{kkd} = \sum_{n=1}^{N} u_{nkd}. \tag{C.13}$$

Now we can re-estimate the centres of Gaussians using equation (3.35) and similarly differentiating with respect to β_d we get equation (3.36) to re-estimate β_d. Similary re-estimation of the other parameters are dicussed in Section 3.3.4.1.

M-step for LTM-FS case:

Now differentiating 2nd part of equation (C.8) with respect to w_{ld} and using equation (5.3)

$$\frac{\partial \mathcal{L}_{2ndpart}}{\partial w_{ld}} = \sum_k \sum_n u_{nkd} \Big[x_{nd} - g((\boldsymbol{\Phi}(\mathbf{z}_k))^T \mathbf{w}_d) \Big] \boldsymbol{\Phi}_l(\mathbf{z}_k), \tag{C.14}$$

and this can be written in the matrix notation as

$$\frac{\partial \mathcal{L}_{2ndpart}}{\partial \mathbf{w}_d} = \boldsymbol{\Phi}^T \left[\mathbf{U}_d \mathbf{x}_d - Eg((\boldsymbol{\Phi}(\mathbf{Z}))\mathbf{w}_d) \right] \tag{C.15}$$

where $\boldsymbol{\Phi}$ is a $K \times L$ matrix, $\mathbf{w}_d$ is a $L \times 1$ weight vector, $\mathbf{U}_d$ is a $K \times N$ matrix obtained from equation (5.11), $\mathbf{x}_d$ is a $N \times 1$ data vector, and $\mathbf{E}_d$ is a $K \times K$ diagonal matrix with elements

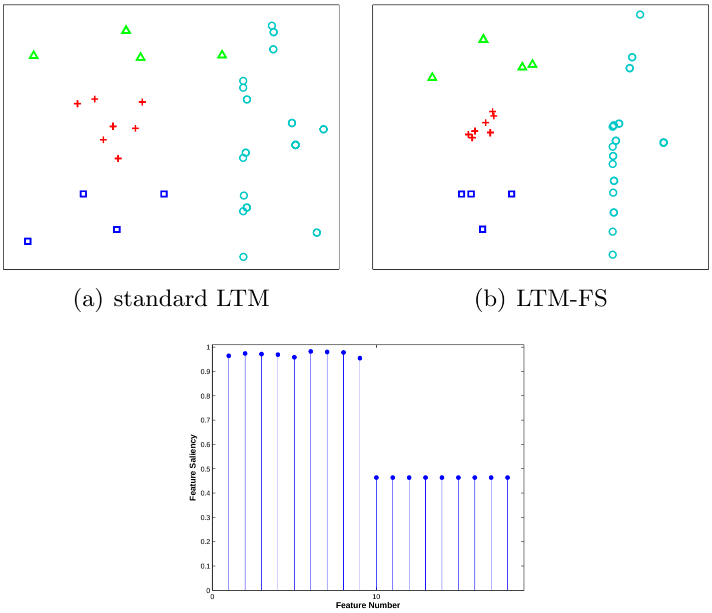
$$e_{kkd} = \sum_{n=1}^{N} u_{nkd}. \tag{C.16}$$

Re-estimation of the other parameters in $p(.)$ and $q()$ are dicussed in Section 5.4.1.

C.3 The LTM and LTM-FS additional visualisation results with different latent and RBF grid sizes

This section explains some of the additional results using different latent grid and RBF grid sizes. We here explains results using LTM and LTM-FS with different grid sizes for synthetic binary dataset-I and -II (as discussed in Section 5.5.1) and MHC class-I binary dataset (as discussed in Section 5.5.3)

- Latent grid: 12×12 and RBF grid: 8×8 (see Figures C.1 and C.6 for synthetic dataset-I and II respectively and see Figure C.11 for MHC binary dataset).
- Latent grid: 12×12 and RBF grid: 4×4 (see Figures C.2 and C.7 for synthetic dataset-I and II respectively and see Figure C.12 for MHC binary dataset).
- Latent grid: 12×12 and RBF grid: 2×2 (see Figures C.3 and C.8 for synthetic dataset-I and II respectively and see Figure C.13 for MHC binary dataset).
- Latent grid: 8×8 and RBF grid: 2×2 (see Figures C.4 and C.9 for synthetic dataset-I and II respectively and see Figure C.14 for MHC binary dataset).
- Latent grid: 4×4 and RBF grid: 2×2 (see Figures C.5 and C.10 for synthetic dataset-I and II respectively and see Figure C.15 for MHC binary dataset).

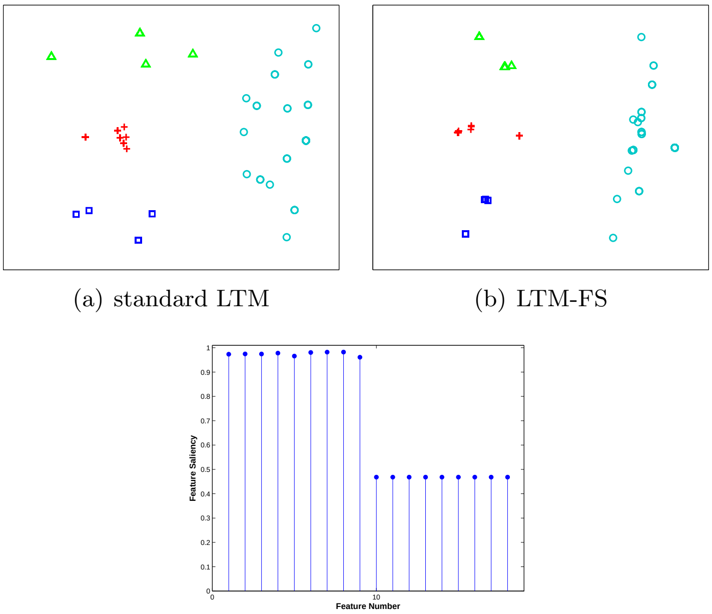


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.1: The LTM and the LTM-FS visualisations of the binary synthetic dataset-I using a latent grid size 12 × 12 and RBF grid size 8 × 8. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

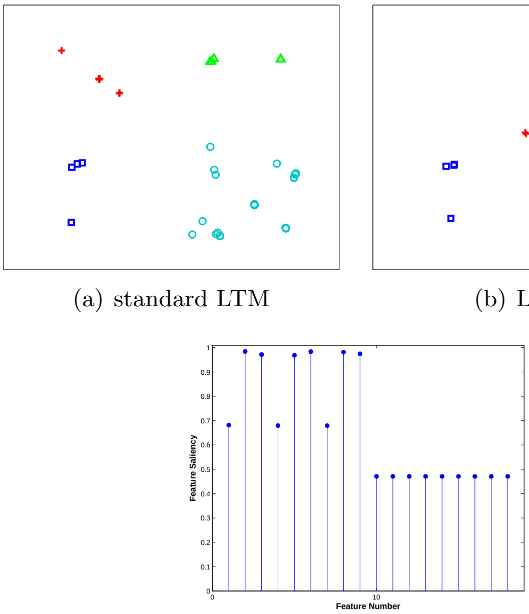


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.2: The standard LTM and the LTM-FS visualisations of the binary synthetic dataset-I using a latent grid size 12 × 12 and an RBF grid size 4 × 4. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

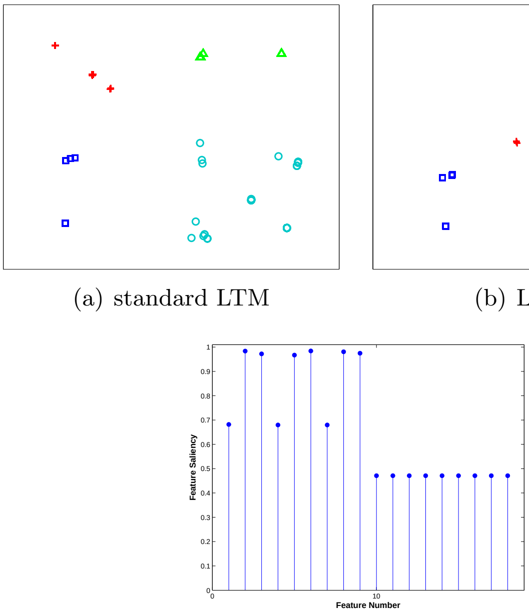


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.3: The LTM and the LTM-FS visualisations of the binary synthetic dataset-I using a latent grid size 12×12 and RBF grid size 2×2. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

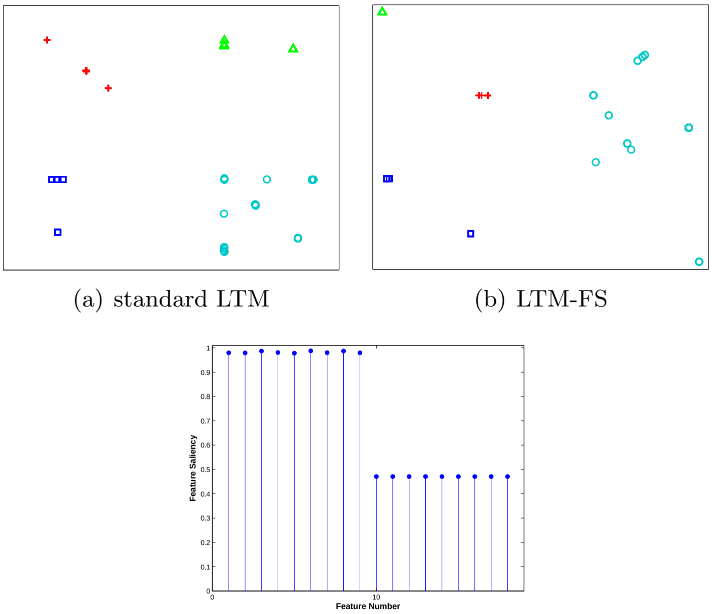


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.4: The LTM and the LTM-FS visualisations of the binary synthetic dataset-I using a latent grid size 8×8 and an RBF grid size 2×2. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

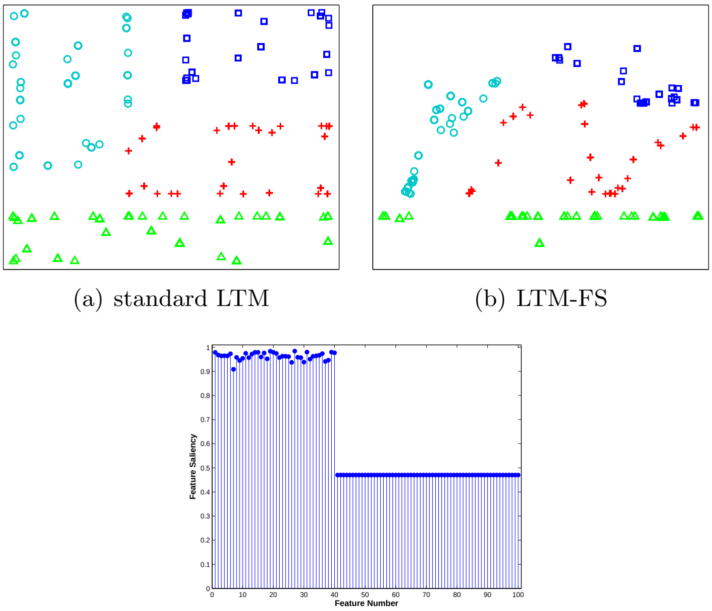


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.5: The LTM and the LTM-FS visualisations of the binary synthetic dataset-I using a latent grid size 4×4 and an RBF grid size 2×2. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

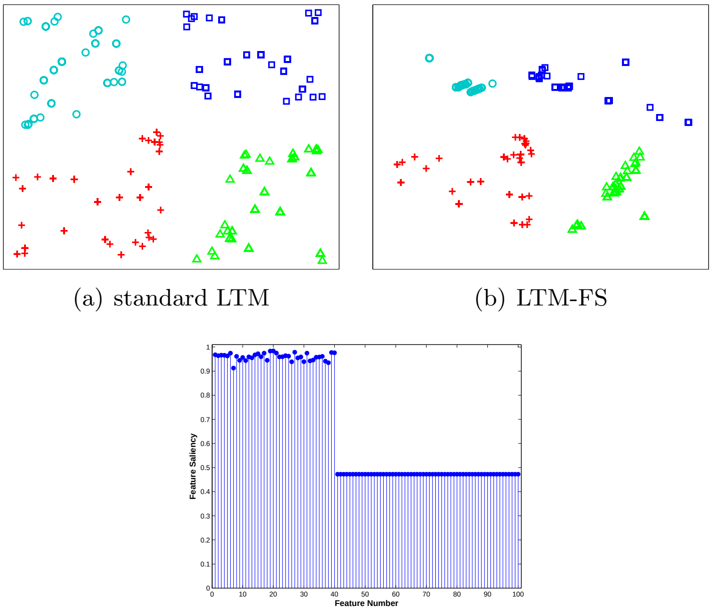


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.6: The LTM and the LTM-FS visualisations of the binary synthetic dataset-II using the latent grid 12×12 and an RBF grid size 8×8. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

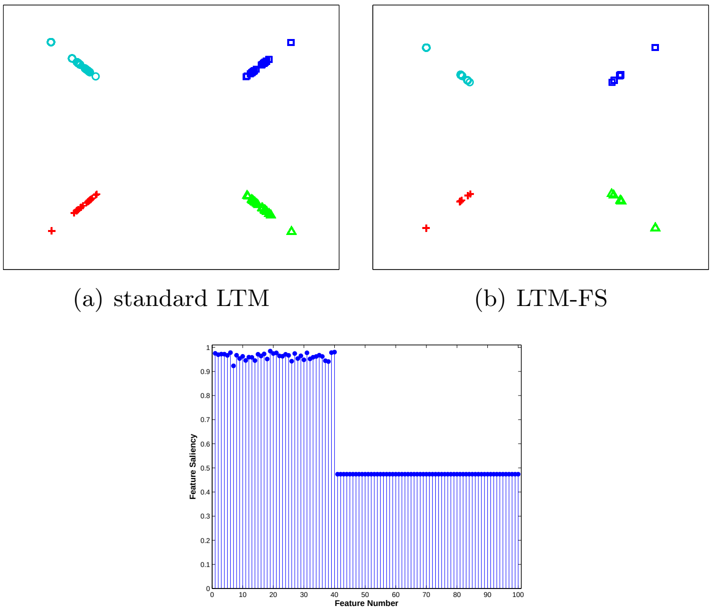


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.7: The LTM and the LTM-FS visualisations of the binary synthetic dataset-II using a latent grid size 12 × 12 and an RBF grid size 4 × 4. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

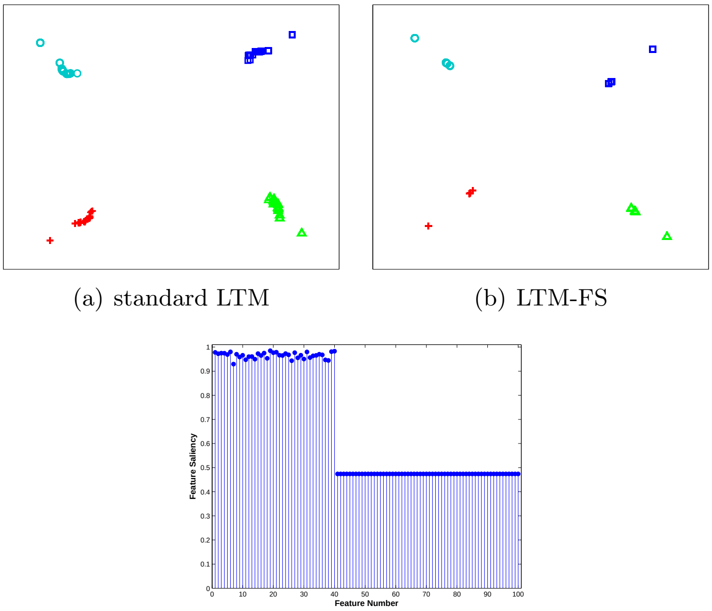


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.8: The LTM and the LTM-FS visualisations of the binary synthetic dataset-II using a latent grid 12 × 12 and an RBF grid size 2 × 2. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

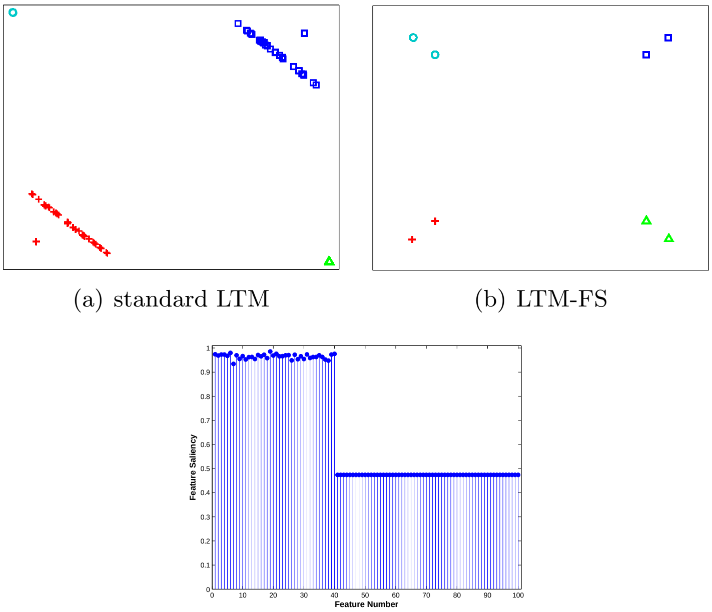


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.9: The LTM and the LTM-FS visualisations of the binary synthetic dataset-II using a latent grid size 8×8 and an RBF grid size 2×2. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

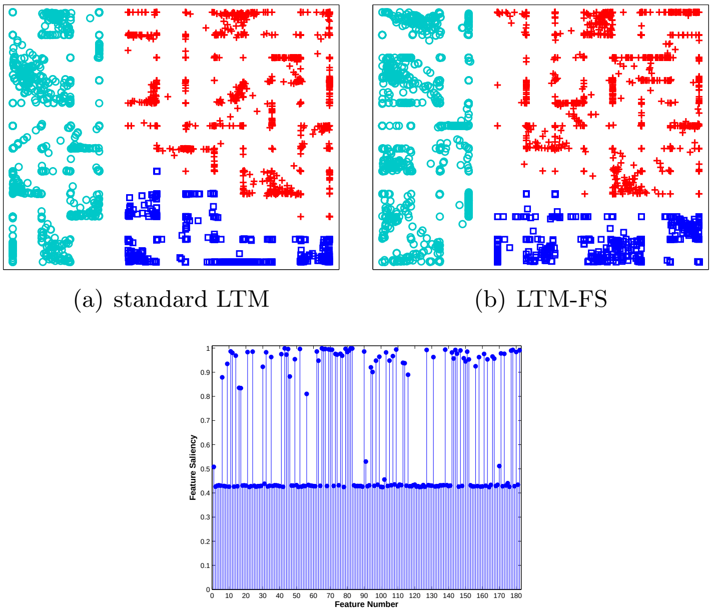


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.10: The LTM and the LTM-FS visualisations of the binary synthetic dataset-II using a latent grid size 4×4 and an RBF grid size 2×2. The LTM-FS visualisation in (b) show better results with compact cluster for each class compared to the LTM visualisation in (a) and (f) shows the estimated feature saliencies from the LTM-FS.

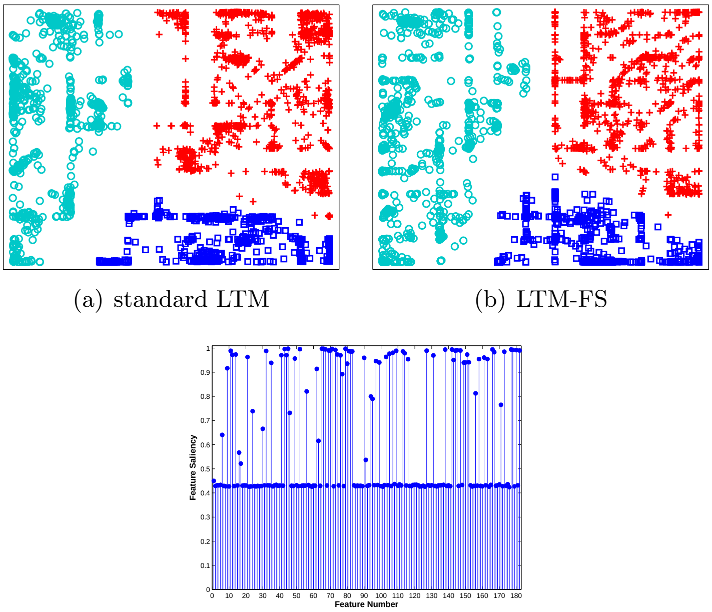


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.11: The LTM and the LTM-FS visualisations of the MHC class-I sequence-based binary dataset using a latent grid size 12 × 12 and an RBF grid size 8 × 8. The data points shown as cyan circles represent alleles of HLA-A, red plus signs for HLA-B and blue squares for HLA-C. Both the LTM and the LTM-FS visualisations (i.e. (a) and (b) respectively) have shown clear separation between three classes (i.e. genes) of MHC class-I dataset hence it is difficult to visually observe better clustering structure from both. Feature saliencies estimated from the LTM-FS are shown in (c) on a scale of 0-to-1.

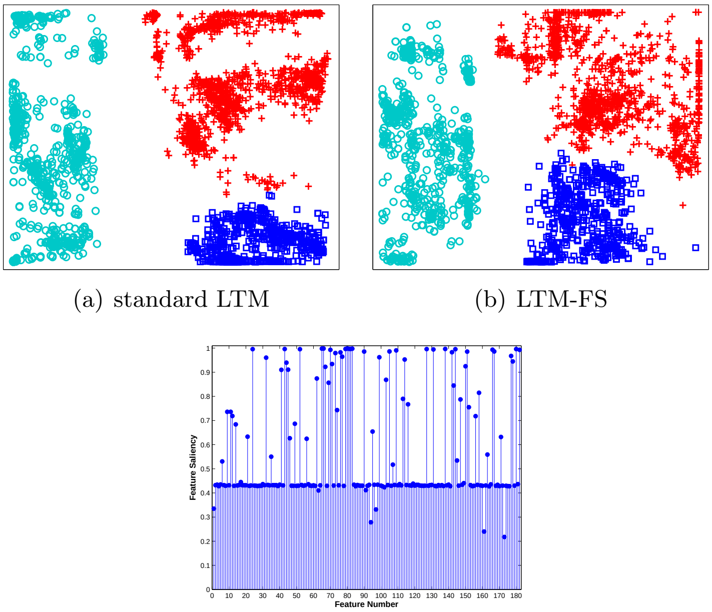


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.12: The LTM and the LTM-FS visualisations of the MHC class-I sequence-based binary dataset using a latent grid size 12×12 and an RBF grid size 4×4. The data points shown as cyan circles represent alleles of HLA-A, red plus signs for HLA-B and blue squares for HLA-C. Both LTM and LTM-FS visualisations (i.e. (a) and (b) respectively) have shown clear separation between three classes (i.e. genes) of an MHC class-I dataset hence it is difficult to visually observe better clustering structure from both. Feature saliencies estimated from the LTM-FS are shown in (c) on a scale of 0-to-1.

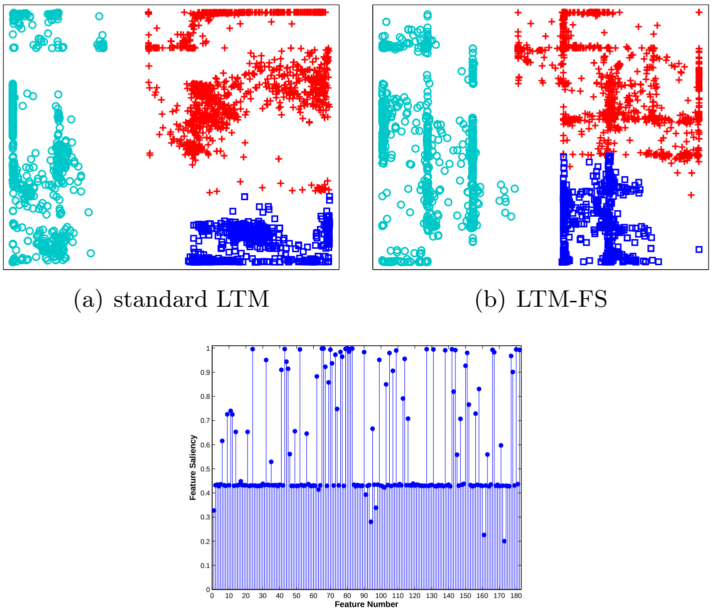


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.13: The LTM and the LTM-FS visualisations of the MHC class-I sequence-based binary dataset using a latent grid size 12×12 and an RBF grid size 2×2. The data points shown as cyan circles represent alleles of HLA-A, red plus signs for HLA-B and blue squares for HLA-C. Both LTM and LTM-FS visualisations (i.e. (a) and (b) respectively) have shown clear separation between three classes (i.e. genes) of MHC class-I hence it is difficult to visually observe better clustering structure from both. Feature saliencies estimated from the LTM-FS are shown in (c) on a scale of 0-to-1.

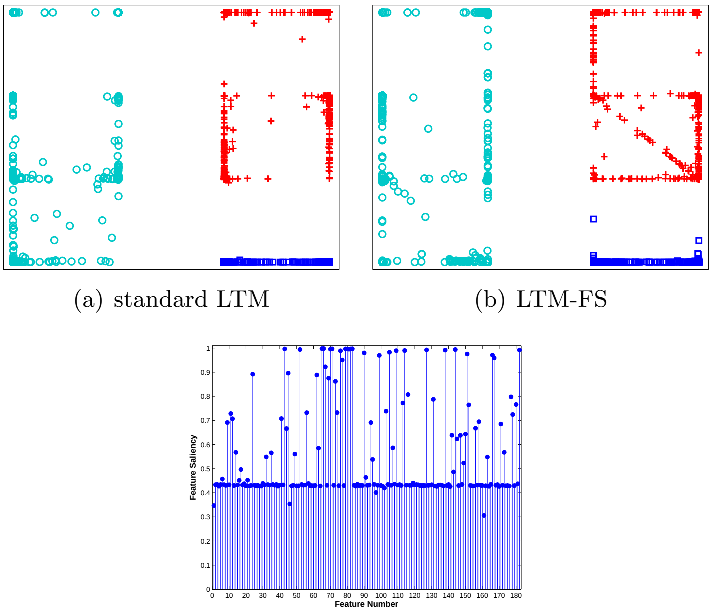


(a) standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.14: The LTM and the LTM-FS visualisations of an MHC class-I sequence-based binary dataset using a latent grid 8×8 and an RBF grid size 2×2. The data points shown as cyan circles represent alleles of HLA-A, red plus signs for HLA-B and blue squares for HLA-C. Both the LTM and the LTM-FS visualisations (i.e. (a) and (b) respectively) have shown clear separation between three classes (i.e. genes) of MHC class-I dataset hence it is difficult to visually observe better clustering structure from both. Feature saliencies estimated from the LTM-FS are shown in (c) on a scale of 0-to-1.



(a) standard LTM

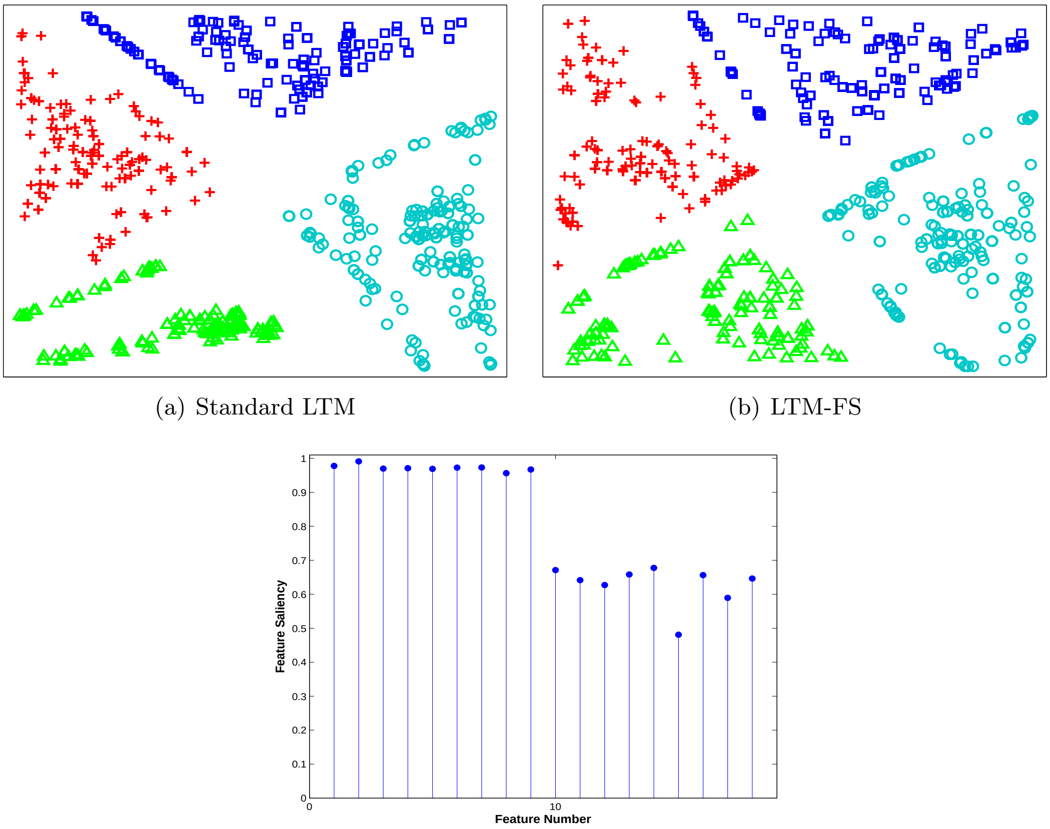
(b) LTM-FS

(c) Estimated saliencies

Figure C.15: THe LTM and the LTM-FS visualisations of the MHC class-I sequence-based binary dataset using a latent grid 4×4 and an RBF grid size 2×2. The data points shown as cyan circles represent alleles of HLA-A, red plus signs for HLA-B and blue squares for HLA-C. Both the LTM and the LTM-FS visualisations (i.e. (a) and (b) respectively) have shown clear separation between three classes (i.e. genes) of an MHC class-I dataset hence it is difficult to visually observe better clustering structure from both. Feature saliencies estimated from the LTM-FS are shown in (c) on a scale of 0-to-1.

C.4 The standard LTM and the LTM-FS models additional results with different noise level in the noisy features

We present here some additional results where we consider different noise densities for the irrelevant features.

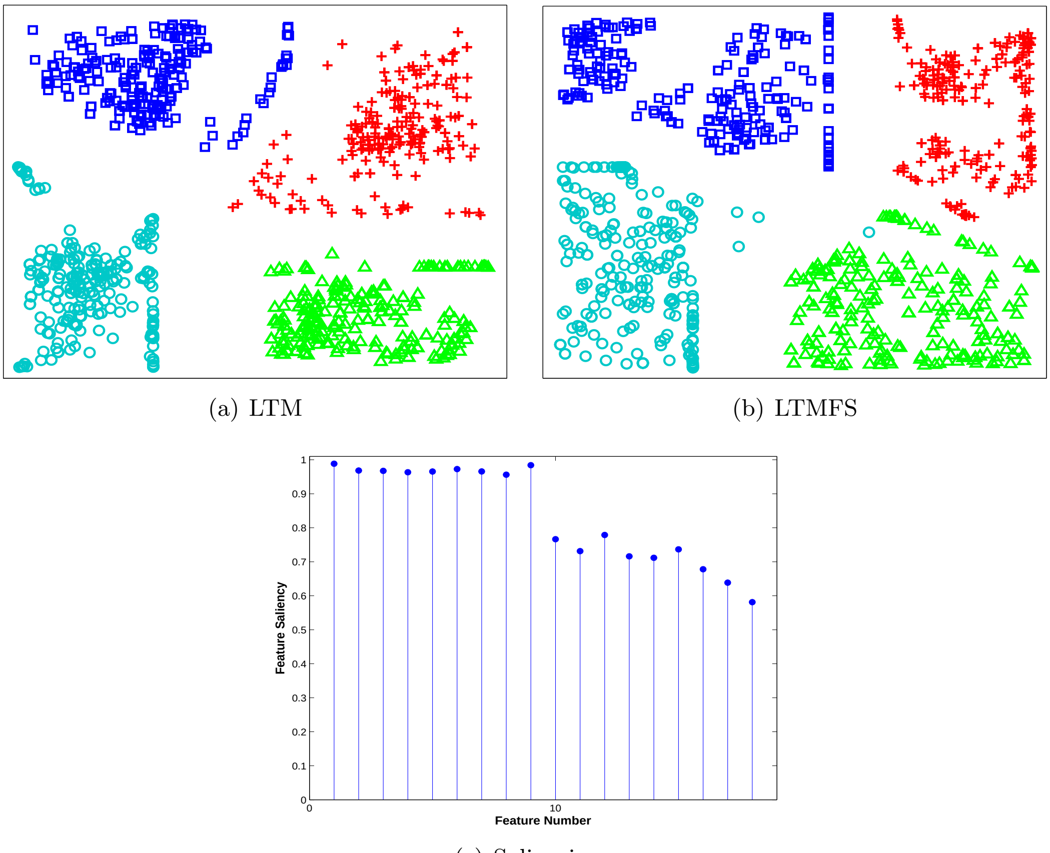


(a) Standard LTM

(b) LTM-FS

(c) Estimated saliencies

Figure C.16: The LTM and the LTM-FS visualisations of the synthetic dataset-I (with binary features). (a) and (b) show visualisations using the LTM and the LTM-FS mdoels whereas (c) show saliencies estimated from the LTM-FS. First 9 features were considered as relevant with clustering information whereas the remaining 9 have randomly distributed 1's with a density $p = 0.2$.



(a) LTM

(b) LTMFS

(c) Saliencies

Figure C.17: Demonstration binary type synthetic dataset-I using the standard LTM and the LTM-FS visualisation. (a) and (b) show visualisation using the LTM and the LTM-FS mdoels whereas (c) show saliencies estimated from the LTM-FS. First 9 features were considered as relevant with clustering information whereas the remaining 9 have randomly distributed 1's with a density $p = 0.6$.

D The GGTM-FS visualisation model, EM derivation of GGTM-FS and additional results for the GGTM and the GGTM-FS

D.1 The mixture model for the GGTM-FS

Recalling equation (6.27), which is the conditional density of $\mathbf{x}$ with the given Ψ (where $\Psi = \{\Psi^{\mathcal{R}}, \Psi^{\mathcal{B}}, \Psi^{\mathcal{C}}\}$ and in more general form $\Psi^{\mathcal{M}} = \{\psi_1^{\mathcal{M}}, \cdots, \psi_{|\mathcal{M}|}^{\mathcal{M}}\}$),

$$p(\mathbf{x}|\Psi) = \sum_{k=1}^{K} \pi_k \left[\prod_{\mathcal{M} \in \{\mathcal{R},\mathcal{B},\mathcal{C}\}} \left[\prod_{d=1}^{|\mathcal{M}|} \left[p(x_d^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}}) \right]^{\psi_d^{\mathcal{M}}} \left[q(x_d^{\mathcal{M}}|\lambda_d^{\mathcal{M}}) \right]^{(1-\psi_d^{\mathcal{M}})} \right] \right] \tag{D.1}$$

We take Ψ as a set of missing variables and we define feature saliency to be estimated as $\rho_d^{\mathcal{M}} = p(\psi_d^{\mathcal{M}} = 1)$, for $d = 1, \cdots, |\mathcal{M}|$. We consider that the ψ_ds are mutually independent and also independent of the hidden component label y for any data pattern $\mathbf{x}$. Hence,

$$\begin{aligned}
p(\mathbf{x},\Psi) &= p(\mathbf{x}|\Psi)p(\Psi)\\
&= \left[\sum_{k=1}^{K}\pi_k\left[\prod_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}}\left[\prod_{d=1}^{|\mathcal{M}|}\left[p(x_d^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})\right]^{\psi_d^{\mathcal{M}}}\left[q(x_d^{\mathcal{M}}|\lambda_d^{\mathcal{M}})\right]^{(1-\psi_d^{\mathcal{M}})}\right]\prod_{d=1}^{|\mathcal{M}|}(\rho_d^{\mathcal{M}})^{\psi_d^{\mathcal{M}}}(1-\rho_d^{\mathcal{M}})^{1-\psi_d^{\mathcal{M}}}\right]\right]\\
&= \sum_{k=1}^{K}\pi_k\left[\prod_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}}\left[\prod_{d=1}^{|\mathcal{M}|}\left[\rho_d^{\mathcal{M}}p(x_d^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})\right]^{\psi_d^{\mathcal{M}}}\left[(1-\rho_d^{\mathcal{M}})q(x_d^{\mathcal{M}}|\lambda_d^{\mathcal{M}})\right]^{(1-\psi_d^{\mathcal{M}})}\right]\right]
\end{aligned}\tag{D.2}$$

The marginal density of $\mathbf{x}$ is defined as

$$\begin{aligned}
p(\mathbf{x}) &= \sum_{\Psi}p(\mathbf{x},\Psi)\\
&= \sum_{k=1}^{K}\pi_k\sum_{\Psi}\left[\prod_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}}\left[\prod_{d=1}^{|\mathcal{M}|}\left[\rho_d^{\mathcal{M}}p(x_d^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})\right]^{\psi_d^{\mathcal{M}}}\left[(1-\rho_d^{\mathcal{M}})q(x_d^{\mathcal{M}}|\lambda_d^{\mathcal{M}})\right]^{(1-\psi_d^{\mathcal{M}})}\right]\right]\\
&= \sum_{k=1}^{K}\pi_k\left[\prod_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}}\left[\prod_{d=1}^{|\mathcal{M}|}\sum_{\Psi}\left[\rho_d^{\mathcal{M}}p(x_d^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})\right]^{\psi_d^{\mathcal{M}}}\left[(1-\rho_d^{\mathcal{M}})q(x_d^{\mathcal{M}}|\lambda_d^{\mathcal{M}})\right]^{(1-\psi_d^{\mathcal{M}})}\right]\right],\\
&= \sum_{k=1}^{K}\pi_k\left[\prod_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}}\left[\prod_{d=1}^{|\mathcal{M}|}\left[\rho_d^{\mathcal{M}}p(x_d^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})\right]+\left[(1-\rho_d^{\mathcal{M}})q(x_d^{\mathcal{M}}|\lambda_d^{\mathcal{M}})\right]\right]\right]
\end{aligned}\tag{D.3}$$

which is (6.28) and also note that the features are considered as independent of the component label y.

D.2 Deriving the EM algorithm for the GGTM-FS

From equation (6.28), the complete-data log-likelihood is defined as,

$$p(\mathbf{x}|\Omega) = \pi_k\left[\prod_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}}\left[\prod_{d=1}^{|\mathcal{M}|}[\rho_d^{\mathcal{M}}p(x_d^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})]+[(1-\rho_d^{\mathcal{M}})q(x_d^{\mathcal{M}}|\lambda_d^{\mathcal{M}})]\right]\right]\tag{D.4}$$

By defining the following terms

$$r_{nk} = p(y_n = k|\mathbf{x}_n)\tag{D.5}$$

$$u_{nkd}^{\mathcal{M}} = p(\psi_d^{\mathcal{M}} = 1, y_n == k|\mathbf{x}_n^{\mathcal{M}})\tag{D.6}$$

$$v_{nkd}^{\mathcal{M}} = p(\psi_d^{\mathcal{M}} = 0, y_n == k|\mathbf{x}_n^{\mathcal{M}}) \tag{D.7}$$

All these quantities are calculated using the current set of parameters $\Omega^{current}$. It is important to note that $u_{nkd}^{\mathcal{M}}+v_{nkd}^{\mathcal{M}} = r_{nk}$ and $\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} = N$. The expected complete data log-likelihood using the current set of parameters $\Omega^{current}$ we get

$$\begin{aligned}
&E_{\Omega^{current}}[\ln P(\mathbf{X},\mathbf{y},\Psi)]\\
&= \sum_{n,k,\Psi} p(y_n = k, \Psi|\mathbf{x}_n)\left(\ln \pi_k + \sum_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}}\sum_{d=1}^{|\mathcal{M}|}\left(\psi_d^{\mathcal{M}}\left(\ln p(x_{nd}^{\mathcal{M}}|\theta_{kd}^{\mathcal{M}}) + \ln \rho_d^{\mathcal{M}}\right)\right.\right.\\
&\quad \left.\left. +(1-\psi_d^{\mathcal{M}})(\ln q(x_{nd}^{\mathcal{M}}|\lambda_d^{\mathcal{M}}) + \ln(1-\rho_d^{\mathcal{M}}))\right)\right)\\
&= \sum_{n,k} p(y_n = k|\mathbf{x}_i)\ln \pi_k + \sum_{n,k}\sum_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}}\sum_{d=1}^{|\mathcal{M}|}\sum_{\psi_d^{\mathcal{M}}=0}^{1} p(y_n = k, \psi_d^{\mathcal{M}}|\mathbf{x}_n^{\mathcal{M}})\\
&\quad \left(\psi_d^{\mathcal{M}}\left(\ln p(x_{nd}^{\mathcal{M}}|\theta_{kd}^{\mathcal{M}}) + \ln \rho_d^{\mathcal{M}}\right) + (1-\psi_d^{\mathcal{M}})(\ln q(x_{nd}^{\mathcal{M}}|\lambda_d^{\mathcal{M}}) + \ln(1-\rho_d^{\mathcal{M}}))\right)\\
&= \underbrace{\sum_{k}(\sum_{n} r_{nk})\ln \pi_k +}_{\text{part 1}} \sum_{\mathcal{M}\in\{\mathcal{R},\mathcal{B},\mathcal{C}\}}\left(\sum_{k,d}^{K|\mathcal{M}|}\underbrace{\sum_{n} u_{nkd}^{\mathcal{M}}\ln p(x_{nd}^{\mathcal{M}}|\theta_{kd}^{\mathcal{M}})+}_{\text{part 2}}\sum_{d=1}^{|\mathcal{M}|}\underbrace{\sum_{n,k} v_{nkd}^{\mathcal{M}}\ln q(x_{nd}^{\mathcal{M}}|\lambda_d^{\mathcal{M}})}_{\text{part 3}}\right.\\
&\quad \left. +\sum_{d=1}^{|\mathcal{M}|}\left(\underbrace{\ln \rho_d^{\mathcal{M}}\sum_{n,k} u_{nkd}^{\mathcal{M}} + \ln(1-\rho_d^{\mathcal{M}})\sum_{n,k} v_{nkd}^{\mathcal{M}}}_{\text{part 4}}\right)\right)
\end{aligned} \tag{D.8}$$

Each part in the above equation can be maximized separately. Recalling that the densities $p^{\mathcal{M}}(.)$ and $q^{\mathcal{M}}(.)$ are univariate Gaussians if $\mathcal{M} = \mathcal{R}$ and are characterized by their means and variances and, if $\mathcal{M} = \mathcal{B}$ then these are univariate Bernoulli which are defined by the means and, if $\mathcal{M} = \mathcal{R}$ then these are univariate Multinomial which are defined by the means. Now maximizing the expected log-likelihood of the complete-data gives the M-step equations (6.35)-(6.41) (i.e. for Gaussian case), (6.42)-(6.46) (i.e. for Bernoulli case) and (6.47)-(6.51) (i.e. for multinomial case). It is also important to note that

$$\begin{aligned} p(\psi_d^{\mathcal{M}}=1|y_n=k,\mathbf{x}_n^{\mathcal{M}}) &= \frac{p(\psi_d^{\mathcal{M}}=1,\mathbf{x}_n^{\mathcal{M}}|y_n=k)}{p(\mathbf{x}_n^{\mathcal{M}}|y_n=k)} \\ &= \frac{\rho_d^{\mathcal{M}}p(x_d^{\mathcal{M}}|\theta_{kd}^{\mathcal{M}})\prod_{d'\neq d}^{|\mathcal{M}|}\left(\rho_{d'}^{\mathcal{M}}p(x_{d'}^{\mathcal{M}}|\theta_{kd'}^{\mathcal{M}})+(1-\rho_{d'}^{\mathcal{M}})q(x_{d'}^{\mathcal{M}}|\lambda_{d'}^{\mathcal{M}})\right)}{\prod_{d'=1}^{|\mathcal{M}|}\left(\rho_{d'}^{\mathcal{M}}p(x_{d'}^{\mathcal{M}}|\theta_{kd'}^{\mathcal{M}})+(1-\rho_{d'}^{\mathcal{M}})q(x_{d'}^{\mathcal{M}}|\lambda_{d'}^{\mathcal{M}})\right)} \\ &= \frac{\rho_d^{\mathcal{M}}p(x_d^{\mathcal{M}}|\theta_{kd}^{\mathcal{M}})}{\rho_d^{\mathcal{M}}p(x_d^{\mathcal{M}}|\theta_{kd}^{\mathcal{M}})+(1-\rho_d^{\mathcal{M}})q(x_d^{\mathcal{M}}|\lambda_d^{\mathcal{M}})} \end{aligned} \tag{D.9}$$

Therefore equation (6.33) becomes

$$\begin{aligned} u_{nkd}^{\mathcal{M}} &= p(\psi_d^{\mathcal{M}}=1|y_n=k,\mathbf{x}_n^{\mathcal{M}})p(y_n=k|\mathbf{x}_n) \\ &= \frac{\rho_d^{\mathcal{M}}p(x_d^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})}{\rho_d^{\mathcal{M}}p(x_d^{\mathcal{M}}|\Theta_{kd}^{\mathcal{M}})]+[(1-\rho_d^{\mathcal{M}})q(x_d|\lambda_d^{\mathcal{M}})}r_{nk}. \end{aligned} \tag{D.10}$$

D.3 The priors over the features saliency parameter update

Maximizing the expected log-likelihood of the complete-data (equation (D.8)), the M-step for the updating the feature saliency is defined as

$$\widehat{\rho_d^{\mathcal{M}}} = \frac{\sum_{n,k}u_{nkd}^{\mathcal{M}}}{\sum_{n,k}u_{nkd}^{\mathcal{M}}+\sum_{n,k}v_{nmd}^{\mathcal{M}}} = \frac{\sum_{n,k}u_{nkd}^{\mathcal{M}}}{N}. \tag{D.11}$$

We take Dirichlet-type prior (which is improper) for saliencies of continuous features (see Equation (D.12) and similar prior was previously used in (Law et al., 2004)), Beta distribution prior for saliencies of binary features (see Equation (D.13) and similar prior was previously used in (Bouguila, 2010)) and a Dirichlet-type prior (which is a natural conjugate prior of the multinomial) for the saliencies of the multi-category features (see Equation (D.14) and similar prior was previously used in (Silvestre et al., 2013)).

$$p(\rho_1^{\mathcal{R}},\cdots,\rho_{|\mathcal{R}|}^{\mathcal{R}}) \propto \prod_{d=1}^{|\mathcal{R}|}(\rho_d^{\mathcal{R}})^{-\frac{KP}{2}}(1-\rho_d^{\mathcal{R}})^{-\frac{T}{2}} \tag{D.12}$$

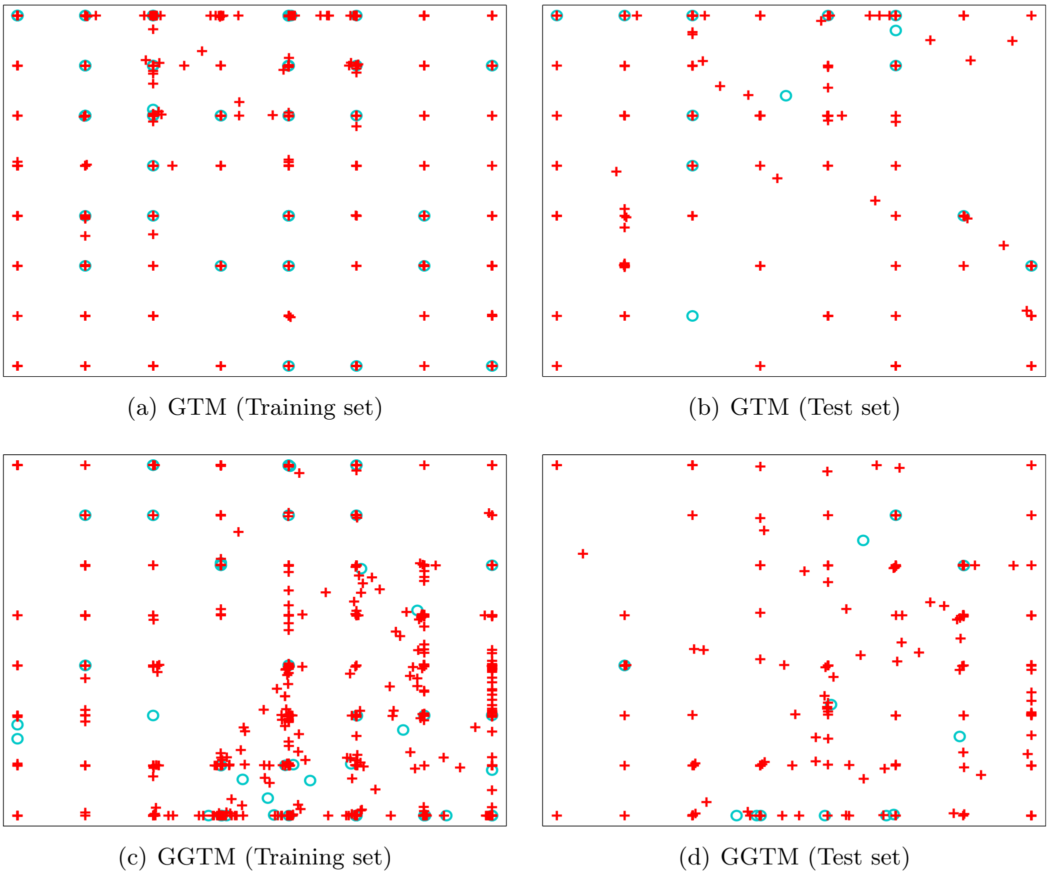
$$p(\rho_1^{\mathcal{B}},\cdots,\rho_{|\mathcal{B}|}^{\mathcal{B}}) = \prod_{d=1}^{|\mathcal{B}|}\frac{\Gamma(\alpha_d+\beta_d)}{\Gamma(\alpha_d)\Gamma(\beta_d)}(\rho_d^{\mathcal{B}})^{\alpha_d-1}(1-\rho_d^{\mathcal{B}})^{\beta_d-1}, \tag{D.13}$$

$$p(\rho_1^{\mathcal{C}},\cdots,\rho_{|\mathcal{C}|}^{\mathcal{C}}) \propto \prod_{d=1}^{|\mathcal{C}|}(\rho_d^{\mathcal{C}})^{-\frac{Kc_d}{2}}(1-\rho_d^{\mathcal{C}})^{-\frac{c_d}{2}} \tag{D.14}$$

Therefore applying the respective priors gives the M-step in equations (6.41) (i.e. for continuous features), (6.46) (i.e. for binary features) and (6.51) (i.e. for binary features).

D.4 Additional Results

D.4.1 Bioassay dataset 'AID1608'


(a) GTM (Training set)

(b) GTM (Test set)

(c) GGTM (Training set)

(d) GGTM (Test set)

Figure D.1: The standard GTM and the GGTM visualisations of a Bioassays dataset 'AID1608' (continuous and binary features). Legend same as in Figure 6.4

		Trustworthiness		**Continuity**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	0.7721	**0.7368**	**0.8333**	0.7984
	GGTM	**0.7878**	0.7291	0.8286	**0.7807**
2	**GTM**	0.7738	**0.7512**	**0.8155**	**0.7936**
	GGTM	**0.7904**	0.7274	0.7924	0.7428
3	**GTM**	0.7492	0.7245	0.7854	0.7348
	GGTM	**0.7593**	**0.7315**	**0.7966**	**0.7421**
4	**GTM**	0.7775	**0.7452**	**0.8185**	**0.7870**
	GGTM	**0.7871**	0.7410	0.7924	0.7468
5	**GTM**	0.7410	0.7198	0.7880	**0.7587**
	GGTM	**0.7621**	**0.7414**	**0.8025**	0.7520
6	**GTM**	0.7306	0.7287	0.7783	**0.7594**
	GGTM	**0.7407**	**0.7296**	**0.7841**	0.7393

Table D.1: The GTM and the GGTM comparison using quality evaluation metrics of the trustworthiness and the continuity for the bioassay dataset 'AID1608'.

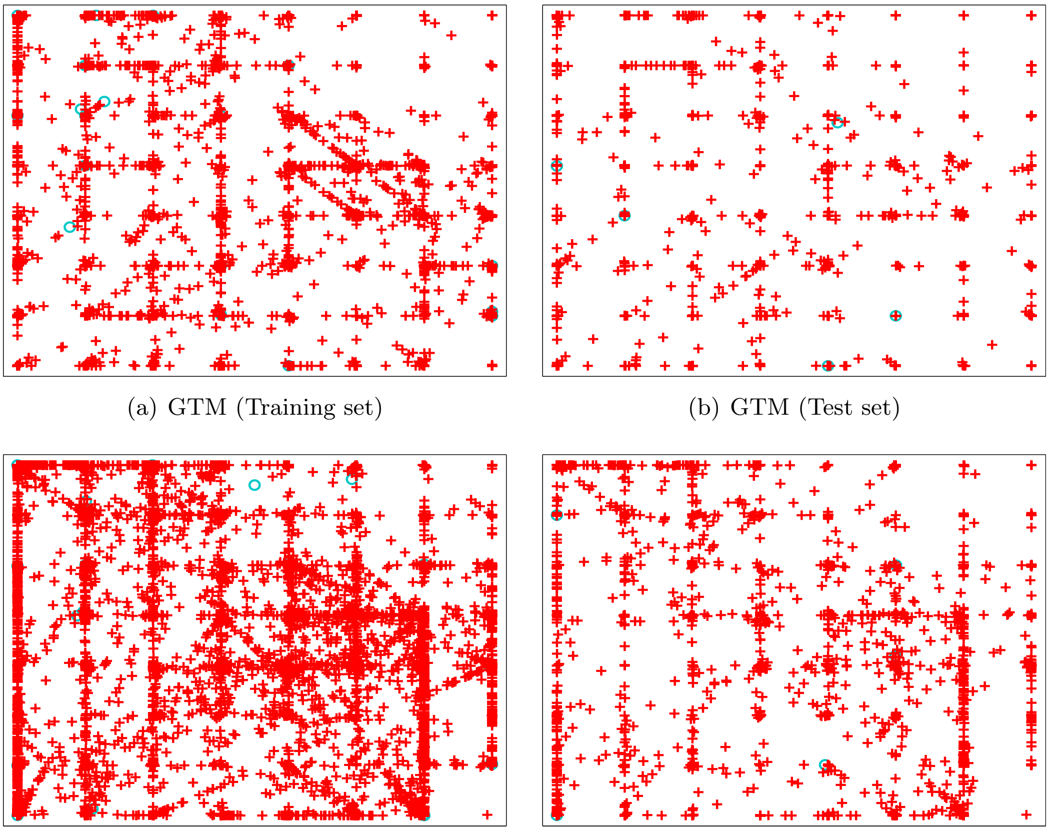
		MRREd		**MRREl**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	**0.0805**	1.9920	**0.0781**	**1.7098**
	GGTM	0.0823	**1.9261**	0.0788	1.7419
2	**GTM**	0.0872	1.8687	**0.0788**	**1.7408**
	GGTM	**0.0844**	**1.7687**	0.0797	1.8668
3	**GTM**	**0.0927**	1.8352	0.0801	1.8785
	GGTM	0.0936	**1.7887**	**0.0789**	**1.8559**
4	**GTM**	0.0905	1.8703	0.0800	**1.8292**
	GGTM	**0.0884**	**1.6508**	**0.0797**	1.8350
5	**GTM**	0.0962	1.9776	**0.0792**	1.8830
	GGTM	**0.0950**	**1.7765**	0.0800	**1.8197**
6	**GTM**	0.1016	2.0492	**0.0769**	1.9262
	GGTM	**0.0951**	**1.9055**	0.0806	1.9575

Table D.2: The GTM and the GGTM comparison using quality evaluation metrics of the mean relative rank errors with respect to the data space and the latent space for the bioassay dataset 'AID1608'.

		AVDD		**NLL**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	**0.7242**	0.6498	**4.4045**	**30.2783**
	GGTM	0.7428	**0.6470**	29.0927	34.0156
2	**GTM**	0.7873	0.7014	**25.8622**	37.6762
	GGTM	**0.7595**	**0.6633**	36.1514	**38.3979**
3	**GTM**	0.8030	0.7469	47.4196	55.5927
	GGTM	**0.7744**	**0.7312**	**42.3488**	**43.0953**
4	**GTM**	0.7901	0.6953	**30.7389**	41.9343
	GGTM	**0.7598**	**0.6687**	37.7940	**39.6991**
5	**GTM**	0.7764	0.7271	51.0882	56.5348
	GGTM	**0.7415**	**0.7119**	**43.5000**	**44.2253**
6	**GTM**	0.7996	0.7358	57.4340	63.9826
	GGTM	**0.7630**	**0.7190**	**45.9382**	**46.6930**

Table D.3: The GTM and the GGTM comparison using quality evaluation metrics of the distance distortion and the negative log-likelihood (per point) for the bioassay dataset 'AID1608'.

D.4.2 Bioassay dataset 'AID456'



(a) GTM (Training set)

(b) GTM (Test set)

(c) GGTM (Training set)

(d) GGTM (Test set)

Figure D.2: The GTM and the GGTM visualisations of bioassays dataset 'AID456' (continuous and binary features). Legend same as in Figure 6.4

		Trustworthiness		**Continuity**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	0.8327	0.7815	0.8056	0.7740
	GGTM	**0.8448**	**0.7978**	**0.8385**	**0.8038**
2	**GTM**	**0.8169**	**0.7874**	0.7917	0.7573
	GGTM	0.8088	0.7862	**0.8062**	**0.7694**
3	**GTM**	**0.7694**	**0.7549**	0.7996	0.7658
	GGTM	0.7595	0.7509	**0.8314**	**0.7880**
4	**GTM**	0.8066	**0.7866**	**0.8190**	**0.7874**
	GGTM	**0.8077**	0.7865	0.8164	0.7771
5	**GTM**	**0.7617**	0.7518	0.8037	0.7767
	GGTM	0.7608	**0.7546**	**0.8334**	**0.7996**
6	**GTM**	0.7142	0.7299	0.8127	0.7819
	GGTM	**0.7604**	**0.7530**	**0.8480**	**0.8146**

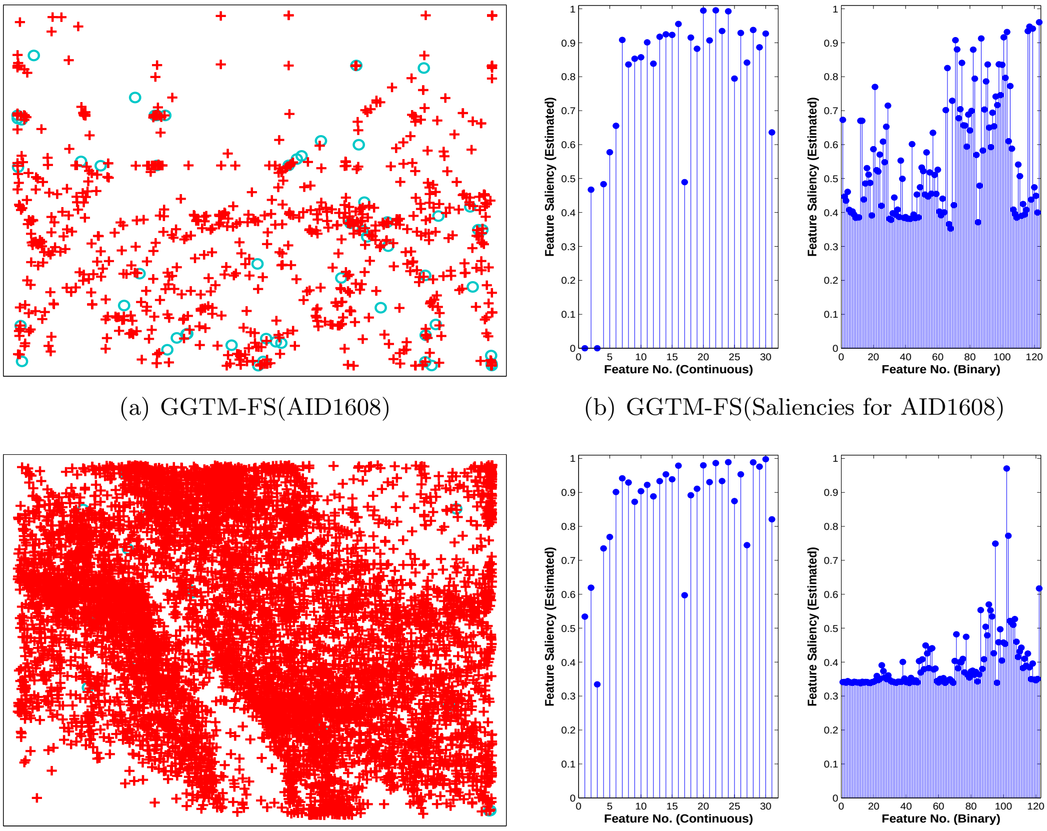
Table D.4: The GTM and the GGTM comparison using quality evaluation metrics of the trustworthiness and the continuity for the bioassay dataset 'AID456'.

		MRREd		**MRREl**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	**0.0077**	**0.0314**	**0.0070**	**0.0314**
	GGTM	0.0079	0.0320	0.0071	0.0315
2	**GTM**	0.0077	**0.0329**	0.0069	0.0311
	GGTM	0.0077	0.0336	0.0069	**0.0310**
3	**GTM**	**0.0073**	**0.0325**	0.0068	0.0299
	GGTM	0.0075	0.0335	**0.0067**	**0.0294**
4	**GTM**	**0.0076**	**0.0337**	**0.0068**	0.0310
	GGTM	0.0077	0.0338	0.0069	**0.0309**
5	**GTM**	**0.0072**	**0.0326**	0.0068	0.0298
	GGTM	0.0073	0.0332	**0.0067**	**0.0295**
6	**GTM**	**0.0072**	**0.0319**	**0.0066**	**0.0287**
	GGTM	0.0074	0.0334	0.0067	0.0294

Table D.5: The GTM and the GGTM comparison using quality evaluation metrics of the mean relative rank errors with respect to the data space and the latent space for the bioassay dataset 'AID456'.

		AVDD		**NLL**	
		Training	**Test**	**Training**	**Test**
1	**GTM**	**0.8407**	**0.7998**	**28.3908**	**31.6213**
	GGTM	0.8535	0.8173	36.5018	36.9206
2	**GTM**	**0.8621**	**0.8225**	**39.0959**	**40.1398**
	GGTM	0.8762	0.8289	41.7087	41.8175
3	**GTM**	**0.8757**	**0.8508**	55.3785	55.5542
	GGTM	0.8874	0.8946	**51.2770**	**51.2801**
4	**GTM**	**0.8585**	**0.8183**	43.5062	44.1405
	GGTM	0.8771	0.8462	**43.3726**	**43.4341**
5	**GTM**	**0.8599**	**0.8392**	57.6106	57.7671
	GGTM	0.9133	0.8933	**50.0265**	**49.9552**
6	**GTM**	**0.8699**	**0.8321**	63.7037	64.6040
	GGTM	0.8838	0.8557	**48.9251**	**48.8951**

Table D.6: The GTM and the GGTM comparison using quality evaluation metrics of the distance distortion and the negative log-likelihood per point for the bioassay dataset ‘AID456’.



(a) GGTM-FS(AID1608)

(b) GGTM-FS(Saliencies for AID1608)

(c) GGTM-FS(AID456)

(d) GGTM-FS(Saliencies for AID456)

Figure D.3: The GGTM-FS visualisation and estimated features saliencies for bioassay datasets (continuous and binary). Legend same as in Figure 6.4

Bibliography

S. Alelyani, J. Tang, and H. Liu. Feature selection for clustering: A review. In *Data Clustering: Algorithms and Applications*, pages 29–60. Chapman and Hall/CRC, 2013.

S. Aluru. *Handbook of computational molecular biology.* CRC Press, 2005.

M. Ankerst. Visual data mining with pixel-oriented visualization techniques. In *Proceedings of the ACM SIGKDD Workshop on Visual Data Mining*, 2001.

C. Azuara, E. Lindahl, P. Koehl, H. Orland, and M. Delarue. PDB_Hydro: Incorporating dipolar solvents with variable density in the Poisson-Boltzmann treatment of macromolecule electrostatics. *Nucleic Acids Research*, 34:38–42, 2006.

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

N. A. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon. Electrostatics of nanosystems: application to microtubules and the ribosome. *Proceedings of the National Academy of Sciences*, 98(18):10037–10041, 2001.

O. Barndorff-Nielsen. *Information and exponential families in statistical theory.* John Wiley & Sons Ltd, 1978.

F. H. Bellamine and A. Elkamel. Model order reduction using neural network principal component analysis and generalized dimensional analysis. *Engineering Computations*, 25(5):443–463, 2008.

R. E. Bellman and R. Corporation. *Dynamic programming.* Rand Corporation Research Study. Princeton University Press, 1957.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Res*, 28(1): 235–242, 2000.

F. C. Bernsten, T. F. Koetzle, G. F. Williams, E. E. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The protein data bank: A computer-based archival file for macromolecular structures. *Journal of Molecular Biology*, 112 (535), 1977. http://www.rcsb.org/pdb/.

C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, New York, 1st edition, 1995a.

C. M. Bishop. *Neural networks for pattern recognition.* Oxford university press, 1995b.

C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

C. M. Bishop and G. D. James. Analysis of Multiphase Flows Using Dual-Energy Gamma Densitometry and Neural Networks. *Nuclear Instruments and Methods in Physics Research*, 327(2-3):580–593, 1993.

C. M. Bishop and M. Svensen. GTM: The generative topographic mapping. *Neural Compuatation*, 10(1):215–234, 1998.

C. M. Bishop and M. E. Tipping. A hierarchical latent variable model for data visualization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):281–293, March 1998.

C. M. Bishop, M. Svensen, and C. K. I. Williams. Magnification factor for the GTM algorithm. In *Proceedings IEE International Conference on Artificial Neural Networks*, pages 64–69, 1997.

C. M. Bishop, M. Svensen, and C. K. I. Williams. Developments of the generative topographic mapping. *Neurocomputing*, 21(1):203–224, 1998.

T. K. Bjarnadóttir, D. E. Gloriam, S. H. Hellstrand, H. Kristiansson, R. Fredriksson, and H. B. Schiöth. Comprehensive repertoire and phylogenetic analysis of the G protein-coupled receptors in human and mouse. *Genomics*, 88(3):263–273, 2006.

N. Blomberg, R. R. Gabdoulline, N. Michael, and R. C. Wade. Classification of protein sequences by homology modeling and quantiative analysis of electrostatic similarity. *Proteins: Structure, Function and Genetics*, 37:379–387, 1999.

N. Bouguila. On multivariate binary data clustering and feature weighting. *Comput. Stat. Data Anal.*, 54(1):120–134, 2010.

M. J. Bower, F. E. Cohen, and Jr. Dunbrack. Sidechain prediction from a backbone-dependent rotamer library: A new tool for homology modeling. *J. Mol. Biol*, 267: 1268–1282, 1997.

F. Cainelli and S. Vento. Infections and solid organ transplant rejection: a cause-and-effect relationship? *The Lancet infectious diseases*, 2(9):539–549, 2002.

N. A. Campbell and J. B. Reeca. *Biology*. Pearson, 8th edition, 2008.

I. O. Caparroso. *Variational Bayesian algorithms for generative topographic mapping and its extensions*. PhD thesis, Universitat Politècnica de Catalunya, 2008.

C. Chothia and A. M. Lesk. The relation between the divergence of sequence and structure in proteins. *The EMBO Journal*, 5(4):823–826, 1986.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Springer, 1st edition, 1991.

T. Cox and M. Cox. Multidimensional scaling. *Chapman&Hall, London, UK*, 1994.

A. R. de Leon and K. C. Chough. *Analysis of Mixed Data: Methods & Applications*. Taylor & Francis Group. Chapman and Hall/CRC, 2013.

P. Demartines and J. Hérault. CCA: Curvilinear component analysis. In *15 Colloque sur le traitement du signal et des images, FRA, 1995*. GRETSI, Groupe d' Etudes du Traitement du Signal et des Images, 1995.

P. Demartines and J. Hérault. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8 (1):148–154, 1997.

A. P. Dempster, N. M. Laird, D. B. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977.

T. J. Dolinsky, J. E. Nielsen, J. A. McCammon, and N. A. Baker. PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations. *Nucleic Acids Research*, 32(Web-Server-Issue):665–667, 2004.

T. J. Dolinsky, P. Czodrowski, H. Li, J. E. Nielsen, J. H. Jensen, G. Klebe, and N. A. Baker. PDB2PQR: Expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic Acids Res*, 35(W522-5), 2007.

F. Dong, B. Oslen, and N. A. Baker. Computation methods for biomolecular electrostatics. *Methods in Cell Biology*, 84:843–870, 2008.

D. L. Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 2000.

A. Dourado, E. Ferreira, and P. Barbeiro. VisRed: numerical data mining with linear and nonlinear techniques. In *ICDM'07: Proceedings of the 7th industrial conference on Advances in data mining*, pages 92–106, Berlin, Heidelberg, 2007. Springer-Verlag.

I. A. Doytchinova and D. R. Flower. In silico identification of supertypes for class II MHCs. *The Journal of Immunology*, 174:7085–7095, 2005.

I. A. Doytchinova, P. Guan, and D. R. Flower. Identifying human MHC supertypes uisng bioinformatics methods. *The Journal of Immunology*, 172:4314–4323, 2004.

R. L. J. Dunbrack. Rotamer libraries in the 21st century. *Current Opinion in Structural biology*, 12(4):431–440, 2002.

D. B. Dunson. Bayesian latent variable models for clustered mixed outcomes. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 62(2):pp. 355–366, 2000.

J. A. Endicott and M. E. Noble. Structural characterization of the cyclin-dependent protein kinase family. *Biochemical Society transactions*, 41(4):1008–1016, 2013.

B. J. J. Frey and D. Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

A. Ghaffar and P. Nagarkatti. MHC: genetics and role in transplantation. http://pathmicro.med.sc.edu/ghaffar/mhc2000.htm. Last accessed on 04-07-2014.

J. Greenbaum, J. Sidney, J. Chung, C. Brander, B. Peters, and A. Sette. Functional classification of class II human leukocyte antigen (HLA) molecules reveals seven different supertypes and a surprising degree of repertoire sharing across supertypes. *Immunogenetics*, 63:325–335, 2011.

W. Gronwald and H. R. Kalbitzer. Automated structure determination of proteins by NMR spectroscopy. *Biological Cybernetics*, 44:33–96, 2004.

K. Gupta, D. Thomas, S. V. Vidya, and S. Ramakimar. Detailed protein sequence alignment based on spectral similarity score(sss). *BMC Bioinformatics*, 6(105):1–16, 2005.

S. Harjanto, L .F .P. Ng, and J. C. Tong. Clustering HLA class I superfamilies using structural interaction patterns. *PloS One*, 9(1):e86655, 2014.

J. Havlicek and S. C. Roberts. MHC-correlated mate choice in humans: a review. *Psychoneuroendocrinology*, 34(4):497–512, 2009.

K. M. Heinonen and C. Perreault. Development and functional properties of thymic and extrathymic t lymphocytes. *Critical Reviews in Immunology*, 28(5), 2008.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

F. Imola. A survey of dimension reduction techniques. Technical report, Lawrence Livermore National Laboratory, 2002.

I. T. Jolliffe. *Principal Component Analysis.* 2nd Edn, Springer Series in Statistics, 2002.

A. Kabán and M. Girolami. A combined latent class and trait model for the analysis and visualization of discrete data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(8):859–872, 2001.

S. Kaski, J. Nikkilä, M. Oja, J. Venna, P. Törönen, and E. Castrén. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*, 4(1):48, 2003.

Daniel A Keim. Information visualization and visual data mining. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):1–8, 2002.

P. Kelchtermans, W. Bittremieux, K. Grave, S. Degroeve, J. Ramon, K. Laukens, D. Valkenborg, H. Barsnes, and L. Martens. Machine learning applications in proteomics research: How the past can boost the future. *Proteomics*, 14(4-5):353–366, 2014.

P. Kiepiela, A. J. Leslie, I. Honeyborne, D. Ramduth, C. Thobakgale, S. Chetty, P. Rathnavalu, C. Moore, K. J. Pfafferott, L. Hilton, et al. Dominant influence of HLA-B in mediating the potential co-evolution of HIV and HLA. *Nature*, 432(7018):769–775, 2004.

T. J. Kindt, R. A. Goldsby, and B. A. Osborne. *Kuby Immunology.* Macmillan, 2007.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2): 273–324, 1997.

T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

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

M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

E. Krieger, E. B. Nabuurs, G. Vriend, E. Philip, E. Bourne, and H. Weissig. Homology modeling. *Methods of Biochemical Analysis*, 44, 2003.

G. G. Krivov, M. V. Shapovalov, and R. L. Dunbrack. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins: Structure, Function, and Bioinformatics*, 77(4):778–795, 2009.

J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2):115–129, 1964.

W. J. Krzanowski. Distance between populations using mixed continuous and categorical variables. *Biometrika*, 70(1):pp. 235–243, 1983.

D. Kuonen. Challenges in bioinformatics for statistical data miners. *Bulletin of the Swiss Statistical Society*, 46:10–17, 2003.

U. Langel, B. F. Cravatt, A. Graslund, G. V. Heijne, T. Land, S. Niessen, and M. Zorko. *Introduction to Proteins and Peptides*. CRC Press Taylor and Francis Group, 2010.

R. A. Laskowski, J. D. Watson, and J. M. Thornton. Protein function prediction using local 3d templates. *Journal of Molecular Biology*, 351(3):614–626, 2005.

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

N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 329–336, Cambridge, MA, 2004. MIT Press.

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

N. D. Lawrence. Local distance preservation in the GPLVM through back constraints. In *In ICML*, pages 513–520. ACM Press, 2006.

N. D. Lawrence. Large scale learning with the Gaussian process latent variable model. Technical report, University of Sheffield, United Kingdom, 2008. ftp://ftp.dcs.shef.ac.uk/home/neil/gplvmSparse.pdf. Last accessed on 04-07-2014.

N. D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems 15*, pages 609–616. MIT Press, 2003.

J. A. Lee and M. Verleysen. Rank-based quality assessment of nonlinear dimensionality reduction. In *ESANN*, pages 49–54, 2008.

J.A. Lee and M. Verleysen. *Nonlinear dimensionality reduction.* Springer, 2007.

K. Lees. *Data projections for the analysis and visualisation of bioinformatics data.* PhD thesis, University of Oxford, 2008.

D. Lowe and M. E. Tipping. Neuroscale: Novel topographic feature extraction using RBF networks. In *NIPS*, pages 543–549, 1996.

N. M. Luscombe, D. GreenBaum, and D. Gerstein. What is bioinformatics? An introduction and overview. *Methods of Information in Medicine*, 40(4):346–358, 2001.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability - Vol. 1*, pages 281–297. University of California Press, Berkeley, CA, USA, 1967.

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

D. M. Maniyar and I. T. Nabney. Visual data mining using principled projection algorithms and information visualzation techniques. In *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 643–647, 2006b.

D. M. Maniyar, I. T. Nabney, B. S. Williams, and A. Sewing. Data visualization during the early stage of the drug discovery. *Journal of Chemical and Information Modelling*, 46(4):1806–1818, 2006.

G. McLachlan and T. Krishnan. *The EM algorithm and extensions.* Wiley, New York, 1997.

B. Meyer. Self-organizing graphs - a neural network perspective of graph layout. In Sue Whitesides, editor, *Graph Drawing*, volume 1547 of *Lecture Notes in Computer Science*, pages 246–262. Springer, 1998.

R. A. Meyers. *Protein: Electron Microscopy of Biomolecules.* Wiley VCH, 2007.

I. Moustaki. A latent trait and a latent class model for mixed observed variables. *British Journal of Mathematical and Statistical Psychology*, 49(2):313–334, 1996.

S. Mumtaz, I. T. Nabney, and D. R. Flower. Novel visualization methods for protein data. In *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2012 IEEE Symposium on*, pages 198 –205, May 2012.

I. T. Nabney. NETLAB*: Algorithms for pattern Recognition.* UK, Springer, 2002.

S. K. Pal and P. Mitra. *Pattern Recognition Algorithms for Data Mining.* Chapmann and Hall/CRC, Boca Raton, London, Washington DC, 2004.

P. Parham. *The Immune System.* Elsevier, 2000.

S. Paul, D. Weiskopf, M. A. Angelo, J. Sidney, B. Peters, and A. Sette. HLA class I alleles are associated with peptide-binding repertoires of different size, affinity, and immunogenicity. *The Journal of Immunology*, 191(12):5831–5839, 2013.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

R. V. Polozov, V. S. Sivozhelezov, V. V. Ivanov, and Y. B. Melnikov. On a classification of E. coli promoters according to their electrostatic potential. *Particles and Nuclei Letters*, 2(4(127)):82–90, 2005.

N. C. Price and J. Nairn. *Exploring Proteins.* Oxford University Press, 2009.

J. Quionero-Candela, C. E. Rasmussen, and R. Herbrich. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

S. Ritcher, A. Wenzel, M. Stein, R. R. Gabdoulline, and R. C. Wade. WebPIPSA: A web server for the comparison of protein interaction propoerties. *Nucleic Acid Research*, 36: 276–280, 2008.

J. Robinson, M. J. Waller, N. Parham, D. Groot, H. R. Kalbitzer, L. J. Bontrop, P. Kennedy, P. Stoehr, and S. G. E. Marsh. IMGT/HLA and IMGT/MHC: Sequence databases for the study of the major histocompatibility complex. *Nucleic Acids Res*, 31 (311), 2003.

J. Robinson, J. A. Halliwell, H. McWilliam, R. Lopez, P. Parham, and S. GE. Marsh. the IMGT/HLA database. *Nucleic acids research*, 41(D1):D1222–D1227, 2013.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

A. Sali. *MODELLER A Program for Protein Structure Modeling Release 9v8 r7145*, 2010. http://www.salilab.org/modeller/. Last accessed on 04-07-2014.

M. D. Sammel, L. M. Ryan, and J. M. Legler. Latent variable models for mixed discrete and continuous outcomes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 59(3):pp. 667–678, 1997.

J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18:401–409, 1969.

P. S. C. Santos, J. A. Schinemann, J. Gabardo, and M. da Graça Bicalho. New evidence that the MHC influences odor perception in humans: a study with 58 southern Brazilian students. *Hormones and Behavior*, 47(4):384–388, 2005.

T. Schreck, T. von Landesberger, and S. Bremm. Techniques for precision-based visual analysis of projected data. *Information Visualization*, 9(3):181–193, 2010.

Martin Schroeder. *Exploratory data analysis with non-linear and missing data in geochemistry*. PhD thesis, Aston University, 2009.

A. Sette and J. Sidney. Nine major HLA class I supertypes account for the vast preponderance of HLA-A and -B poly-morphism. *Immunogenetics*, 50:201–212, 1999.

B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualization. In *IEEE Symposium on Visual Languages*, pages 336–343, 1996.

C. M. V. Silvestre, M. M. G. Cardoso, and M A. T. Figueiredo. Clustering and selecting categorical features. In Lus Correia, Lus Paulo Reis, and Jos Cascalho, editors, *EPIA*, volume 8154 of *Lecture Notes in Computer Science*, pages 331–342. Springer, 2013.

M. S. Smyth and J. H. J Martin. X-ray crytallography. *Clin Pathol:Mol Pathol*, 53:8–14, 2000.

R. C. Stevens. The cost and value of three-dimensional protein structure. *Drug Discovery World*, 4(3):35–48, 2003.

J. Su, Q. Xie, Y. Xu, XC. Li, Z. Dai, et al. Role of CD8+ regulatory T cells in organ transplantation. *Burns and Trauma*, 2(1):18, 2014.

Y. Sun, P. Tino, and I. T. Nabney. GTM-based data visualization with incomplete data. Technical report, Aston Univerisity Birmingham UK, 2001.

Johan FM Svénsen. *GTM: The generative topographic mapping.* PhD thesis, Aston University, 1998.

J. M. Thornton, A. E. Todd, D. Milburn, N. Borkakoti, and C. A. Orengo. From structure to function, applications and limitations. *Nature Structure to function: Approaches and Limitations*, 7:991–994, 2000.

P. Tino and I. T. Nabney. Hierarchical GTM: Constructing localized nonlinear projection manifolds in a principled way. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24, pages 639–656, 2002.

P. Tino, I. T. Nabney, and Y. Sun. Using directional curvatures to visualize folding patterns of the GTM projection manifolds. In *Artificial Neural Networks, ICANN 2001*, pages 421–428. Springer, 2001.

M. E. Tipping. Probabilistic visualisation of high-dimensional binary data. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, pages 592–598, Cambridge, MA, USA, 1999. MIT Press.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(1-41):66–71, 2009.

M. Vanderstraete, N. Gouignard, A. Ahier, M. Morel, J. Vicogne, and C. Dissous. The venus kinase receptor (VKR) family: structure and evolution. *BMC genomics*, 14(1): 361, 2013.

D. L. Veenstra and P. A. Kollman. Modelling protein stability: a theoretical analysis of the stability of T4 lysozyme mutants. *Protein Engineering*, 10(7):789–807, 1997.

A. Vellido. Preliminary theoretical results on a feature relevance determination method for generative topographic mapping. Technical report, Universitat Politecnica de Catalunya (UPC)LSI-05-13-R, Barcelona, Spain, 2005.

A. Vellido. Assessment of an unsupervised feature selection method for generative topographic mapping. In *Proceedings of the 16th International Conference on Artificial Neural Networks - Volume Part II*, ICANN'06, pages 361–370, Berlin, Heidelberg, 2006. Springer-Verlag.

A. Vellido, P. J. G. Lisboa, and D. Vicente. Robust analysis of MRS brain tumour data using *t*-GTM. *Neurocomputing*, 69(7-9):754–768, 2006.

J. Venna and S. Kaski. Neighborhood preservation in nonlinear projection methods: An experimental study. In *Proceedings of the International Conference on Artificial Neural Networks*, ICANN '01, pages 485–491, London, UK, 2001. Springer-Verlag.

J. Venna and S. Kaski. Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity, 2005. URL http://eprints.pascal-network.org/archive/00001233/. Last accessed on 04-07-2014.

L. Vinh, S. Lee, Y-T Park, and B. J. d'Auriol. A novel feature selection method based on normalized mutual information. *Applied Intelligence*, 37(1):100–120, 2012.

J. Wang, P. Cieplak, and P. A. Kollman. How well does a restrained electrostatic potential for molecular modelling and dynamics studies of proteins. *Journal of Computational Chemistry*, 21(12):1049–1074, 2000.

X. Wang and A. Kabán. Finding uninformative features in binary data. In Marcus Gallagher, JamesP. Hogan, and Frederic Maire, editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2005*, volume 3578 of *Lecture Notes in Computer Science*, pages 40–47. Springer Berlin Heidelberg, 2005.

J. Warwicker and H. C. Watson. How well does a restrained electrostatic potential (resp) model perform in calculating conformational energies of organic and biological molecules? *Journal of Molecular Biology*, 5(157):671–679, 1982.

K. Yu and V. Tresp. Heterogenous data fusion via a probabilistic latent-variable model. In Christian Müller-Schloer, Theo Ungerer, and Bernhard Bauer, editors, *ARCS*, volume 2981 of *Lecture Notes in Computer Science*, pages 20–30. Springer, 2004.