Probabilistic Topographic Information Visualisation

IAIN TIMOTHY RICE

Doctor Of Philosophy

ASTON UNIVERSITY

June 2015

ASTON UNIVERSITY

Probabilistic Topographic Information Visualisation

IAIN TIMOTHY RICE

Doctor Of Philosophy, 2015

Report Summary

The focus of this thesis is the extension of topographic visualisation mappings to allow for the incorporation of uncertainty. Few visualisation algorithms in the literature are capable of mapping uncertain data with fewer able to represent observation uncertainties in visualisations. As such, modifications are made to NeuroScale, Locally Linear Embedding, Isomap and Laplacian Eigenmaps to incorporate uncertainty in the observation and visualisation spaces. The proposed mappings are then called Normally-distributed NeuroScale (N-NS), T-distributed NeuroScale (T-NS), Probabilistic LLE (PLLE), Probabilistic Isomap (PIso) and Probabilistic Weighted Neighbourhood Mapping (PWNM). These algorithms generate a probabilistic visualisation space with each latent visualised point transformed to a multivariate Gaussian or T-distribution, using a feed-forward RBF network.

Two types of uncertainty are then characterised dependent on the data and mapping procedure. Data dependent uncertainty is the inherent observation uncertainty. Whereas, mapping uncertainty is defined by the Fisher Information of a visualised distribution. This indicates how well the data has been interpolated, offering a level of 'surprise' for each observation.

These new probabilistic mappings are tested on three datasets of vectorial observations and three datasets of real world time series observations for anomaly detection. In order to visualise the time series data, a method for analysing observed signals and noise distributions, Residual Modelling, is introduced.

The performance of the new algorithms on the tested datasets is compared qualitatively with the latent space generated by the Gaussian Process Latent Variable Model (GPLVM). A quantitative comparison using existing evaluation measures from the literature allows performance of each mapping function to be compared.

Finally, the mapping uncertainty measure is combined with NeuroScale to build a deep learning classifier, the Cascading RBF. This new structure is tested on the MNist dataset achieving world record performance whilst avoiding the flaws seen in other Deep Learning Machines.

Keywords: Visualisation, Uncertainty, Dissimilarity, RBF network

Contents

List of Figures

List of Tables

List of Frequently Used Symbols

$\mathbf{x}_i$	The i^{th} observation vector
$\mathbf{t_i}$	The target corresponding to observation i
$\mathbf{y_i}$	The i^{th} visualised vector corresponding to observation X_i
Γ	The Gamma function
Λ	The diagonal matrix of eigenvalues in descending order
ϕ	Nonlinear function or functional
$\boldsymbol{\phi}_i$	The nonlinear vector given by $\phi\left(d(X_i, C_j)\right)$
Φ	The matrix set of $\boldsymbol{\phi}_i$ vectors of dimensions $N \times M$
Ψ	The Digamma function
Σ	A covariance matrix
$A^{\dagger}$	The Moore-Penrose pseudo-inverse of a matrix A
C_j	The j^{th} centre of an RBF network
D	A square $N \times N$ dissimilarity matrix where the ij^{th} element is given by $d(i, j)$
$d(i, j)$	A pairwise dissimilarity measure between observations or latent points i and j.
$d_x(i, j)$	The dissimilarity between observations i and j
$d_y(i, j)$	The dissimilarity between visualised points i and j
$E_{\mathbf{x}}$	The expectation over $\mathbf{x}$
$I_{O \times P}$	An augmented Identity matrix of the first P columns of the Identity matrix I_O
I_P	The Identity matrix of dimensions $P \times P$
M	The number of centres, C_j, in an RBF network
N	The number of observations
$p(z)$	The probability distribution over z
S	An observed or estimated covariance matrix
T	The matrix set of targets t_i

W	A weight matrix
X	The set of all observations X_i. In the case where X_i is a vector, $\mathbf{x_i}$, X is a matrix of dimensions $N \times O$.
X^*	A new, unseen observation
$X_{t-m:t}$	A delay matrix of observations from time $t-m$ to time t (current)
Y	The matrix set of visualised points $\mathbf{y_i}$ of dimensions $N \times P$
$\mathbb{R}^O$	Set of real numbers in observation dimension, O
$\mathbb{R}^P$	Set of real numbers in latent / visualised dimension, P
$\mathcal{N}$	A Gaussian distribution

To Becky and Ryan

Acknowledgements

First and foremost I am grateful to my supervisor Professor David Lowe for all of the support, guidance and encouragement I have received throughout my time at Aston.

I wish to thank Thales, EPSRC and the KTN for their financial support which has allowed me to complete my PhD. I am particularly thankful to Les Hart, Rob Taylor, Geoff Williams, David Allwright and Roger Benton for their support on what has been an entirely different collaboration project for Thales.

Throughout my PhD studies I have received the strongest support from my wife Becky and son Ryan, as well as from my parents Joe and Joyce and from Tina and Shaun, without whom this work would not have been possible.

1 Introduction

> ‘If people do not believe that mathematics is simple, it is only because they do not realise how complicated life is.’
>
> - John von Neumann

1.1 Motivation

The work in this thesis stems from the inescapable fact that real world data is, in some way or another, uncertain. Data uncertainties are typically characterised as the result of the observation, measurement or analysis frameworks. Moreover, the data we are often most interested in is complex and, in the case it is vectorial, high dimensional. Non-vectorial data poses its own set of unique problems. With these elements coupled it makes the task of understanding and generating reliable conclusions from data a difficult task. The mathematical analysis performed on such data typically conforms to the

general supervised regression or classification framework, involving a mapping from data observations to a set of targets. These scenarios have dominated research in pattern analysis over the past fifty years, [1],[2],[3].

Sometimes, however, there don't exist any targets to map the data to. In this case one approach is to use summary statistics as a descriptor for data, or some feature-based representation of the data. An alternative, and often more useful analysis tool, is to generate a low-dimensional visualisation space, allowing for human interpretation of the data. The ability of humans in deciphering patterns in data, taking into account expertise, historical information or additional information not characterised in observations can surpass that of automated systems. Mapping observed data to a space where it can be visually interpreted relies on a visualisation algorithm. The 'optimum' positions of data observations in this (typically 2 or 3-dimensional) visualisation space depends on the algorithm being used. In general, the aim of such a mapping algorithm is to preserve global or local data structure, in which case they are called 'topographic'. A prominent issue in the field of data visualisation is that many algorithms, for instance Locally Linear Embedding [4] or Isomap [5], suffer in quality when data is noisy, or uncertain. In addition to this there are often assumptions made as to the underlying manifold on which observations sit. These deficiencies presents a significant problem for real world data analysis.

In order to tackle the data uncertainty problem, this thesis extends current algorithms to incorporate inherent observation uncertainty and the uncertainty imposed by the mapping from observation to visualisation space. A framework for representing these uncertainties in visualisations is also introduced, allowing for an informative visualisation of data. Finally it is shown that the benefits of a thorough approach to manifold leaning, through topographic mapping, extends beyond data visualisation to areas such as deep learning classifiers.

1.2 Contributions

In this thesis a probabilistic framework is outlined for topographic information visualisation accounting for uncertainty. Specifically:

- Probabilistic extensions to NeuroScale, Locally Linear Embedding, Isomap and Laplacian Eigenmaps are introduced, accounting for observation uncertainty, allowing for feed-forward projection of new data.
- A framework for interpreting observation uncertainty and the imposed mapping uncertanty in visualisation spaces is outlined.
- A novel method for detecting anomalies in time series data using topographic visualisation is described.
- A new form of deep learning machine consisting of topographically pre-trained RBF networks is implemented in a classification setting.

1.3 Thesis Organisation

Chapter 2 offers an introductory background to some of the popular methods for visualising data. Three criteria for quantitatively analysing visualisation performance are also outlined. Chapter 3 extends the deterministic mappings outlined in chapter 2 to allow for observation uncertainty. Chapter 4 proposes a method for representing both the uncertainties generated by observations and the visualisation mapping itself. Chapter 5 implements the methods of chapters 3 and 4 on three vectorial datasets, accounting for data uncertainty. In chapter 6 a process for visualising anomalies in time series data is introduced and demonstrated on three datasets. Chapter 7 combines topographic mapping with a deep learning machine in a classification setting. Finally, chapter 8 concludes the thesis.

2 Background

> 'A mathematician is a device which turns coffee into theorems.'
>
> - Alfred Rényi

2.1 Data Visualisation

This chapter forms an introductory section for the thesis describing the tools used for visualisation of data.

Firstly, the notion of visualisation must be described in terms of some data. The simplest and most intuitive case being where the data consists of a set of vectors. A few popular visualisation mechanisms require the data to be of this form (for instance [6], [7] and [8]). The purpose of a visualisation algorithm in this case is to reduce the dimensionality of these vectors such that the observations, $\mathbf{x} \in \mathbb{R}^O$, are mapped by some function to a new co-ordinate system; $\mathbf{y} \in \mathbb{R}^P$. P should be lower than O and is typically two or three

so that the new points, $\mathbf{y}$, can be visually interpreted.

Many other visualisation algorithms do not require pointwise observations and can construct a visualisation space with only relative pairwise dissimilarities, in the form of a dissimilarity matrix, D, as inputs, the most commonly used being the Sammon map [9]. This allows for perceptual analysis of more abstract notions than data-points; for instance, in visualising different time series, probability distributions or graphs. This is a significant benefit since these notions cannot be properly characterised by an observed vector point.

2.1.1 Methods

As with all areas of Machine Learning, there exist multiple different methods for construction of the functional mappings which generate a visualisation space. Each of these offer different results depending on the data and mapping parameters. These methods can be split into 3 groups:

1. Dissimilarity Mappings (section 2.2)
2. Graph Distance Mappings (section 2.3)
3. Latent Variable Models (section 2.4)

A taxonomy diagram showing examples of visualisation algorithms conforming to these groups and their links is shown in figure 2.1. Some of these algorithms are not included in this thesis but are shown for completeness. The Geodesic Nonlinear Mapping (GNLM) [10] is a special case of the Sammon map with Geodesic dissimilarities, but the Sammon map in general does not specify the input dissimilarity; so GNLM is not discussed in this thesis. Curvilinear Component Analysis (CCA) [11], and also Curvilinear Distance Analysis (CDA) [12], extnsions to the Sammon map (and GNLM) requiring the specification of a neighbourhood weighting function and, for many popular function choices have little global impact on the visualisations generated. As such these are not discussed in this thesis. The Deep GP [13] and T-SNE [14] are not topographic,

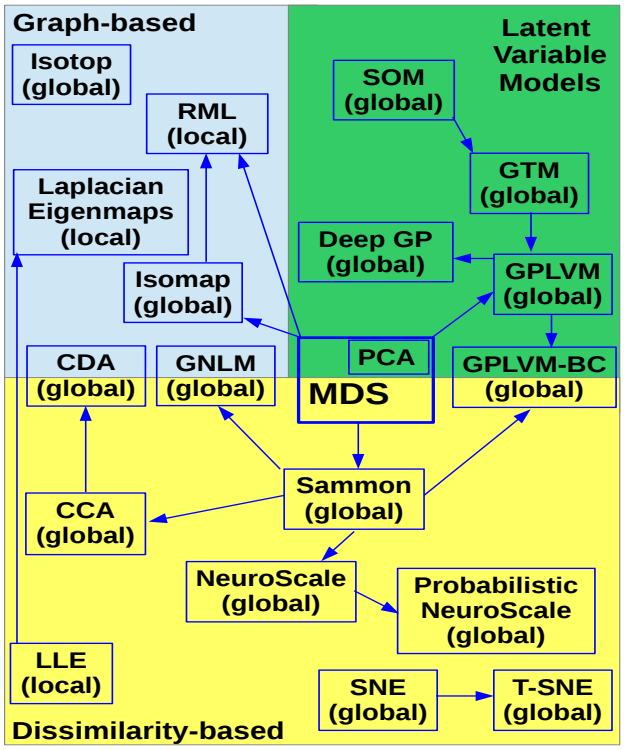


Figure 2.1: Taxonomy diagram showing the grouping and links between popular visualisation algorithms. Arrows indicate a connection between algorithms, with arrows showing extensions to previous techniques. Most algorithms can be shown to be extensions to, or reliant upon, Multidimensional Scaling (MDS), of which PCA is a special case.

but it may not be clear initially why and as such are included in Appendix B.

Riemannian Manifold Learning (RML) [15] is a principled local approach to manifold learning with impressive results. It does, however, require a background in Differential Geometry and is thus outside the scope of this thesis. Isotop [16] is an altogether different method for generating data visualisations, again with impressive mapping performance [17]. Despite this there is no clear cost function or knowledge of how it generates these visualisations and as such is not included in this thesis.

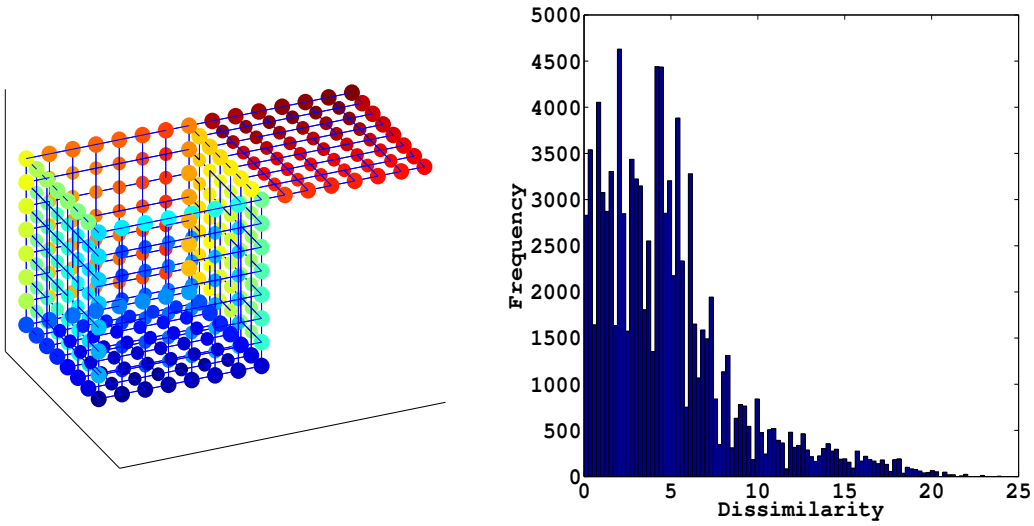
Firstly, Principal Component Analysis (PCA) [6] will be discussed in section 2.2. It will be shown that since it is a special case of metric Multidimensional Scaling (MDS) [18], it can be thought of as a dissimilarity-based mapping. Following this Locally Linear Embedding (LLE) [4] and Sammon mapping [9] will be introduced. These methods reconstruct observations by attempting to preserve the relative dissimilarities between the observations. Graph distance mappings including Isomap [5] and Laplacian Eigenmaps (LE) [19] attempt to describe the observation space with a connected graph

and preserve the graph distances when generating visualised points. Latent Variable models such as Generative Topographic Mapping (GTM) [7] and the Gaussian Process Latent Variable Model (GPLVM) [8] attempt to define the most likely latent visualisation space which generates the observation space. These methods impose specific restrictions on the latent space and require observations to be pointwise vectors. The figures generated in this thesis rely upon Matlab toolboxes for their implementation. The list below shows the algorithms and their relevant toolboxes:

- PCA/MDS, Isomap, LLE, Sammon Mapping, LE - drtoolbox [20],
- GTM, NeuroScale - Netlab toolbox [21],
- GPLVM - GPMat toolbox [22].

These toolboxes are widely used and thus considered robust for analysis in this thesis.

In order to gain insight into the differences between the algorithms, and to later introduce mapping performance criteria, a comparison dataset will be used for visualisation by all algorithms introduced in this chapter. The Open Box dataset [23] is a suitable benchmark, existing in 3-dimensional space with six 2-dimensional connected faces, one of which is an open lid. This is shown in figure 2.2a. The structure is extensively analysed using variants of the nonlinear MDS in [24] and used to compare many different visualisation algorithms in [17]. The colouring of points represents the topological ordering of observations. The visualisations generated in this chapter should preserve the local neighbourhoods, keeping points from the base (dark blue), front face (cyan), sides (orange and light blue), connected side (yellow) and lid (red) in similar groupings. This benchmark serves as a comparison; however, it is an entirely artificial dataset and is therefore useful for visual comparison but not for drawing definitive conclusions as to which algorithm is ‘best’. The histogram of dissimilarities, where the dissimilarities between observations are the Euclidean distance, is shown in figure 2.2b. It is clear that the structure consists largely of local neighbourhoods with $d_{ij} \leq 5$. Larger dissimilarities exist because of the distance between the points on the lid at the far right of the plot and those in the bottom left corner of the front face.



(a) (b)

Figure 2.2: 3-dimensional plot of the Open Box dataset. It is clear that the structure is composed of six 2-dimsensional planes with an open lid (red). The points here have been connected to their nearest neighbours to assist in checking how the visualisation algorithms distort neighbourhoods in the mapping process (left). The histogram of dissimilarities is also shown where the dissimilarities are taken as the Euclidean distance between points (right).

2.2 Dissimilarity Mappings

2.2.1 The PCA/MDS Mapping

Principal Component Analysis (PCA) and Multi-Dimentional Scaling (MDS) are essentially different sides of the same coin as they both construct the same latent representations through slightly different methods. Firstly, PCA is introduced prior to explaining the process of MDS, following which the link between the two will be shown. PCA has been the standard method for visualising data across multiple fields for many years and is the starting point for many more robust visualisation algorithms shown in figure 2.1.

PCA can be derived from multiple perspectives, the two most popular being the minimal reconstruction error approach [6] or maximal preserved variance and decorrelation [25]. In this thesis the former is the more suitable so it will be introduced in that format. The minimal reconstruction error approach was derived by Pearson [6] where the dual

relationship in the linear model is defined as:

$$\mathbb{R}^O \to \mathbb{R}^P, \mathbf{x}_i \to \mathbf{y}_i = W^T \mathbf{x}_i, \tag{2.1}$$

$$\mathbb{R}^P \to \mathbb{R}^O, \mathbf{y}_i \to \mathbf{x}_i = W \mathbf{y}_i. \tag{2.2}$$

W is an orthogonal matrix such that $W^T = W^\dagger$, where $W^\dagger$ is the Moore-Penrose Pseudo-Inverse of W. This ensures that $W^T W = W^\dagger W = I_P$. The P subscript here indicates the identity matrix is a square matrix of dimensions $P \times P$. The squared reconstruction error is given by:

$$E_{PCA} = E_X \left[\|\mathbf{x}_i - W W^T \mathbf{x}_i\|_2^2 \right],$$

where $\|.\|_2$ is the Euclidean distance. In the ideal case of $\mathbf{x}_i$ generated by equation (2.2), the mapping results in a reconstruction error of zero. This is because W will be full rank, ensuring $WW^T = I_O$ where I_O is the $O \times O$ identity matrix. Unfortunately this is in almost all real situations not the case. In order to determine W, the above expectation can be expanded as follows:

$$\begin{aligned}
E_{PCA} &= E_X \left[\left(\mathbf{x}_i - W W^T \mathbf{x}_i \right)^T \left(\mathbf{x}_i - W W^T \mathbf{x}_i \right) \right], \\
&= E_X \left[\mathbf{x}_i^T \mathbf{x}_i - 2 \mathbf{x}_i^T W W^T \mathbf{x}_i + \mathbf{x}_i^T W W^T W W^T \mathbf{x}_i \right], \\
&= E_X \left[\mathbf{x}_i^T \mathbf{x}_i - 2 \mathbf{x}_i^T W W^T \mathbf{x}_i + \mathbf{x}_i^T W W^T \mathbf{x}_i \right], \\
&= E_X \left[\mathbf{x}_i^T \mathbf{x}_i - \mathbf{x}_i^T W W^T \mathbf{x}_i \right], \\
&= E_X \left[\mathbf{x}_i^T \mathbf{x}_i \right] - E_X \left[\mathbf{x}_i^T W W^T \mathbf{x}_i \right].
\end{aligned}$$

Splitting the error into these two parts allows for the optimum W to be found. The minimsisation of E_{PCA} is given by maximising $E_X \left[\mathbf{x}_i^T W W^T \mathbf{x}_i \right]$, found when $WW^T = I_O$. Since data samples in X are finite, we can approximate this expression with the sample mean:

$$E_X \left[\mathbf{x}_i^T W W^T \mathbf{x}_i \right] \approx \tfrac{1}{N} \textstyle\sum_i^N \left(\mathbf{x}_i^T W W^T \mathbf{x}_i \right) = \tfrac{1}{N} tr \left(X^T W W^T X \right).$$

Here, X is the matrix set of observations, $\{\mathbf{x}_i\}_{i=1:N}$, such that the i-th row of X is $\mathbf{x}_i$ with dimensions $O \times N$. Using a singular value decomposition, $X = V \Sigma U^T$ with U and V

orthonormal matrices ($U^TU = I_N$ and $V^TV = I_O$) and Σ matrix with the diagonal elements given by the singular values, E_{PCA} can be re-written as:

$$E_{PCA} = E_X\left[\mathbf{x}_i^T\mathbf{x}_i\right] - E_X\left[\mathbf{x}_i^TWW^T\mathbf{x}_i\right] = tr\left(X^TX\right) - tr\left(X^TWW^TX\right).$$

Since $tr\left(X^TX\right) = tr\left(U\Sigma^TV^TV\Sigma U^T\right)$ from the singular value decomposition and using the following two relations from [26, p. 6]:

- $tr(ABC) = tr(CBA)$,
- $tr(X^TX) = tr(\Sigma^T\Sigma)$,

it is clear that:

$$\begin{aligned} E_{PCA} &= tr\left(U\Sigma^TV^TV\Sigma U^T\right) - tr\left(U\Sigma^TV^TWW^TV\Sigma U^T\right), \\ E_{PCA} &= tr\left(U^TU\Sigma^TV^TV\Sigma\right) - tr\left(U^TU\Sigma^TV^TWW^TV\Sigma\right), \\ E_{PCA} &= tr\left(\Sigma^T\Sigma\right) - tr\left(\Sigma^TV^TWW^TV\Sigma\right). \end{aligned}$$

In the case where $P = O$, E_{PCA} is zero for $W = V$. Since the typical use of PCA is for dimension reduction and $P < O$ an approximation must be used to make W as linearly close to V as possible, namely $W = VI_{O\times P}$. Here $I_{O\times P}$ is a matrix made up of the first P columns of the identity matrix I_O. The P dimensional latent variables are approximated by computing:

$$\hat{\mathbf{y}}_{PCA}^i = W^T\mathbf{x}_i = I_{P\times O}V^T\mathbf{x}_i. \tag{2.3}$$

Classical multidimensional scaling (MDS) [18] will now be outlined as the other side of the coin to PCA. MDS seeks to preserve vector inner products from observations when generating visualisation points. Using a linear model, as with PCA, we denote the inner product matrix S by:

$$\begin{aligned} S &= X^TX, \\ &= (WY)^T(WY), \\ &= Y^TW^TWY, \\ &= Y^TY. \end{aligned}$$

MDS has a particularly useful property that X need not be a vectorial observation. Often observations are characterised by a pairwise dissimilarity matrix, D, of dimensions $N \times N$, where the ij^{th} element, $D_{ij} = d(i, j)$, is the pairwise dissimilarity between observations i and j. From D, the equivalent inner product matrix S, known as the Gram matrix, is found by double centering:

$$S = -\frac{1}{2}\left(D^2 - \frac{1}{N}D^2\mathbf{1}_N\mathbf{1}_N^T - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T D^2 + \frac{1}{N^2}\mathbf{1}_N\mathbf{1}_N^T D^2\mathbf{1}_N\mathbf{1}_N^T\right), \tag{2.4}$$

where D^2 is the element-wise square of the matrix D. This double centering removes the row and column means before adding back the total mean. In order to find Y the eigendecomposition of S is performed:

$$\begin{aligned} S &= U\Lambda U^T, \\ &= (U\Lambda^{\frac{1}{2}})(\Lambda^{\frac{1}{2}}U^T), \\ &= (\Lambda^{\frac{1}{2}}U^T)^T(\Lambda^{\frac{1}{2}}U^T), \end{aligned}$$

The optimal linear reconstruction (in a Least-Mean Square sense) of Y, $\hat{Y}$, is then given by:

$$\hat{Y}_{MDS} = I_{P\times N}\Lambda^{\frac{1}{2}}U^T, \tag{2.5}$$

where $I_{P\times N}$ is the first P columns of the $N \times N$ identity matrix I_N. This ensures that only the required P dimensions are recovered by the MDS mapping algorithm. The embeddings in equations (2.3) and (2.5) can be shown to be equivalent [17, p. 74-75]:

$$\begin{aligned} \hat{Y}_{PCA} &= \hat{Y}_{MDS}, \\ I_{P\times O}V^T X &= I_{P\times N}\Lambda^{\frac{1}{2}}U^T, \\ I_{P\times O}V^T V\Sigma U^T &= I_{P\times N}(\Sigma^T\Sigma)^{\frac{1}{2}}U^T, \\ I_{P\times O}\Sigma U^T &= I_{P\times N}\Sigma U^T. \end{aligned}$$

The PCA/MDS embedding of the open box dataset is given in figure 2.3.

2.2.2 Locally Linear Embedding

MDS attempts to preserve global dissimilarities in visualisation spaces, however this can lead to a good overall mapping at the expense of good local reconstruction. Locally

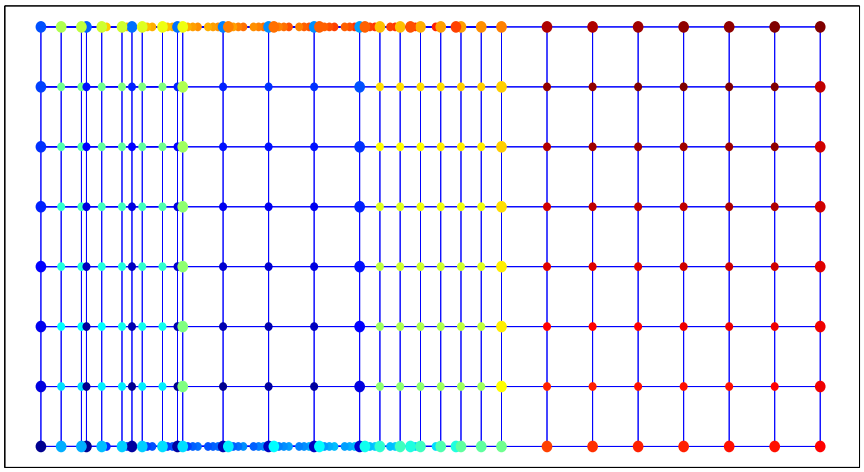


Figure 2.3: Open box embedded by PCA/MDS. The embedding is a poor representation of the original box as it is a top oriented squashed view. The top of the box remains separated from the other five sides, however the two open sides of the box have points overlapping which is not a true representation of their relative position in the observation space. This is because the linear relationship of equation (2.2) does not hold for the observed manifold, X.

Linear Embedding (LLE) [4] attempts to preserve dissimilarities in observation space by describing observations in terms of their local neighbours. This is done by imposing a locally Euclidean space on a manifold. The observed manifold is then characterised by a series of weighted neighbourhoods (either by k-nearest neighbours or an ε-ball). The visualisation space is constructed in a two step process.

The first step is to determine the weights associated with each neighbourhood, minimising the following error:

$$E_{LLE}(W) = \sum_{i=1}^{N} \|\mathbf{x}_i - \sum_{j \in \mathbb{N}(i)} W_{ij}\mathbf{x}_j\|^2, \tag{2.6}$$

where $\mathbb{N}(i)$ is the set containing all neighbours of $\mathbf{x}_i$. This essentially sums all squared distances between an observation and its locally linear reconstruction. Constraints are imposed on W such that:

- $\sum_j W_{ij} = 1$,
- $W_{ij} \geq 0$,

- $W_{ij} = 0 \; \forall \; j \notin \mathbb{N}(i)$.

The weights are determined by re-casting the error:

$$E_i = |\mathbf{x}_i - \sum_j W_{ij}\boldsymbol{\eta}_j|^2 = |\sum_j W_j(\mathbf{x} - \boldsymbol{\eta}_j)|^2 = \sum_{jl} W_j W_l C(i)_{jl},$$

where $\{\boldsymbol{\eta}_j, j = 1,\ldots,k\}$ are the set of k nearest neighbours of a point i. The second part comes from the first constraint above. $C(i)_{jl} = (\mathbf{x}_i - \boldsymbol{\eta}_j).(\mathbf{x}_i - \boldsymbol{\eta}_l)$ is the local covariance matrix. The weights corresponding to each observation 'i' denoted by the vector, $\mathbf{w}_i$, are then given by:

$$W_{ij} = \frac{\sum_l C(i)^{-1}_{jl}}{\sum_{jl} C(i)^{-1}_{jl}}, \tag{2.7}$$

for $j = 1,\ldots,k$, which are concatenated into the weight matrix $W = \{\mathbf{w}_i\}_{i=1:N}$. Alternatively W can be found by solving the linear system:

$$\sum_j C(i)_{jl} W_{il} = 1,$$

and rescaling so that $\sum_j W_j = 1$. It is proposed in [4] that if C_{jl} is singular or nearly singular the following augmentation can be used, such that:

$$C_{jl} \leftarrow C_{jl} + \left(\frac{\Delta^2 tr(C_{jl})}{K}\right) I,$$

where Δ^2 is small compared to the trace of C_{jl}. This augmentation ensures that the matrix can be inverted thanks to the 'jitter' term (right). This is a typical jitter modification used to ensure numerically unstable matrices are invertible. Typical values of Δ, for instance as used in [17], are 10^{-3}. Alternatively a simpler jitter such as ΔI can be added to ensure that the matrix is not singular in a less principled way.

The second step consists of embedding the points using their local reconstruction. This amounts to manipulating the visualised points $\mathbf{y}_i$ to minimise the error with respect to the set of latent points, Y:

$$E_{LLE}(Y) = \sum_{i=1}^{N} \|\mathbf{y}_i - \sum_{j \in \mathbb{N}(i)} W_{ij}\mathbf{y}_j\|^2, \tag{2.8}$$

where W is given from equation (2.7). Two constraints are imposed upon Y:

- $\sum_i \mathbf{y}_i = 0 \Rightarrow$ centred around the origin
- $C_{YY} = \frac{1}{N} YY^T = I \Rightarrow$ unit covariance so $E_{LLE}(Y)$ cannot be minimised by arbitrary rotations or rescalings.

The embedding is found by the well-posed eigenvalue problem:

$$E_{LLE}(Y) = \sum_i |\mathbf{y}_i - \sum_{j \in N(i)} W_{ij}\mathbf{y}_j|^2 = \sum_i |\sum_{j \in N(i)} W_{ij}(\mathbf{y}_i - \mathbf{y}_j)|^2 = \sum_{ij} M_{ij}\mathbf{y}_i^T\mathbf{y}_j,$$

using the same properties as above. The entries of M are given by:

$$M = (I - W)^T(I - W),$$

which is sparse (since the elements of W are non-zero only for the k neighbours of each point i), symmetric and positive definite. The co-ordinates of Y are found by computing the bottom $P+1$ eigenvectors of M (where P is the visualisation dimension, e.g. 2) and discarding the bottom eigenvector as its eigenvalue is 0 (since $\sum_i \mathbf{y}_i = 0$):

$$Y_{LLE} = \hat{U},$$

where $\hat{U}$ is the bottom $N-1 : N-P$ eigenvectors. This ensures that the best linear reconstruction of the neighbourhoods of X are given by Y.

The Open Box visualisation using LLE with four neighbours (the same as that of [17]) is shown in figure (2.4). In contrast to the PCA mapping, the sides of the box (light blue and orange) are no longer flattened. The six surfaces of the box are all well reconstructed in themselves, appearing as parallelograms. On the other hand, the relative distances of the sides with respect to the bottom of the box (dark blue) are not well preserved. This is clear from the overlap of points in visualisation space which are not close in observation space, for instance the front face which overlaps the bottom face. The good local reconstruction comes at the cost of the global distribution of points caused by the LLE error function.

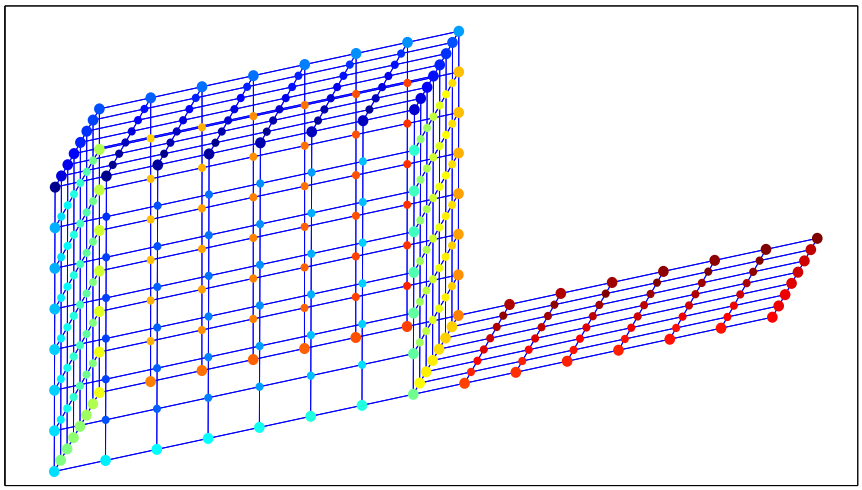


Figure 2.4: Open box embedded by LLE. The six surfaces of the box are all well reconstructed in themselves, appearing as rectangles. The relative distances of the sides (light blue and orange) with respect to the bottom of the box (dark blue) are not well preserved.

2.2.3 Sammon Mapping & NeuroScale

This section will outline the Sammon Mapping process for visualisation before describing the NeuroScale mapping.

Sammon map

Taking a more global approach to visualisation, the Sammon map [9] attempts to construct a set of visualisation points, Y, by preserving relative dissimilarities. This is done by matching the dissimilarity matrices, as opposed to inner product matrices as MDS does. This constructs more reliable visualisations, as shown in [17]. Denoting the dissimilarities between observations $d_x(i,j)$ and between visualised points $d_y(i,j)$ the error to be minimised is:

$$E_{Sammon} = \frac{1}{c} \sum_{i,i<j}^{N} \frac{(d_x(i,j) - d_y(i,j))^2}{d_x(i,j)}, \tag{2.9}$$

where the normalisation constant $c = \sum_{i,i<j}^{N} d_x(i,j)$. This function is commonly known as the Standardised Residual Sum of Squares (STRESS) measure. It is important to note that no assumption is made about $d_x(i,j)$ and so can be application-specific (e.g. [27] or [28]). However, it is typical that for vector observations $d_x(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$. $d_y(i,j)$ is

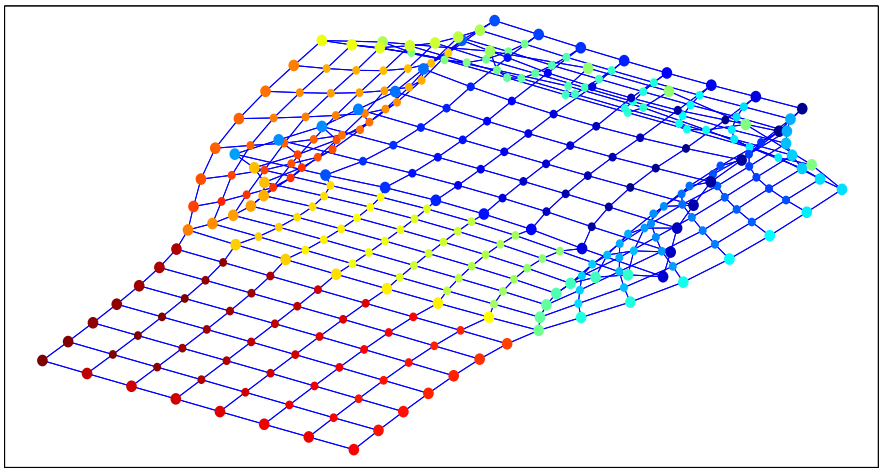


Figure 2.5: Open box embedded by Sammon mapping. The sides of the box are still attached to the top and bottom faces, but are correctly placed directly on top. The front face of the box opposite the open lid is squashed in a similar way to that of PCA/MDS and the bottom corners appear torn.

usually taken to be the Euclidean distance; $d_y(i,j) = \|\mathbf{y}_i - \mathbf{y}_j\|_2$. Originally Sammon proposed an iterative quasi-Newton style update rule such that:

$$\mathbf{y}_i \leftarrow \mathbf{y}_i - \alpha \frac{\partial E_{Sammon}/\partial \mathbf{y}_i}{|\partial^2 E_{Sammon}/\partial \mathbf{y}_i^2|},$$

where α is a learning rate. In reality this can result in quickly finding poor local minima so other gradient-based optimisation procedure can produce more reliable visualisation spaces. PCA or MDS can be used as an initialisation of Y, but this can provide minima close to, but not as optimal as, the global minima. The derivative of (2.9) is given by:

$$\frac{\partial E_{Sammon}}{\partial \mathbf{y}_i} = \frac{-2}{c} \sum_{j,j\neq i} \frac{d_x(i,j) - d_y(i,j)}{d_x(i,j) d_y(i,j)} (\mathbf{y}_i - \mathbf{y}_j), \tag{2.10}$$

where c is again given by $c = \sum_{i,i<j}^{N} d_x(i,j)$. The use of Quasi-Newton optimisation is not essential here, other gradient-based optimisers could be used e.g. Scaled Conjugate Gradients (SCG). Unlike MDS, PCA and LLE, Sammon Maps embed in a nonlinear way.

The 2-dimensional embedding of the Open Box dataset using Sammon mapping is shown in figure 2.5. The mapping is optimised using Quasi-Newton gradient descent

with random initialisation to avoid the potential PCA-initialisation sink. The nonlinear embedding process allows for curvature to be imposed on the manifold, by not placing a linear mapping on the observation space. This causes the sides of the box to still be attached to the top and bottom faces, without being placed directly on top. The front face of the box opposite the open lid is squashed in a similar way to that of PCA/MDS and the bottom corners appear torn. Despite these inaccuracies the overall shape of the manifold can be easily recognised from the visualisation.

In [29] an extension to the Sammon map using feed forward Radial Basis Function (RBF) networks was outlined which will be described in the next section. An introduction to RBF networks is given in appedix A.

NeuroScale

The extension of the Sammon map using RBF's is called NeuroScale (NS). Variants using a Multi-Layer Perceptron network were also proposed in [30]. As already mentioned, the STRESS function, in contrast to the standard learning procedure of RBFs, requires nonlinear optimisation. Learning weights through gradient descent is the standard approach in the training of Artificial Neural Networks. However, a more robust and efficient method for training the NS RBF network was described in [31]. True observation targets, T, do not exist but the 'Shadow Targets' algorithm involves generating a series of synthetic targets, $\mathbf{t}_i$:

$$\mathbf{t}_i = \mathbf{y}_i - \alpha \frac{\partial E_{Sammon}}{\partial \mathbf{y}_i},$$

$$\hat{W} = \Phi^{\dagger} T,$$

$$\hat{Y} = \Phi \hat{W},$$

with $\frac{\partial E_{Sammon}}{\partial \mathbf{y}_i}$ given by equation (2.2.3). This iterative steepest descents process is repeated until convergence using α as a learning rate. The NS algorithm works best when Φ is as representative as possible of the data, i.e. when the number of centres is as close to the number of training points as possible. Unlike training in standard

parameterised machine learning tasks, NS cannot overtrain [29], [32]; performing implicit auto-regularisation due to the network centres and curvature with respect to the STRESS function. In addition to this, the RBF network is infinitely smooth meaning out of sample observations will also be topographically mapped. With this in mind a suitably interpolated data space in NS would generate an identical Open Box visualisation to that of the Sammon mapping and is therefore not included here. The Shadow Targets algorithm is used extensively in chapters 3 and 7 in this thesis as an optimisation procedure.

Standard NS was extended in [33] to account for uncertainty using isotropic Gaussians to describe observations and mapped points. This method will be discussed in chapter 3.

2.3 Graph Distance Mappings

Graph distance mappings take a slightly different approach to visualisation than dissimilarity-based mappings. They treat observations as objects of a graph to be represented in a visualisation space. Two methods are outlined; Isomap, relying on graph distances, and Laplacian Eigenmaps (LE) using the graph Laplacian for optimisation.

2.3.1 IsoMap

The Isomap algorithm [5] uses neighbourhood structures like LLE, k-neighbourhoods or ε-balls, to construct a graph characterising observations. Graph edges are labelled with Euclidean lengths, giving a sparse weighted graph (note that other dissimilarity measures can be used, though this is not common in the literature). The remaining graph distances between observations are computed in a pairwise manner using geodesic distances computed by Djikstra's [34] or Floyd's [35] algorithms and stored as a dissimilarity matrix, D (many implementations such as that in [20] use Djikstra's algorithm as default). This dissimilarity matrix is treated as an alternative to dissimilarities in MDS, but the embedding procedure is then identical for Isomap as for MDS.

The dissimilarity matrix D is converted into an inner product (Gram) matrix, S, by double centering (equation 2.4). As with MDS the eigendecomposition of S gives

$S = U\Lambda U^T$ with eigenvectors U and eigenvalues as diagonal elements of Λ. The P-dimensional embedding of the observations X, given by D, is:

$$Y = I_{P\times N}\Lambda^{\frac{1}{2}}U^T, \tag{2.11}$$

This embedding attempts to minimise the standard MDS error:

$$E_{Iso} = \sum_{i,j} \|d_x(i,j) - d_y(i,j)\|^2, \tag{2.12}$$

by inner product eigendecomposition. Isomap is an efficient and popular tool for creating representative visualisations of complex data. Geodesic distances are a much more realistic dissimilarity between points on a manifold than the assumption that a manifold is Euclidean, for example in Riemannian manifolds [36]. This fact is reinforced by the work in Machine Learning on Riemannian Manifolds (for instance [36], [15]). There are three particular weaknesses worth noting with Isomap:

1. The sensitivity of the map to choice of k or ε,
2. The calculation of dissimilarities in the presence of noise or uncertainty.
3. The linear embedding formed by MDS.

These can cause short-circuits in the graph construction leading to an incorrect over- or underestimation of the distance between observations. An important note is that k or ε should be chosen such that the graph is fully connected (no geodesic distances should be infinite). The embedding generated is a linear mapping and is therefore unable to appropriately characterise a highly nonlinear mapping function. An alternative method using geodesic dissimilarities was proposed in [37],[38]. These dissimilarities were combined with the Sammon map, relying on the benefits of the two methods, called the Geodesic Nonlinear Map (GNLM) . The training procedure for GNLM is the same as that of the Sammon map but with $d_x(i,j)$ given by geodesic distances and graph neighbourhoods.

The Isomap embedded box is shown in figure (2.6). A connected graph was achieved for

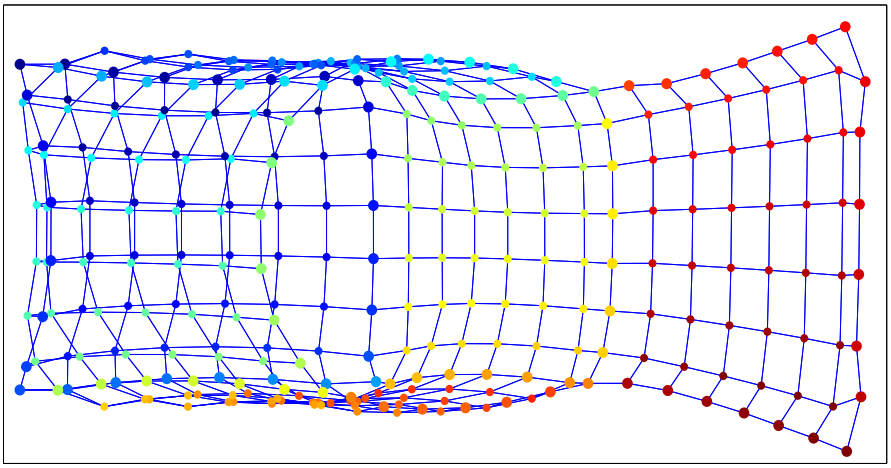


Figure 2.6: Open box embedded by Isomap using four neighbours. The front face of the box has been overlapped with the bottom of the shape and the mapping has imposed a curved surface on the box lid which is in fact rectangular in the original space. The box sides are squashed and therefore not representative of the original structure.

$k = 4$ neighbours and the overall box structure is clear. This seems an improvement on the LLE box, but there are still squashed sides similar to those of the PCA/MDS box. The front face of the box has been overlapped with the bottom of the shape and the mapping has imposed a curved surface on the box lid which is in fact rectangular in the original space. If the neighbourhood structure is extended to incorporate $k = 8$ neighbours, a more visually satisfactory image is achieved in figure 2.7. Here the lid is made approximately rectangular and there is less overlapping in the box sides due to the curvature imposed here. The distances from the box front to the bottom are more faithfully preserved with less overlapping. This does highlight a main issue with neighbourhood based mappings, namely that the change in visualisation spaces can be significant with changes in k or ε. It is noteworthy that the visualisation space remains largely unchanged for increases in neighbourhood size beyond eight neighbours. The only differences are seen in the lid and bottom becoming more rectangular, as in the MDS mapping of figure 2.3.

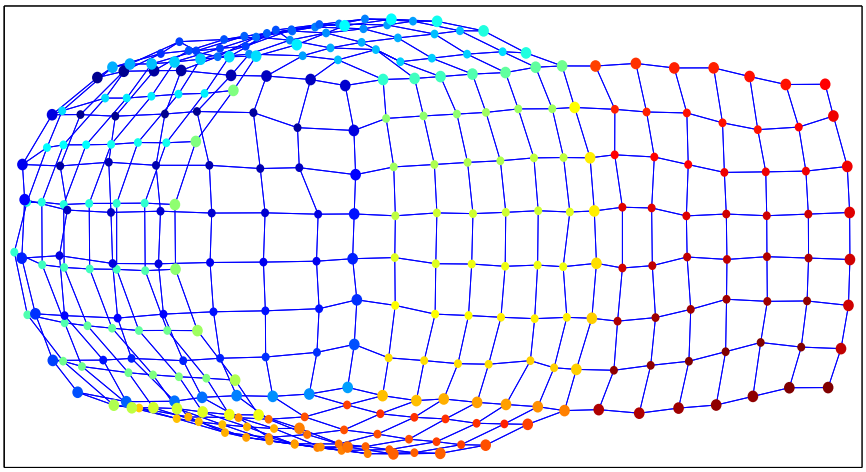


Figure 2.7: Open box embedded by Isomap with eight neighbours. The lid is made approximately rectangular and there is less overlapping in the box sides than the $k = 4$ mapping thanks to the curvature imposed here. The distances from the box front to the bottom are also more faithfully preserved with less overlapping.

2.3.2 Laplacian Eigenmaps

Laplacian Eigenmaps [19] is another graph-based embedding process with connections to LLE. The algorithm begins with a dissimilarity matrix, D, constructed by pairwise dissimilarities between observations. Following this step, a k or ε-ball neighbourhood is found. These neighbourhoods are used to build a graph with corresponding adjacency matrix A (an (i,j) binary matrix with elements 1 when observations (i,j) are adjacent, or neighbours, and 0 otherwise). The graph weight matrix W is then calculated by use of the 'heat kernel' (this is typically known as a Gaussian function in other areas of the literature, but is referred to here as the 'heat kernel' as it is in [19]):

$$W_{ij} = A_{ij} \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2T^2}\right), \tag{2.13}$$

where T is the temperature parameter. T is a user-specified parameter in the interval $[1, \infty)$ with popular choices being 1 or ∞. The dissimilarity measure does not necessarily need to be Euclidean and can be replaced with other measures capable of dealing with uncertainty as will be shown in section 3. A simpler weight function, often used in the literature is where T tends to infinity such that $W = A$. These weights are then used to

compute the graph Laplacian [39]:

$$L = W - G,$$

where G is a diagonal matrix with entries $G_{ii} = \sum_{j=1}^{N} W_{ij}$. In order to preserve the range of eigenvalues to create a standard embedding framework, and therefore a standard co-ordinate range, the Laplacian is then normalised:

$$L' = G^{-\frac{1}{2}} L G^{-\frac{1}{2}}.$$

This ensures the eigenvalues are within the range $0 \leq \lambda \leq 2$ [40]. Two Laplacians for entirely different graphs can then be compared without the issue of rescaling; only co-ordinate rotations need to be considered. The embedding error to be minimised is:

$$E_{LE} = \frac{1}{2} \sum_{i,j=1}^{N} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 W_{ij}, \tag{2.14}$$

subject to $YGY^T = I_{P \times P}$, ensuring that the error cannot be minimised by the arbitrary rescaling of Y. This error can be minimised by computing the eigendecomposition of $L' = U \Lambda U^T$. The embedded co-ordinates are found by taking the smallest $P+1$ eigenvectors and discarding the smallest eigenvector (since the above constraint forces the eigenvalue to be 0). This is because the error function in equation (2.14) can be re-written as '$tr(YL'Y)$', the solution of which is given by the same eigen-formulation. The remaining eigenvectors, $\hat{U}$ (of dimensions $P \times N$) give the embedding as:

$$Y = \hat{U} G^{\frac{1}{2}}.$$

Figure 2.8 shows the embedded Open Box computed by LE. The graph was constructed with four neighbours (creating a fully connected graph) as with Isomap and LLE; however, here the visualisation space remains largely unchanged with increasing k. The temperature parameter used here is set to unity, as is common in the literature, but tests were also run with increasing T (values uniformly sampled in the range $(1, 10^6)$),

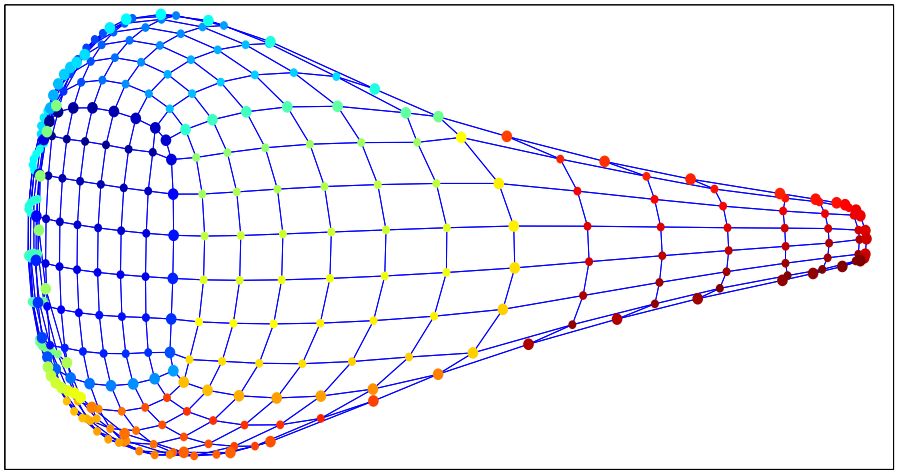


Figure 2.8: Open box embedded by Laplacian Eigenmapping with four neighbours. The algorithm has successfully unfolded the box from the open section. The box lid on the right hand side is separated from the bottom and front face. The front surface undergoes a level of squashing which is unrepresentative of the observations and the other two open sides to a lesser degree as well. The global and neighbourhood structure has however been preserved faithfully.

resulting in no change in the visualised coordinates. This is likely due to the relatively small and identical Euclidean distances between local points in the observation space, ensuring T in the heat kernel plays a relatively insignificant role. The artificial curvature imposed here appears on first inspection to have distorted the mapping. However, the algorithm has successfully unfolded the box from the open section. The box lid on the right hand side is separated from the bottom and front face. The front surface undergoes a level of squashing which is unrepresentative of the observations and the other two open sides to a lesser degree as well. The global and neighbourhood structures have however been preserved faithfully.

2.4 Latent Variable Models

The approach for generating visualisation spaces in Latent Variable Models (LVMs) is altogether different to that of dissimilarity preservation and graph-based mappings. LVMs assume a generative process in which observations are treated as the functional output and the latent points, (representing the visualisation space) which most likely

generated those observations, are found. In this sense PCA is also a LVM. LVMs therefore seek to learn the inverse function to dissimilarity preservation mappings. As such there are rigid assumptions with each method. There will be a change in notation from the previous sections; denoting observations by Y and latent points by X such that $Y = f(X)$, consistent with that of the literature (e.g. [7],[2],[8]). Two LVMs are discussed below; the Generative Topographic Mapping and the Gaussian Process Latent Variable Model. A currently popular LVM called the Deep Gaussian Process [13] is described in Appendix B and not here as it is not topographic.

2.4.1 Generative Topographic Mapping

The probabilistic extension of Kohonen's Self Organising Map [41] is known as the Generative Topographic Map (GTM) [7]. It is a generative model assuming data observations are created by a latent grid, often assumed rectangular.
The distribution of observations, $p(\mathbf{y}_i|\mathbf{x},W,\beta)$ are spherical Gaussian kernels, $\mathcal{N}(\mathbf{m}(\mathbf{x},W),\beta^{-1}I)$. The precision of each Gaussian is β and the mean given by a parameterised mean function with weights W, $\mathbf{m}(\mathbf{x},W)$. The distribution is therefore:

$$p(\mathbf{y}_i|\mathbf{x},W,\beta) = \left(\frac{\beta}{2\pi}\right)^{\frac{l}{2}} \exp\left[-\frac{\beta}{2}\|\mathbf{y}_i - \mathbf{m}(\mathbf{x},W)\|^2\right], \tag{2.15}$$

where l is the dimensionality of the observations. The prior distribution over the latent grid, $p(\mathbf{x})$, is given by:

$$p(\mathbf{x}) = \frac{1}{c}\sum_{r=1}^{c}\delta(\mathbf{x}-\mathbf{g}(r)) = \begin{cases} 0 & \text{if } \mathbf{x} \neq \mathbf{g}(r), \\ \frac{1}{c} & \text{if } \mathbf{x} = \mathbf{g}(r), \end{cases} \tag{2.16}$$

where the c points $\mathbf{g}(r)$ are on a (rectangular) grid. Visualisation of the grid requires knowledge of $p(\mathbf{x}|\mathbf{y},W,\beta)$ which by Bayes' rule is:

$$p(\mathbf{x}|\mathbf{y}_i,W,\beta) = \frac{p(\mathbf{y}_i|\mathbf{x},W,\beta)p(\mathbf{x})}{p(\mathbf{y}_i|W,\beta)}.$$

In order to compute this posterior, the marginal likelihood must be calculated:

$$p(\mathbf{y}_i|W,\beta) = \int p(\mathbf{y}_i|\mathbf{x},W,\beta)p(\mathbf{x})d\mathbf{x}.$$

This integral is typically analytically intractable for many prior choices but since the prior is a grid of delta points, the marginal likelihood becomes:

$$p(\mathbf{y}_i|W,\beta) = \frac{1}{c}\sum_{r=1}^{c} p(\mathbf{y}_i|\mathbf{g}(r),W,\beta).$$

The data log-likelihood is given by:

$$\mathcal{L}(W,\beta) = \sum_{i=1}^{N} \log\left(p(\mathbf{y}_i|W,\beta)\right).$$

The mean function, $\mathbf{m}(\mathbf{x},W)$, in equation (2.15) is typically taken to be an RBF network as described in appendix A. Other extensions using Gaussian Processes (GPs) and mean field approximations for the marginal likelihood have also been proposed [42]. Using an RBF network in this framework allows for an Expectation-Maximisation (EM) optimisation procedure outlined in Appendix C.

Visualisation

In order to generate the visualisation space, summary statistics of the posterior must be used. The mean can be approximated by:

$$\hat{\mathbf{x}}_i = \sum_{r=1}^{c} \mathbf{g}(r)p(\mathbf{g}(r)|\mathbf{y}_i) = \sum_{r=1}^{c} \mathbf{g}(r)P_{ir}(W^{opt},\beta^{opt}).$$

The posterior can be multimodal, which is revealed by a comparison of the mean and mode of the distribution, where the mode is given by:

$$\hat{\mathbf{x}}_i = \arg\max_{\mathbf{g}(r)} p(\mathbf{g}(r)|\mathbf{y}_i) = \arg\max_{\mathbf{g}(r)} P_{ir}(W^{opt},\beta^{opt}).$$

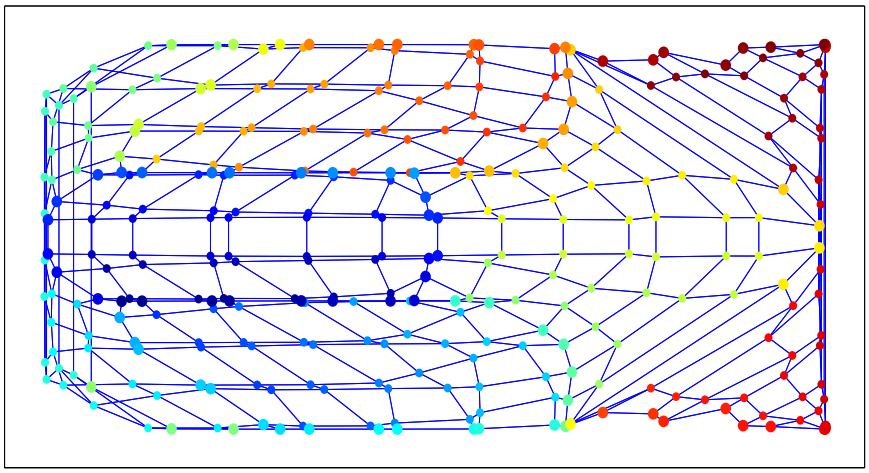


Figure 2.9: Open box embedded by GTM. The global structure has been unfolded from the open top, but the box front (cyan) and lid (red) are clearly squashed. It is clear that the mapping has torn the corners of the box open leading to a separation of naturally close observation points but the box floor (dark blue) and side connecting the floor to the lid (yellow) are faithfully represented.

Large discrepencies between means and modes of latent visualised points will indicate that a less reliable distribution has been created.

The GTM visualisation of the Open Box dataset is shown in figure 2.9 using a 10×10 latent grid and a 4×4 grid of basis functions with mean points shown, following the mapping procedure of [17]. The global structure has been unfolded from the open top, but the box front (cyan) and top (red) are clearly squashed. It is clear that the mapping has torn the corners of the box open, leading to a separation of naturally close observation points. On the other hand the box floor and side connecting the floor to the lid are faithfully represented. The posterior distribution is multimodal, causing many of the modal points to be separated from the mean. Many of the mode points sit atop each other which reinforces the notion that clusters in observation space are not ideally represented. The analysis was repeated with larger latent grids (20×20 and 30×30) where the transition between faces becomes smoother, but the tears appear in the same regions. Even with these larger latent grids the distribution is still multimodal.

Finally, the magnification factors [43] of the latent grid mapping are superimposed into the visualisation, showing which areas of the observation space have been magnified or

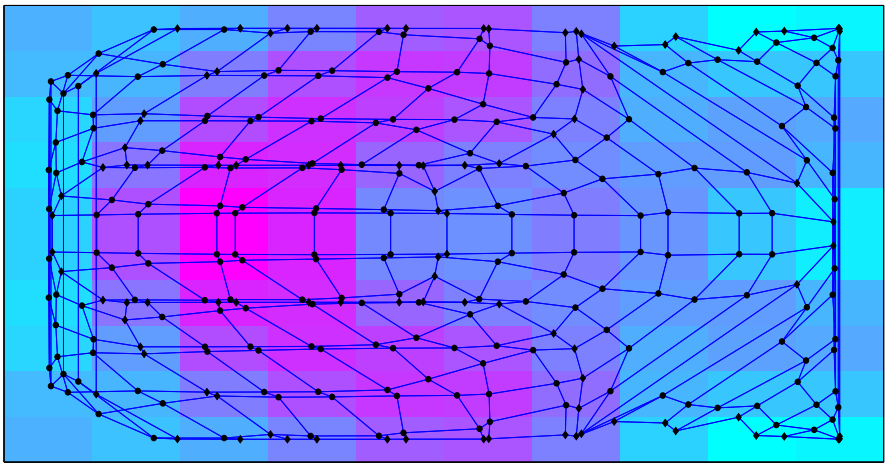


Figure 2.10: Open box embedded by GTM with points superimposed upon the magnification factors. These magnification factors indicate that the areas well preserved in visualisation space (the box bottom and connecting side) have been magnified more than the corners and lid indicating the trustworthiness of the mapping.

shrunk to accomodate the data. These magnification factors indicate that the areas well preserved in visualisation space (the box bottom and connecting side) have been magnified more than the corners and lid indicating the trustworthiness of the mapping in these regions.

GTM offers an interesting alternative to data visualisation when compared with the methods described above but there are certain drawbacks:

- The noise model of isotropic Gaussians is not a realistic situation due to the geometry of Gaussians in high dimensions. This will be more thoroughly explained in chapter 3.
- The rectangular grid is an unrealistic latent space and is limited to a 1 or 2-dimensional visualisation.
- The number of kernels for interpolation of data is limited to be at maximum the size of the latent grid. This ensures the learning phase is quick and not relatively complex or highly parameterised whereas the ideal situation would allow N kernels such as in NS.

On reflection the weaknesses of GTM are shared largely with all generative methods of

visualisation. In order to assume a generative model, restrictions must be placed upon the observations, latent space and mapping functions. These restrictions can often be too restrictive for real world observations. In order to preserve observations in a topographic way, the latent grid should be as large as is possible whilst keeping the number of basis functions low, to avoid overfitting in the regression framework ($\mathbf{m}(\mathbf{x}, W)$). This should circumvent the short circuiting in the training phase where two points close in observation space sit directly atop of one another in latent space.

2.4.2 Gaussian Process Latent Variable Model

The Gaussian Process Latent Vatiable Model (GPLVM) is a probabilistic model using a latent space similar to that of GTM. The two main differences between GTM and GPLVM are:

1. The mapping function from latent to observation space is restricted to a Gaussian Process (GP).

2. The latent space is no longer restricted to a lattice of delta functions.

A short introduction to GPs in the context of the GPLVM will now be given; a thorough introduction is given in [2] and [44]. For observations $Y \in \mathbb{R}^M$: $y_i = f(\mathbf{x}_i) + \varepsilon_i$. GP outputs are scalar by nature, but some methods for extending to 'multiple output GPs' (vector outputs) exist (for example [45],[46] and [13]). The GP used by the GPLVM uses a much simpler notion to create vector outputs, demanding that output dimensions are independent using separate mapping functions: $y_i^m = f_m(\mathbf{x}_i) + \varepsilon_i^m$. In the GP formulation $p(\varepsilon_i) = \mathcal{N}(\varepsilon_i|0, \beta^{-1})$. The GPLVM specifies an independent prior over the latent space, X, such that: $p(X) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{x}_i|\mathbf{0}, I)$. The likelihood $p(Y|X)$ is assumed to be zero mean in general and can be written as:

$$p(Y|X) = \prod_{m=1}^{M} p(\mathbf{y}_m|X) = \prod_{m=1}^{M} \mathcal{N}(\mathbf{y}_m|\mathbf{0}, \mathcal{K}^{NN} + \beta^{-1} I), \tag{2.17}$$

where $\mathbf{y}_m$ is a column vector containing the N entries from Y for dimension m. $\mathcal{K}^{NN}$ represents an $N \times N$ kernel matrix, the most popular choice for which being the squared

exponential (SE), or Gaussian, kernel:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T W (\mathbf{x}_i - \mathbf{x}_j)\right), \tag{2.18}$$

where σ_f^2 is the process variance and W an automatic relevance detection (ARD) diagonal weight matrix. The ARD matrix learns the dimensions of X which are significant in the mapping process. In the standard regression case where X is observed as well as Y, the parameters are learned using gradient descent in a maximum likelihood (ML) fashion from the likelihood equation (2.17). In the GPLVM case X must be learned as well as the kernel hyperparameters, for which there are two main methods:

1. [8] Iterative optimisation of the kernel hyperparamters based on the ML approach for the current X, then optimising $p(\mathbf{y}|\mathbf{x}, \sigma_f^2, W, \beta)$ with respect to X.

2. [47] In a fully Bayesian framework a variational lower bound is used to optimise the marginal likelihood $p(Y) = \int p(Y|X)p(X)dX$. This integral is in general analytically intractable due to the nonlinear interactions in the kernel functions.

The automatic training of the ARD parameters in W allows for the dimension of X to be larger than two and only the two most relevant dimensions visualised. As with RBFs, GPs with SE kernels are infinitely smooth but are not topographic without imposing a 'back constraint'. This back constraint involves the addition of the Sammon STRESS error function from equation (2.9) to the GP likelihood with a multilayer perceptron (MLP) network used to minimise this error [48]. A formal definition of MLP networks is given in chapter 7, all that is important to note here is that it optimises over the STRESS function with respect to the latent points Y and the observations, X. The use of the MLP and imposition of the STRESS measure ensures that the latent points learned are topographic.

Figure 2.11 shows the GPLVM visualisation of the open box using back constraints to ensure the mapping learned is topographic. The algorithm unfolds the structure from the open top and curves all sides to preserve the topological ordering. The points from the box floor remain relatively uniform in the mapping which is an improvement on the

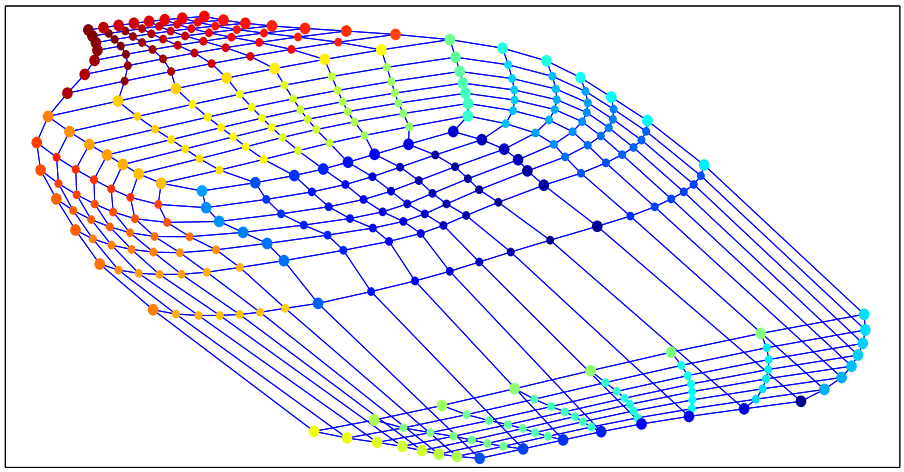


Figure 2.11: Open box embedded by GPLVM with back constraints. It is clear that the algorithm has unfolded the structure from the open top and curves all sides to preserve the topological ordering. The points from the box floor remain relatively uniform in the mapping. The points from the lid appear to have been squashed into a relatively small area, in the top-left of the latent space, compared to that of the sides and bottom, but are still correctly ordered and their relative distance within the lid points is preserved. The box front (cyan) is not well mapped with the vertical dimension of points almost placed directly atop of one another and the entire front face is placed unusually far from the rest of the box.

GTM mapping. The points from the lid appear to have been squashed into a relatively small area, in the top-left of the latent space, compared to that of the sides and bottom, but are still correctly ordered and their relative distance within the lid points is preserved. Each row of points which make up the front face is removed from their local neighbourhours on the other faces of the box. The box front is not well mapped with the vertical dimension of points almost placed directly atop of one another and the entire front face is placed unusually far from the rest of the box. The GPLVM also allows for computation of the posterior probability $P(X|Y)$ which can be superimposed into the visualisation of the box, as shown in figure 2.12. The areas of higher posterior probability are shown in pink; with blue denoting low probabilities. This probability map indicates that the GPLVM has not faithfully interpolated the data space as there are regions of apparent high probability which contain either a low density, or no points. The box lid has the opposite problem of significantly high density of points with a low posterior probability of observation. This is contrary to the fact that the box lid was

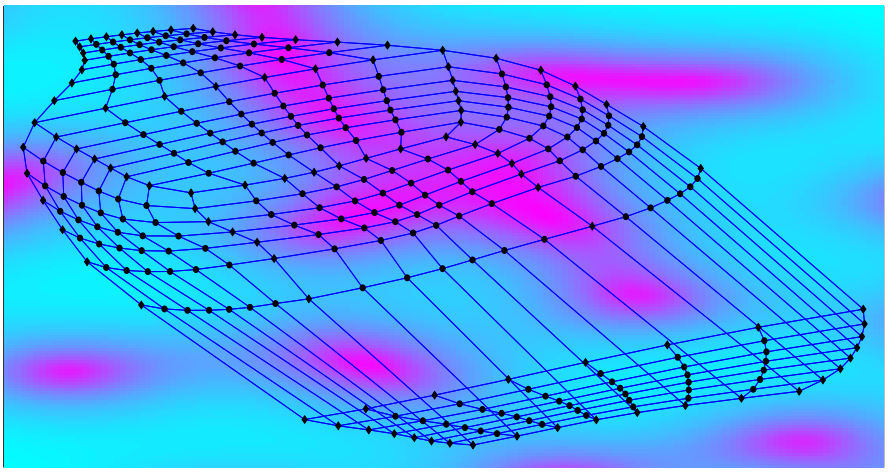


Figure 2.12: Open box embedded by GPLVM with back constraints with posterior probability surface shown. The areas of higher posterior probability are shown in red with blue denoting low probability. This probability map indicates that the GPLVM has not faithfully interpolated the data space as there are regions of apparent high probability which contain either a low density or no points. The box lid has the opposite problem of significantly high density of points with a low posterior probability of observation.

uniformly generated with the same number of points as the other sides.

Compared to GTM this approach is more robust, but it still suffers from restrictions of independence between dimensions of Y and between latent points X. The assumption that all observations, Y, are normally distributed is typically taken to be realistic. The extent to which this is true depends on the application. The concept of the visualisation of posterior probabilities is useful for judging expected location of visualised datapoints. However, even in this simple case of a 3-dimensional embedding the probability surface has been incorrect and misleading. An interesting note about the back constraints imposed here is that the distance measures used for d_x and d_y have not been explicitly specified. The Euclidean distance used in the algorithm in [22] can be deemed appropriate for the latent variables X as the priors over X are isotropic, unit covariance Gaussians. For this case of distribution many dissimilarity measures incorporating uncertainty reduce to the Euclidean distance. On the other hand, the observations Y are learned with specific covariance measures - the main benefit between Gaussian Processes over other Machine Learning tools. Therefore, taking the Euclidean distance as d_y wastes this additional learned information. Other methods for visualisation

incorporating uncertainties such as probabilistic NeuroScale (outlined in Chapter 3) allow for user-specified dissimilarity measures.

2.5 Quality Criterion

The mappings discussed in this chapter all work in different ways and optimise separate objective functions. To this end, it is difficult to assess how well one algorithm performs in generating a visualisation compared to another, particularly since it is easy for visualisations to appear to have structure when there is none [49]. The different open box embeddings show this; some researchers favour the Isomap embedding over the Sammon map. This section will outline some quality criteria which can offer a comparison between mappings based on data ranking. For a more thorough guide see [24].

2.5.1 Rank

The notion of rank, $R(i,j)$, is outlined in [50]. $R(i,j)$ is defined as the number of observations closer to i than j is. Formally this is:

$$R_{data}(i,j) = \left| \left\{ k : D^O_{ik} < D^O_{ij} \right\} \cup \left\{ k : D^O_{ik} = D^O_{ij}, k < j \right\} \right|, \tag{2.19}$$

where U is the union between sets, $|.|$ denotes set cardinality and D^O_{ij} is the dissimilarity between observations i and j. The rank of points i and j in the latent visualisation space is:

$$R_{Latent}(i,j) = \left| \left\{ k : D^L_{ik} < D^L_{ij} \right\} \cup \left\{ k : D^L_{ik} = D^L_{ij}, k < j \right\} \right|, \tag{2.20}$$

where D^L_{ij} is the dissimilarity between latent visualised points. Note that $R_{data}(i,i) = R_{latent}(i,i) = 0$ and $R_{data}(i,j) \neq R_{latent}(i,k)$ even when $D_{ik} = D_{ij}$ but $j \neq k$.

2.5.2 Trustworthiness and Continuity

Two important ways to characterise whether a visualisation is topographic were introduced in [51]. Firstly, an error in visualisation occurs when dissimilar observations

become close in observation space, impairing the trustworthiness (T) of the embedding. The opposite case being where similar observations are made dissimilar in visualisation space, causing a loss in continuity (C).

Using $\mathbb{N}^k_{data}(i)$ and $\mathbb{N}^k_{latent}(i)$ to define the set of the k nearest neighbours of points i in data and latent spaces respectively. In order to define both T and C, neighbourhood intruders and leavers must first be defined. Intruders$_k$(i) is the set of points in the k-neighbourhood of observation (i) in latent space but not in the original observation space:

$$\text{Intruders}_k(i) = \mathbb{N}^k_{latent}(i) \backslash \mathbb{N}^k_{data}(i),$$

where '\' represents the intersection of the relative complement of the set. Consequently the Leavers$_k(i)$ are the set of observations in the k-neighbourhood of (observation) i in the observation space but not in the latent space:

$$\text{Leavers}_k(i) = \mathbb{N}^k_{data}(i) \backslash \mathbb{N}^k_{latent}(i).$$

Trustworthiness can now be defined as:

$$T(k) = 1 - \frac{2}{\Gamma_{TC}} \sum_{i=1}^{N} \sum_{j \in \text{Intruders}_k(i)} \left(R_{data}(i,j) - k\right), \tag{2.21}$$

and Continuity as:

$$C(k) = 1 - \frac{2}{\Gamma_{TC}} \sum_{i=1}^{N} \sum_{j \in \text{Leavers}_k(i)} \left(R_{latent}(i,j) - k\right), \tag{2.22}$$

where:

$$\Gamma_{TC} = \begin{cases} Nk(2N - 3k - 1) & \text{if } k < \frac{N}{2}, \\ N(N-k)(N-k-1) & \text{if } k \geq \frac{N}{2}. \end{cases} \tag{2.23}$$

Better projections are characterised by higher values of T and C indicating less intrusions and extrusions from the neighbourhood. T and C are combined into one

quality measure described in [24] as:

$$Q_{TC}(k) = 2\frac{T(k)C(k)}{T(k)+C(k)}. \tag{2.24}$$

As with the individual T and C measures, a higher value of Q_{TC} indicates a better mapping.

2.5.3 Mean Relative Rank Error

Working similarly to the T and C measures, mean relative rank error (MRRE) [17] compares the ranks of the observation and visualisation spaces:

$$MRRE_{data}(k) = \frac{1}{\Gamma_{MRRE}} \sum_{i=1}^{N} \sum_{j \in \mathbb{N}^k_{data}(i)} \frac{|R_{latent}(i,j) - R_{data}(i,j)|}{R_{data}(i,j)}, \tag{2.25}$$

which corresponds to trustworthiness. On the other hand, continuity relates to $MRRE_{latent}$:

$$MRRE_{latent}(k) = \frac{1}{\Gamma_{MRRE}} \sum_{i=1}^{N} \sum_{j \in \mathbb{N}^k_{latent}(i)} \frac{|R_{data}(i,j) - R_{latent}(i,j)|}{R_{latent}(i,j)}, \tag{2.26}$$

where:

$$\Gamma_{MRRE} = N \sum_{u=1}^{k} \frac{|2u - N - 1|}{u}.$$

Better visualisation quality is indicated by lower values of MRRE. The values are zero when $R_{latent}(i,j) = R_{data}(i,j)$. MRRE weights rank error, unlike T and C. As with Q_{TC} the two MRRE measures are combined in [24]:

$$Q_{MRRE}(k) = 2\frac{(1 - MRRE_{data})(1 - MRRE_{latent})}{(1 - MRRE_{data}) + (1 - MRRE_{latent})}. \tag{2.27}$$

The higher Q_{MRRE} is, the better the embedding.

2.5.4 Local Continuity Meta-Criterion

The continuity measure, C, looks at global continuity, whereas the local continuity meta-criterion (LCMC) [52] measure is interested in strict neighbourhoods:

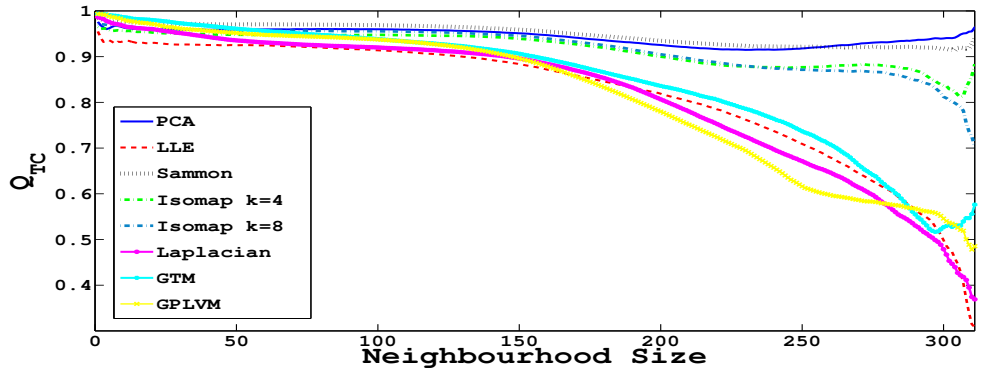
$$LCMC(k) = \frac{1}{Nk}\sum_{i=1}^{N} |\mathbb{N}^k_{data}(i) \cap \mathbb{N}^k_{latent}(i)| - \left(\frac{k}{N-1}\right), \tag{2.28}$$

where $|.|$ denotes set cardinality and '$\cap$' the intersection of two sets. The higher LCMC is, the more truthfully representative the visualisation is. Whereas Q_{TC} is concerned with the number of intruders and leavers of a neighbourhood, $LCMC$ is concerned with what they actually are.

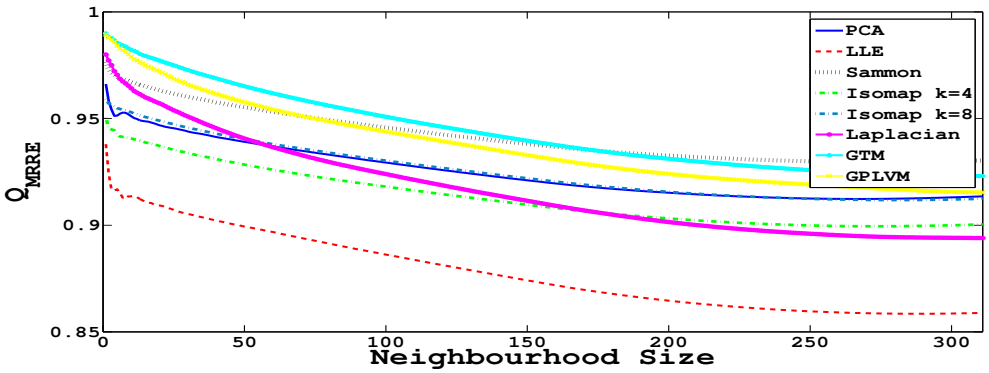
2.5.5 Quality of Open Box embeddings

In this thesis Q_{TC}, Q_{MRRE} and $LCMC$ from equations (2.24), (2.27) and (2.28) will be used to quantitatively assess the performance of the visualisation algorithms used. Figures 2.13a, 2.13b and 2.13c show the Q_{TC}, Q_{MRRE} and LCMC respectively for the Open Box embeddings in this chapter.

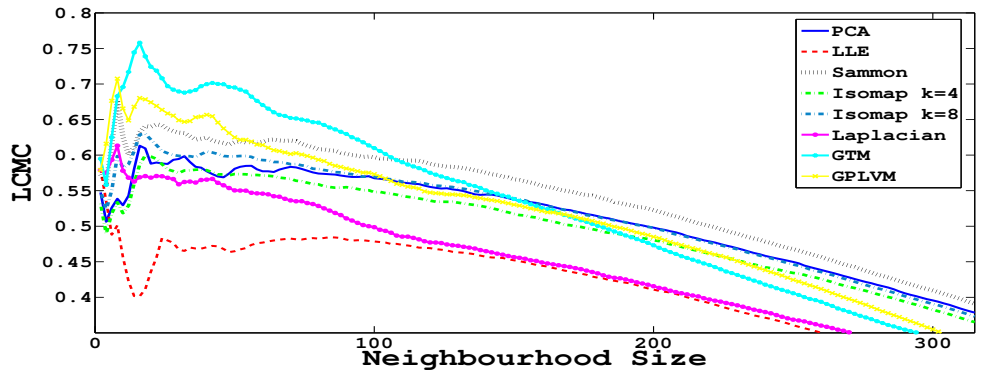
The Q_{TC} for GTM, LE, GPLVM and LLE are similarly grouped, decaying steadily for low neighbourhood sizes and then more rapidly from a neighbourhood of size 150. This is due to the tight clustering of sections of the box, for instance the lid in the GPLVM, corners in the GTM, front face in LE and the overlapping of the sides in the LLE visualisation. These clusters cause high levels of T and C for within cluster neighbourhoods but rapidly decay when outside points are not preserved in relative distances. Isomap with eight neighbours is more trustworthy than with four neighbours for neighbourhoods smaller than 220, but beyond this point the opposite holds. This may be caused by the curvature in the sides and front face of the $k = 8$ mapping, stretching the distances between points compared to the more flat and overlapping sides in the $k = 4$ case. The Sammon map outperforms all topographic mappings for neighbourhoods greater than thirty. The mapping is trustworthy on both a local and global scale, as opposed to the four grouped mappings (GTM, LE, GPLVM and LLE). PCA performs



(a)



(b)



(c)

Figure 2.13: (a) Q_{TC}, (b) Q_{MRRE} and (c) LCMC quality criteria for visualisations of the Open Box dataset. Higher values (1) indicate better performance than lower values (0). (a) It can be seen that all algorithms perform similarly in terms of Q_{TC} up to neighbourhood sizes of 150, following which the Sammon map, Isomap and PCA perform better than the other algorithms. The global trustworthiness of PCA is superior, with LLE performing the worst. (b) GTM and GPLVM perform well in terms of Q_{MRRE} for small to medium neighbourhoods, with Sammon mapping achieving the best global results. LLE performs poorly and the remaining algorithms achieve good performance with a Q_{MRRE} score above 0.9 for the entire structure. (c) GTM and GPLVM preserve local structures well but for neighbourhoods greater than 110 the Sammon mapping preserves local continuity better than the other algorithms. LLE and LE achieve the worst global representations of the Open Box structure.

surprisingly well given the simple underlying generative model. This is largely due to the fact that the visualisation space is a squashed version of the structure which has, in this case, preserved the topological ordering of points, leading to a high measure of Q_{TC}. The MRRE values shown in figure (2.13b) paint a different picture than that of the Q_{TC}. LLE performs the worst out of the methods used, again due to the fact that two of the box sides and floor are essentially superimposed, leading to large differences between data and latent ranks. LE preserves local structures well, but for neighbourhood sizes greater than ten this performance decays rapidly. The overlapping front face and tightly clustered lid are the cause of this. The Isomap embeddings achieve similar results with a steady decay for expanding neighbourhood sizes. The $k = 8$ mapping is superior to that of $k = 4$ where relative rank errors are considered. Both the GTM and GPLVM visualisations behave similarly, outperforming the other algorithms for neighbourhoods of less than 170 and 75 respectively. The rank errors are positively impacted by the clusters used in GTM and the good preservation of topological ordering in the GPLVM mapping. As with the Q_{TC} measure, the Sammon map is best at preserving global structures and rankings. Again PCA performs surprisingly well, comparable with the results of Isomap ($k = 8$) due to the squashed nature of the mapping.

The LCMC values are shown in figure (2.13c). The results are largely similar to those of the Q_{MRRE}. For more local neighbourhoods GTM and GPLVM outperform the Sammon map (110 and 70 respectively) but their performance decays rapidly beyond this point. LLE achieves poor results again, due to the poor preservation of local structures in the overlapping of the box sides. The performance of LE is comparable to that of Isomap initially but quickly decays past a neighbourhood size of eighty. The two Isomap embeddings behave similarly in terms of LCMC but again the $k = 8$ mapping is better than that of $k = 4$. PCA achieves similar results to that of Isomap and the Sammon projection preserves global structure the best.

An additional quality criterion is the STRESS measure as it describes how well relative dissimilarities are preserved. This measure is not as reliable as the three outlined above, since the Sammon mapping optimises over STRESS it is naturally expected to outperform other algorithms. Table 2.1 shows these values for the various mapping

Method	STRESS
PCA	0.0875
LLE	0.9947
Sammon	**0.0810**
Isomap ($k = 4$)	1.7984
Isomap ($k = 8$)	0.4002
LE	0.9986
GTM	0.5035
GPLVM	0.8788

Table 2.1: STRESS measures for Open Box mappings. It is clear that the Sammon map visualisation offers the lowest STRESS, as expected. PCA also performs well with these two measures, almost an order of magnitude better than GTM and Isomap ($k = 8$). The worst performance is seen in the Isomap ($k = 4$) visualisation, likely due to the difference between geodesic and Euclidean distances in the mapping process.

functions on the Open Box dataset. It is interesting to note that Isomap with $k = 8$ is considerably better than with $k = 4$, a difference which is not clear when assessing the quality criteria of figure 2.13. Here LLE outperforms Isomap with $k = 4$ and LE which is surprising given the poorer results of the other quality measures. Naturally the Sammon map outperforms the other visualisations in terms of STRESS. Again we see that PCA performs surprising well on this dataset, but this success is not typically seen in other visualisations.

2.6 Conclusion

The chapter has introduced popular topographic mapping techniques for visualisation of high-dimensional data. The Open Box dataset was used as a comparative standard here, as it was in [24] and [17]. Of the mappings discussed LLE performs poorly, largely because the manifold is uniformly distributed. The algorithm is less effective when observed points are equidistant making neighbourhood reconstruction difficult. LE performs very similarly to LLE in terms of Q_{TC}, has identical gradient structure for Q_{MRRE} and the same LCMC quality beyond neighnourhoods of approximately 120. GTM performs well on this dataset, using performance measures, but the visualisation is very poor; backed up by looking at STRESS in table 2.1. The STRESS values for GTM

are twice that of the Sammon map due to the broken structure in the corners of the manifold.

The quality measures introduced in this chapter are indicative of which mappings achieve good topographic preservation, but not necessarily the best visualisations. For this dataset, PCA performs well, but Isomap is visually more representative of the original manifold. Even using back constraints with GPLVM, the quality of the visualisation generated using the Sammon mapping is globally better.

This example is a toy dataset, but serves the purpose to compare an easily, humanly-interpretable manifold in 3-dimensions, where it can be judged whether or not the visualisations 'makes sense'. The results would likely be different for a cluster-based mapping, favouring LLE and LE, such as the twin peaks or three clusters, used for instance in [15].

The deterministic topographic methods mentioned above will be extended in chapter 3 to incorporate uncertaintes in both observation and visualisation spaces. The GPLVM will be used as a comparison with these extensions in chapters 5 and 6. For reasons which will be made clear in chapter 3, GTM is deemed unsuitable for these types of situations, especially when the concept of high dimensional visualisation is considered.

3 Incorporating Observation Uncertainty into Visualisations

> ‘Geometry is not true, it is advantageous.’
>
> - Henri Poincaré

3.1 Introduction

The purpose of a visualisation space is to provide a representation of observational data such that structure and anomalies can be interpreted. Feed-forward projection methods for visualisation are trained using prototype data generating the ‘expected’ space, i.e. the optimum visualisation space given our current state of knowledge. This chapter introduces and extends the principles of probabilistic visualisation mappings, incorporating current knowledge. The first such method we introduce is Probabilistic NeuroScale [33]. This method attempts to utilise uncertainty in observations in an isotropic sense, generating distributions as opposed to points in the visualisation space. This method will be extended to account for elliptical distributions with non-isotropic

covariance matrices characterising uncertainty. A similar extension is then made for three other popular topographic mapping methods - Locally Linear Embedding, Isomap and Laplacian Eigenmaps.

Constructing a probabilistic visualisation space is more computationally demanding than a pointwise mapping, but there are two significant benefits:

1. It is a more robust measure of dissimilarity between uncertain observations than treating the observations as noise-free datapoints.

2. Visualising data with Uncertainty Surfaces, accounting for expected data positions, provides human users with more information about observations. This is particularly important for real world applications such as SONAR and hospital patient monitoring systems where complex systems offer unnacceptably high false alarm rates, but the nature of the uncertain high dimensional data makes it impossible for a human to analyse the raw data, comparing to 'normal' behaviour. The notion of Uncertainty Surfaces will be defined in chapter 4.

With these benefits in mind, current popular topographic mapping methods will be extended accounting for observation uncertainty. Given the probabilistic nature of the visualisation spaces, the terms visualisation space and latent space will be used as synonyms, in keeping with the terms of GTM and GPLVM.

3.2 Current approaches to uncertainty mappings

Amongst the mappings outlined in chapter 2 the only algorithms which incorporate data uncertainty are GTM and GPLVM. Here, Probabilistic NeuroScale will be outlined before an explanation of the underlying geometric assumption that is made by the two mappings.

3.2.1 Probabilistic NeuroScale

Standard NeuroScale uses an RBF network to minimise the STRESS measure:

$$E = \sum_{i \neq j} \frac{\left(d_x(X_i, X_j) - d_y(Y_i, Y_j)\right)^2}{d_x(X_i, X_j)}, \tag{3.1}$$

where it is assumed each observation, X_i, is subject to an uncertainty σ_i. The observations can be treated as being the mean of a probability distribution with variance σ_i^2. Here each observation is a Gaussian distribution, $\mathcal{N}(X_i, \sigma_i^2)$, and this then maps to a generated Gaussian distribution in the visualisation space. The visualised Gaussian is centred at Y_i with variance given by σ_i^2, $\mathcal{N}(Y_i, \sigma_i^2)$. Observation uncertainties are preserved, allowing decisions based on the mapping to be more informed than those of merely point-wise mappings. This is ensured by altering the STRESS measure of equation (3.1) to incorporate dissimilarities between distributions:

$$E = \sum_{i,j} \frac{\left(KL(\mathcal{N}(X_i, \sigma_i^2), \mathcal{N}(X_j, \sigma_j^2)) - KL(\mathcal{N}(Y_i, \sigma_i^2), \mathcal{N}(Y_j, \sigma_j^2))\right)^2}{KL(\mathcal{N}(X_i, \sigma_i^2), \mathcal{N}(X_j, \sigma_j^2))}, \tag{3.2}$$

where KL in equation (3.2) is the Kullback-Leibler divergence:

$$KL(P_1, P_2) = \int P_1(x) \log\left(\frac{P_1(x)}{P_2(x)}\right) dx,$$

with $P_1(x)$ and $P_1(x)$ two probability distributions. In the case these are isotropic Gaussians as in equation (3.2) is:

$$KL(\mathcal{N}(X_i, \sigma_i^2), \mathcal{N}(X_j, \sigma_j^2)) = \frac{1}{2}\left(\frac{\sigma_i^2 + (X_j - X_i)^T (X_j - X_i)}{\sigma_j^2} - O + O\log(\frac{\sigma_j^2}{\sigma_i^2})\right),$$

where O is the observation dimension. $KL(\mathcal{N}(Y_i, \sigma_i^2), \mathcal{N}(Y_j, \sigma_j^2))$ is of the same form with $O = 2$ for a 2-dimensional visualisation space. It is clear that the dissimilarity is no longer reliant only on the observations, as it is in the Euclidean distances of standard NS. The scaling of the dissimilarities and the '$O\log(\frac{\sigma_j^2}{\sigma_i^2})$' term ensure that the relative uncertainty of each observation is taken into account. Although this is a large improvement on standard mappings, it has a deficiency due to the nature of isotropic Gaussian distributions geometrically. Clarification of this fact will follow.

3.2.2 Geometry of hyperspheres

An isotropic Gaussian is a probability distribution in O dimensions with equiprobable contours. Geometrically this becomes a hyper-sphere, centred at the mean of the Gaussian. Despite the probability distribution having infinite support, we often consider uncertainty as a finite quantity, expressed in terms of one or two standard deviations. This then gives meaning to the radius, r, of the hyper-sphere in terms of uncertainty. In fact, the amount of uncertainty, whether one or two standard deviations, is strictly arbitrary as far as r is concerned. The volume of a hypersphere with radius r is given by [53]:

$$V_{h-s}(r) = \frac{\pi^{\frac{O}{2}} r^O}{\Gamma(1+\frac{O}{2})}, \tag{3.3}$$

where Γ is the Gamma function, $\Gamma(z) = (z-1)!$. As O grows large we see that equation (3.3) tends towards 0. More significantly we can compute the relative volume of a thin hyper-spherical shell:

$$V_{s-shell} = \frac{V_{h-s}(r) - V_{h-s}(r(1-\varepsilon))}{V_{h-s}(r)} = \frac{(1)^O - (1-\varepsilon)^O}{(1)^O}, \tag{3.4}$$

where $\varepsilon \ll 1$. Again as O grows this ratio tends to 1, stating that the thin shell contains the entire volume of the hyper-sphere. In effect, for any arbitrary uncertainty, r, a high dimensional observation is then seen with almost exact certainty on the surface of the imposed hyper-sphere. This physically means that an observation, X_i, with isotropic Gaussian uncertainty, σ_i, occurs with probability approaching 1 on the surface of the hypersphere and with probability approaching 0 elsewhere, including where the actual observation was made. The radius of the hypersphere is dictated by σ_i, typically $2\sigma_i$. Isotropic uncertainties are therefore insufficient to characterise observations where the dimension of observations is high.

This deficiency in high dimensional isotropic Gaussians is problematic not just for the current implementation of Probabilistic NeuroScale but also for GTM. Another popular mapping which generates probabilistic visualisation spaces is the Deep-GP (outlined in appendix B) where the covariance matrices of the high dimensional observations are

restricted to diagonal entries only. In some fields of signal processing the requirement of independence between features is unrealistic, but it is acknowledged that this avoids the deficiencies of isotropic Gaussians, as can be shown by considering the geometry of hyper-ellipsoids.

3.2.3 Geometry of hyper-ellipsoids

In order to circumvent the deficiencies of hyper-spheres, distributions of a hyper-elliptical nature can be used. The equation of a hyper-ellipsoid can be expressed as:

$$(\mathbf{x}-\mathbf{c})A(\mathbf{x}-\mathbf{c}) = 1, \tag{3.5}$$

where $\mathbf{c}$ is the centre of the hyper-ellipsoid and A is a positive semidefinite matrix whose eigenvalues are the lengths of the prinicipal sub-axes of the hyper-ellipsoid. The hyper-ellipsoid can again be thought of as a probability distribution where the elements of A are given by the covariance matrix through the relation $A = \Sigma^{-1}$. The volume of a hyper-ellipse [54] is given by the equation:

$$V_{h-e}(A) = \frac{\pi^{O/2}}{\Gamma(1+\frac{O}{2})}\prod_{i=1}^{O} a_i, \tag{3.6}$$

where a_i is the length of the principal sub-axes for dimension i and Γ is the Gamma function. These can be found through an eigendecomposition of A or these elements in equation (3.6) can be rewritten as: $\prod_{i=1}^{O} a_i = |A| = |\Sigma^{-1}|$. In the same fashion to the calculation above we now explicitly write the formula for a thin elliptical shell:

$$V_{e-shell} = \frac{V_{h-e}(\Sigma^{-1}) - V_{h-e}((\Sigma^{-1}-\varepsilon\mathbf{1}))}{V_{h-e}(\Sigma^{-1})} = \frac{|\Sigma^{-1}| - |\Sigma^{-1}-\varepsilon\mathbf{1}|}{|\Sigma^{-1}|}, \tag{3.7}$$

where $\mathbf{1}$ is a square matrix in O dimensions where all the elements are unity and $\varepsilon \ll 1$. Equation (3.7) can be simplified by using the Sylvester determinant theorem [55] which states that for column vector $\mathbf{c_1}$ and row vector $\mathbf{c_2}$:

$$\det(A+\mathbf{c_2}\mathbf{c_1}) = \det(\mathbf{A})(\mathbf{1}+\mathbf{c_1}\mathbf{A}^{-\mathbf{1}}\mathbf{c_2}), \tag{3.8}$$

where A is any $m \times m$ invertible square matrix and $\mathbf{c_1}$ is a $1 \times m$ vector where the elements are all $-\sqrt{\varepsilon}$ and $\mathbf{c_2} = -\mathbf{c_1^T}$. Equation (3.7) therefore reduces to:

$$\frac{|\Sigma^{-1}|}{|\Sigma^{-1}|} - \frac{|\Sigma^{-1}|(1+\mathbf{c}_1\Sigma\mathbf{c}_2)}{|\Sigma^{-1}|} = 1-(1+\mathbf{c}_1\Sigma\mathbf{c}_2). \tag{3.9}$$

We have the requirement that since Σ is positive semi-definite, and therefore Σ^{-1} is also, that $(-\mathbf{c}_1\Sigma\mathbf{c}_2) \geq 0$. Now since $\varepsilon \ll 1$ and the expression $(-\mathbf{c}_1\Sigma\mathbf{c}_2)$ can be rewritten as: $-\varepsilon\sum_{i,j}\Sigma_{ij}$ which, for finite values of Σ_{ij}, is approximately equal to zero. Equation (3.9) therefore tends to zero for large O.

This shows that in general the volume of a thin hyper-elliptical shell, even in high dimensions, is approximately zero. The difference between the volumes of hyper-ellipsoids and hyper-spheres is seen when the identity matrix is used for Σ in equation (3.7), where the second determinant in the numerator tends quickly to zero. Numerical tests with uniformly random generated covariance matrices, Σ, (i.e. $\Sigma_{ij} \in (0,1)$) in equation (3.7) showed that determinants for dimensions as low as 1000 are numerically infinity (computed by MatLab), forcing the numerator to be 0, contrary to when the identity matrix is used where it is equal to 1. The importance of this result is that all of the probability for a high dimensional isotropic Gaussian distribution will be contained within a thin shell at the surface of the hyper-sphere. This amounts to an observation occuring at the surface of the hyper-sphere with probability approximately 1. This contradicts the traditional reasoning behind using isotropic Gaussians where the most likely location of an observation within the (hyper) sphere is the centre, i.e. the observation itself. On the contrary, the approximately empty shell at the surface of a hyper-ellipsoid is the least-probable location of an observation, as would be expected. This result serves as the motivation for the extensions on uncertainty mappings which follow in this chapter. The isotropic uncertainty measures relying on hyper-spherical geometry are extended to full rank uncertainty matrices which translate to hyper-elliptical probability distributions.

3.3 Elliptical Gaussian Probabilistic NeuroScale - N-NS

The natural extension of the isotropic Gaussian-based Probabilistic NeuroScale described previously, is to a non-isotropic, elliptical O-dimensional observed and P-dimensional visualised Gaussian distribution. For the purpose of visualisation in this chapter it will be assumed that $P = 2$. This will be called Normally-distributed NeuroScale (N-NS) in this thesis. The framework for this model begins with an observation X_i with some uncertainty given by the matrix S_i. In practice the uncertainty of an observation and its covariance matrix are treated as the same. S_i is used here instead of Σ_i since it will most likely be a sample covariance matrix for real world data. The dissimilarity matrix d_x is constructed by computing the dissimilarities between the observations and their uncertainties. In [28] the dissimilarity measure in data space is based solely on the uncertainty matrices corresponding to each observation:

$$d_x(X_i, X_j) = tr[S_j^{-1} S_i] + \log\left(\frac{|S_i|}{|S_j|}\right). \tag{3.10}$$

This measure is motivated by the data where only uncertainties for each observation (S_i) were given. As such the dissimilarity measure specified was the Kullback-Leibler divergence between zero mean multivariate Gaussians. Other measures could have been used and this does not impose a Gaussian distribution over the uncertainties; it merely allows comparison between the covariance matrices in two popular ways. Once the dissimilarity matrix d_x is constructed an initial set of latent distributions with means $\mathbf{y}_i$ and covariance matrices $\bar{S}_i$ are generated. The entries of $\bar{S}_i$ are given by the low rank approximation of the observation uncertainty matrix using singular value decompositions:

$$S_i = U\Lambda V^T \Rightarrow \bar{S}_i = \bar{U}\bar{\Lambda}\bar{V}^T,$$

where $\bar{U}$ consists of the first two left singular vectors from U, likewise for $\bar{V}$. $\bar{\Lambda}$ is the 2×2 upper left square of the Λ matrix, therefore containing the first two singular values in the diagonals. This low rank approximation can be used to reduce the dimensionality of any matrix whilst preserving the matrix structure in a mean-square error (MSE) sense.

An eigendecomposition can be performed, but the SVD is typically more numerically stable. An alternative is to use the information contained within all principal axes of S_i, such that $\bar{S}_i = |S_i|I_2$ with $|S_i|$ the determinant of S_i and I_2 the 2×2 identity matrix. The issue with this latent covariance matrix is that it forces the visualised distributions to be isotropic, when in fact $\bar{S}_i$ from the SVD approach can have prominent elliptical extrusions. The impact of these ellipses will be seen in the visualisations in chapters 5 and 6.

A natural measure for dissimilarity between latent distributions in the visualisation space is the Kullback-Leibler divergence between distributions, and specifically between two Gaussians:

$$KL(\mathcal{N}_i||\mathcal{N}_j) = \frac{1}{2}\left[tr(\bar{S}_j^{-1}\bar{S}_i) + (\mathbf{y}_j - \mathbf{y}_i)^T\bar{S}_j^{-1}(\mathbf{y}_j - \mathbf{y}_i) - K - \log\left(\frac{|\bar{S}_i|}{|\bar{S}_j|}\right)\right],$$

where K is the dimensionality of the visualisation space, in this thesis taken to be two. In the case of the Kullback-Leibler divergence between Gaussians the gradients required for training can be analytically computed as:

$$\frac{\partial E}{\partial \mathbf{y}_i} = \sum_j 2(\frac{d_y}{d_x} - 1)\bar{S}_i^{-1}(\mathbf{y}_i - \mathbf{y}_j), \tag{3.11}$$

With E given by equation (3.2). In this form the weights of the RBF network used for visualisation could be optimised in a gradient descent fashion using:

$$\frac{\partial E}{\partial W} = \frac{\partial E}{\partial \mathbf{y}_i}\frac{\partial \mathbf{y}_i}{\partial W} = \frac{\partial E}{\partial \mathbf{y}_i}\Phi.$$

Alternatively, as with NS, the weights can be learned through the iterative shadow targets scheme from section 2.2.3. This specifies a set of deterministic hidden targets, $\mathbf{t}_i$, in the visualisation space to be learned:

$$\mathbf{t}_i = \mathbf{y}_i + \eta\frac{\partial E}{\partial \mathbf{y}_i}.$$

and updating the weights with $W = \Phi^{\dagger} T$ using T to denote the matrix set of shadow targets. Here however, the targets cannot be deterministic since the latent space consists of a set of Gaussians, $\mathcal{N}(\mathbf{y}_i, \bar{S}_i)$, and as such the shadow targets must account for these probabilistic changes. In the same approach as [33], each shadow target, $\mathbf{t}_i$, can be fit with a full rank Gaussian kernel function:

$$\phi(\mathbf{t}_i|X_i) = ((2\pi)^{\frac{K}{2}}|\bar{S}_i|^{\frac{1}{2}})^{-1} \exp\left[-\frac{1}{2}(\mathbf{t}_i - \mathbf{y}(X_i))^T \bar{S}_i^{-1}(\mathbf{t}_i - \mathbf{y}(X_i))\right]. \tag{3.12}$$

We seek to update the mapping parameters by optimising over the shadow targets to fit the latent distributions appropriately. Differentiation of the log-likelihood over the distribution in visualisation space gives:

$$\begin{aligned} \mathcal{E} &= \sum_i -\log(\phi(\mathbf{t}_i|X_i)), \\ \frac{\partial \mathcal{E}}{\partial \mathbf{y}_i} &= \bar{S}_i^{-1}(\mathbf{y}_i - \mathbf{t}_i), \\ \mathbf{t}_i &= \mathbf{y}_i - \bar{S}_i \frac{\partial \mathcal{E}}{\partial \mathbf{y}_i}. \end{aligned} \tag{3.13}$$

Replacing $\frac{\partial \mathcal{E}}{\partial \mathbf{y}_i}$ from equation (3.13) with that of the STRESS measure in equation (3.11) the shadow targets updating rule is given by:

$$\hat{\mathbf{t}}_i = \mathbf{y}_i - \bar{S}_i \sum_j 2\left(\frac{d_y}{d_x} - 1\right)\bar{S}_i^{-1}(\mathbf{y}_i - \mathbf{y}_j). \tag{3.14}$$

Since the NeuroScale approach utilises RBF networks to create projections to the visualisation space, the visualised means, $\mathbf{y}_i$, are given by:

$$\mathbf{y}_i = \Phi[X_i, C]W,$$

where Φ is a matrix, with dimensions $N \times M$, consisting of nonlinear basis functions over the dissimilarity between X_i and the M centres, C. W comprises the weight matrix where there are $M \times K$ weights. When the network is trained over N O-dimensional observations the matrix T, of dimension $N \times K$, is denoted as the matrix comprising the

N shadow targets, i.e. $T = \{\mathbf{t}_i, i = 1 \ldots N\}$. Due to the linear-output nature of the network, the mean-square error optimum parameters are found using the Moore-Penrose pseudo inverse:

$$\hat{W} = \Phi^{\dagger} T,$$

where $\hat{W}$ represents the updated weight matrix. The iterative updating rule alternates between learning the shadow targets, T, and the new weights, $\hat{W}$. This provides the framework for constructing a projection between observations, X_i, with uncertainties, S_i, to a new set of elliptical Gaussian distributions in the visualisation space.

3.4 Elliptical T-distributed NeuroScale - T-NS

In this section the distributions governing the visualised variables, Y, are modified from being elliptical Gaussians to multivariate T-distributions. T-distributions have been used as part of a visualisation framework in [14], extending the Stochastic Neighbour Embedding to approximate dissimilarities with a T-distribution. Other probabilistic modelling areas use the T-distribution in regression and modelling tasks to account for a finite number of degrees of freedom; distinguishing it from the Gaussian distribution, for instance the extension to GPs; the t-process [56]. The probability density function (PDF) of the T-distribution in K dimensions is given by [57, p. 1]:

$$P(\mathbf{z}|\mu, \Omega, \nu) = \frac{\Gamma(\frac{\nu+K}{2})}{\Gamma(\frac{\nu}{2})\nu^{\frac{K}{2}}\pi^{\frac{K}{2}}|\Omega|^{\frac{1}{2}}} \left[1 + \frac{1}{\nu}(\mathbf{z}-\mu)^T \Omega^{-1}(\mathbf{z}-\mu)\right]^{-\frac{\nu+K}{2}}, \tag{3.15}$$

where the mean of the distribution is μ, the covariance, $\Omega = \frac{\nu}{\nu-2}\bar{S}$, for observation uncertainty S, and degrees of freedom $\nu > 2$. As usual, K is expected to be two for a visualisation space.

Using the above framework, an RBF network can be utilised to map the observations, X, and uncertainties, S, to find the optimum means of the T-distributions, μ, with fixed covariances, Ω. The observational dissimilarity measure, d_x, can be the same as the above described case for N-NS since only the dissimilarity and uncertainty matrices are required for the mapping. Given the distribution from equation (3.15) for the mapping

process, a dissimilarity measure for each of the distributions over the latents, $P(Y_i)$ corresponding to observation X_i, is required. The notation $P(Y_i)$ here is used to ensure the model is flexible enough to use different distributions for each observation. In the case of the multivatiate T-distribution this corresponds to each observation having a unique degree of freedom, ν_i.

Due to there only being a recent interest in applications of the multivariate T-distribution, there has been little work in the literature concerning dissimilarity measures between the distributions. The Shannon entropy of a multivariate T-distribution was computed in [58] as:

$$H(P_i) = -\log\left(\frac{\Gamma(\frac{\nu_i+K}{2})}{(\pi\nu_i)^{\frac{K}{2}}\Gamma(\frac{\nu_i}{2})}\right) + \left(\frac{\nu_i+K}{2}\right)\left[\Psi\left(\frac{\nu_i+K}{2}\right) - \Psi\left(\frac{\nu_i}{2}\right)\right] + \frac{1}{2}\log|\Omega_i|, \tag{3.16}$$

where Ψ is the digamma function. The only aspect of the above expression which changes between distributions i is: $\frac{1}{2}\log|\Omega_i|$, under the realistic assumption that the degrees of freedom do not change between distributions, P_i. This allows for a more efficient calculation of $H(P_i)$ when $\nu_i = \nu\ \forall\ i$, but to allow for flexibility in T-NS the subscript i (and also j) in ν_i are kept when calculating the cross entropy.

An asymptotic approximation to the cross entropy between distributions is given in [59]. It is asymptotic in the sense that as the values ν_i and ν_j grow, the approximation approaches the true cross entropy. In [59] it is shown that for $\nu_i = \nu_j = 5$ the absolute error between the Kullback-Leibler divergence and asymptotic Kullback-Leibler divergence is 0.1 in the 1-dimensional skew-T distribution case. It is therefore assumed that the approximation made by the asymptotic Kullback-Leibler divergence is reliable for the implementation in T-NeuroScale. The asymptotic cross entropy is given as:

$$\begin{aligned} CH(P_i,P_j) \approx\ & \tfrac{1}{2}\log((2\pi)^K|\Omega_j|) + \\ & + \tfrac{1}{2}\left(\tfrac{\nu_j+K}{\nu_j}\right)\left[\left(\tfrac{\nu_i}{\nu_i-2}\right)tr(\Omega_j^{-1}\Omega_i) + (\mu_i-\mu_j)^T\Omega_j^{-1}(\mu_i-\mu_j)\right]. \end{aligned} \tag{3.17}$$

From equations (3.16) and (3.17) $d_y(Y_i, Y_j)$ can be written as:

$$d_y(Y_i, Y_j) = KL(P_i \| P_j) = CH(P_i, P_j) - H(P_i).$$

In order to optimise the weights in NeuroScale, the STRESS given by equation (3.1) must be minimised. Re-writing the error as:

$$E = \sum_{ij} \left(d_x(X_i, X_j) - 2d_y(Y_i, Y_j) + \frac{d_y(Y_i, Y_j)^2}{d_x(X_i, X_j)} \right),$$

and therefore:

$$\frac{\partial E}{\partial d_y} \frac{\partial d_y}{\mathbf{y}_i} = 2 \left(\frac{d_y}{d_x} - 1 \right) \left(\frac{\partial d_y}{\mathbf{y}_i} \right).$$

Differentiating the cross entropy in equation (3.17) with respect to $\mathbf{y}_i$ (after substituting $\mathbf{y}_i$ for μ_i) it is clear that:

$$\frac{\partial CH(P_i, P_j)}{\partial \mathbf{y}_i} = \left(\frac{\nu_j + K}{\nu_j} \right) \Omega_j^{-1} (\mathbf{y}_i - \mathbf{y}_j).$$

Since $H(P_i)$ in equation (3.16) is independent of $\mathbf{y}_i$, $\frac{\partial d_y}{\mathbf{y}_i} = \frac{\partial CH(P_i, P_j)}{\partial \mathbf{y}_i}$. Making use of the linearity in weights of the RBF the gradients can now be evaluated as:

$$\frac{\partial E}{\partial \mathbf{y}_i} = \sum_j 2 \left(\frac{d_y}{d_x} - 1 \right) \left[\left(\frac{\nu_j + K}{\nu_j} \right) \Omega_j^{-1} (\mathbf{y}_i - \mathbf{y}_j) \right], \tag{3.18}$$

$$\frac{\partial \mathbf{y}_i}{\partial W} = \Phi, \tag{3.19}$$

which can be summarised as:

$$\frac{\partial E}{\partial W} = \Phi \frac{\partial E}{\partial \mathbf{y}_i}. \tag{3.20}$$

This expression can then be used in the optimisation of W using gradient descent methods such as steepest descents or scaled conjugate gradients (SCG).

3.4.1 Shadow Targets for T-NS

For the Shadow Targets process a T-distributed kernel is specified:

$$\phi(\mathbf{t}_i|X_i) = P_i(\mathbf{t}_i|\mathbf{y}(X_i),\Omega_i,\nu_i).$$

As with the Gaussian kernel approach of section 3.3, the negative log-likelihood function can be written as:

$$\mathcal{E} = \sum_i -\log(\phi(\mathbf{t}_i|X_i)) = \sum_i \left\{ C + \left(\frac{\nu+K}{2}\right) \log\left[1 + \frac{1}{\nu}(\mathbf{t}_i - \mathbf{y}(X_i))^T \Omega_i^{-1}(\mathbf{t}_i - \mathbf{y}(X_i))\right]\right\},$$

where C represents a set of additive constants independent of $\mathbf{y}(X_i)$ which will disappear during the differentiation step which follows:

$$\frac{\partial \mathcal{E}}{\partial \mathbf{y}_i} = \frac{\left(\frac{\nu+K}{2}\right)\frac{2}{\nu}\Omega_i^{-1}(\mathbf{y}_i - \mathbf{t}_i)}{\left[1 + \frac{1}{\nu}(\mathbf{t}_i - \mathbf{y}_i)^T \Omega_i^{-1}(\mathbf{t}_i - \mathbf{y}_i)\right]}. \tag{3.21}$$

It should be noted that the i subscript has been removed from ν under the assumption that the degrees of freedom do not change, but re-introducing different ν values for each observation does not change the end result. Equation (3.21) can be solved as follows. Replacing $\frac{\partial \mathcal{E}}{\partial \mathbf{y}_i}$ with $\mathbf{u}_i$ in order to simplify notation we have:

$$\mathbf{u}_i = \frac{(\nu+K)\,\Omega_i^{-1}(\mathbf{y}_i - \mathbf{t}_i)}{\left[\nu + (\mathbf{t}_i - \mathbf{y}_i)^T \Omega_i^{-1}(\mathbf{t}_i - \mathbf{y}_i)\right]},$$

through factoring out $\left(\frac{1}{\nu}\right)$ from the denominator.

$$\mathbf{u}_i\left[\nu + (\mathbf{t}_i - \mathbf{y}_i)^T \boldsymbol{\Omega_i}^{-1}(\mathbf{t}_i - \mathbf{y}_i)\right] = ((\nu+K))\,\Omega_i^{-1}(\mathbf{y}_i - \mathbf{t}_i),$$

which is expanded to give:

$$\begin{aligned}&\mathbf{u}_i\left(\tfrac{\nu}{\nu+K}\right) + \mathbf{u}_i\mathbf{t}_i^T\Omega_i^{-1}\mathbf{t}_i\left(\tfrac{1}{\nu+K}\right) - \mathbf{u}_i\left(\tfrac{2}{\nu+K}\right)\mathbf{y}_i^T\Omega_i^{-1}\mathbf{t}_i \\ &+\mathbf{u}_i\mathbf{y}_i^T\Omega_i^{-1}\mathbf{y}_i\left(\tfrac{1}{\nu+K}\right) = \Omega_i^{-1}\mathbf{y}_i - \Omega_i^{-1}\mathbf{t}_i.\end{aligned}$$

Multiplying through by $\mathbf{u}_i^\dagger$ and collecting terms in $\mathbf{t}_i$ gives:

$$\begin{aligned}\mathbf{t}_i^T\Omega_i^{-1}\mathbf{t}_i + \left[(\nu+K)\mathbf{u}_i^\dagger\Omega_i^{-1} - 2\mathbf{y}_i^T\Omega_i^{-1}\right]\mathbf{t}_i \\ + \left[\nu + \mathbf{y}_i^T\Omega_i^{-1}\mathbf{y}_i\right] - (\nu+K)\mathbf{u}_i^\dagger\Omega_i^{-1}\mathbf{y}_i = 0,\end{aligned} \tag{3.22}$$

which is a quadratic in $\mathbf{t}_i$. By completing the square we see that:

$$(\mathbf{t}_i+\Omega_i\mathbf{b})^T\Omega_i^{-1}(\mathbf{t}_i+\Omega_i\mathbf{b}) = \left(\mathbf{b}^T\Omega_i\mathbf{b} - c\right),$$

$$\mathbf{b} = \left[(\nu+K)\mathbf{u}_i^T\Omega_i^{-1} - 2\mathbf{y}_i^T\Omega_i^{-1}\right],$$

$$c = \left[\nu + \mathbf{y}_i^T\Omega_i^{-1}\mathbf{y}_i\right] - (\nu+K)\mathbf{u}_i^T\Omega_i^{-1}\mathbf{y}_i,$$

with $\mathbf{b}$ and c used as auxiliary variables from equation (3.22). Making use of an eigendecomposition of $\Omega_i^{-1} = V\Lambda V^T$ where Λ is a diagonal matrix of eigenvalues and V an orthogonal matrix of eigenvectors:

$$\mathbf{z} = V^T(\mathbf{t}_i+\Omega_i\mathbf{b}),$$

which gives:

$$\mathbf{z}^T\Lambda\mathbf{z} = \left(\mathbf{b}^T\Omega_i\mathbf{b} - c\right). \tag{3.23}$$

Despite there being infinitely many solutions sitting on the circle of equation (3.23) in 2 dimensions, which can be solved for $\mathbf{z} = [z_1, z_2]$, a suitable solution can be found by fixing z_1 then finding z_2 by:

$$z_2^2 = \lambda_2\left[\left(\mathbf{b}^T\Omega_i\mathbf{b} - c\right) - \frac{z_1^2}{\lambda_1}\right],$$

and finally $\mathbf{t}_i$ is given by:

$$\mathbf{t}_i = V\mathbf{z} - \Omega_i\mathbf{b}. \tag{3.24}$$

It is useful to ensure that the value found for z_2 is real by ensuring the z_1 chosen is less than $\left(\mathbf{b}^T\Omega_i\mathbf{b} - c\right)$ by, for example, subtracting a suitable uniform random number.

Once the shadow target value is obtained it can be used to amend the training process. It is apparent that the form of the target in equation (3.24) is far from that of the N-NS case in equation (3.14). Since the target itself has no physical meaning it can be thought of as a pointer for the new location of the latent points. With this in mind the proposed RBF-updating rule is:

$$W = \Phi^{\dagger}\left[Y + \eta(T - Y)\right],$$

where η is the learning rate and $(T - Y)$ gives the difference between the shadow targets and current latent points. This net difference allows for a more incremental but efficient momentum-style updating scheme for the network than the standard gradient descent procedure of equation (3.20). This extension to NS using T-distributions will be called T-NS in this thesis.

3.5 Probabilistic Locally Linear Embedding - PLLE

LLE as introduced in section 2.2.2 is a local method based on pointwise neighbourhood reconstruction. The method assumes local neighbourhoods are Euclidean, in a similar way to other topographic methods such as Riemannian Manifold Learning [15]. The combination of Euclidean dissimilarity measures and neighbourhood reconstructions makes LLE sensitive to noise and uncertainty. As such, LLE can be extended to incorporate uncertain data in a similar way to N-NS.

The first step in the LLE algorithm is to find the neighbourhood structure of each observation on the manifold. Since each observation, in this framework is (assumed) uncertain it can characterised with a PDF, P_i. The probabilistic LLE (PLLE) process requires a dissimilarity measure over distributions. The most popular measure used is the Kullback-Leibler divergence:

$$D_{ij} = KL\left(P_i(X_i) \| P_j(X_j)\right),$$

Naturally other measures for comparing dissimilarities can be used, but the benefit of using the Kullback-Leibler divergence becomes clear when the neighbourhood weight matrix is constructed. The dissimilarity matrix D is then used to find the k-nearest

neighbours of the observation. The original LLE error:

$$E(W) = \sum_i |X_i - \sum_j W_{ij} X_j|^2,$$

can be adapted for distributions as:

$$E(\pi) = \sum_i f\left(P_i(X_i), \sum_j \pi_{ij} P_j(X_j)\right), \tag{3.25}$$

where f is some dissimilarity measure. The weights π_{ij} for the observations X_j not in the k-neighbourhood of X_i are then set to 0. This is equivalent to comparing the distribution $P_i(X_i)$ to a mixture model made up of its neighbours. The Kullback-Leibler divergence between mixture models is analytically intractable, however a variational approximation is described in [60]:

$$KL\left(G^i \| G^j\right) \geq KL_{var} = \sum_a \pi_a^{G^i} \log \frac{\sum_\alpha \pi_\alpha^{G^i} e^{-D_{KL}(G_a^i \| G_\alpha^i)}}{\sum_b \pi_b^{G^j} e^{-D_{KL}(G_a^i \| G_b^j)}}, \tag{3.26}$$

where G^i and G^j are two mixture models with weights $\pi_a^{G^i}$ and $\pi_b^{G^j}$ respectively. An upper bound is also presented in [60] for the Kullback-Leibler divergence measure, and in [61] it is shown that the mean of these bounds is a robust estimator for the true Kullback-Leibler divergence. The reason for not using the upper bound in the PLLE algorithm, other than increased computational complexity, is that the upper bound relies on using a product of Gaussians approximation for the Kullback-Leibler divergence between two Gaussian Mixture Models (GMMs) and does not appear to naturally extend to arbitrary distributions like that of equation (3.26). The only requirement for using the lower bound in equation (3.26) is that one can analytically compute the Kullback-Leibler divergence between individual distributions used in the mixture. A useful trait of KL_{var} above is that the mixtures G^i and G^j are not required to be of the same order, particularly when ε-ball neighbourhoods are used. This allows for incorporation into equation (3.25). The mixture G^i will now be the observation $P_i(X_i)$ with only one mixture component such that $\pi_{G^i} = 1$. The mixture G^j will be made up of the neighbours of $P_i(X_i)$ where the weights are to be determined. The initial PLLE error of equation (3.25) can be re-written

as:

$$E(\pi) = \sum_i KL_{var}\left(P_i(X_i), \sum_j \pi_{ij} P_j(X_j)\right) = \sum_i \left(-\log \sum_j \pi_{ij} e^{-D_{KL}(P_i(X_i)\|P_j(X_j))}\right). \quad (3.27)$$

In chapter 5 the distributions are always Gaussian, i.e. $P_i = P_j = \mathcal{N}$, but this is not a requirement as shown in chapter 6 and as such the subscripts are retained here. In contrast to standard LLE, these weights cannot be learned using an exact process but must be learned by gradient descent. A Lagrange multiplier is introduced to ensure the mixture weights sum to unity:

$$E(\pi) = \sum_i \left(-\log \sum_j \pi_{ij} e^{-D_{KL}(P_i(X_i)\|P_j(X_j))}\right) + \lambda(\sum_j \pi_{ij} - 1), \quad (3.28)$$

following which the gradients are given by:

$$\frac{\partial}{\partial \pi_{ij}} \sum_i \left(-\log \sum_j \pi_{ij} e^{-D_{KL}(P_i(X_i)\|P_j(X_j))} + \lambda(\sum_j \pi_{ij} - 1)\right),$$

giving:

$$\frac{\partial E(\pi)}{\partial \pi_{ij}} = -\frac{e^{-D_{KL}(P_i(X_i)\|P_j(X_j))}}{\sum_{j'} \pi_{ij'} e^{-D_{KL}(P_i(X_i)\|P_{j'}(X_{j'}))}} + \lambda = 0.$$

As with standard EM of GMMs multiplying this expression by π_{ij} and summing, it is clear that $\lambda = 1$. The gradients for π_{ij} are then given by:

$$\frac{\partial E(\pi)}{\partial \pi_{ij}} = 1 - \frac{e^{-D_{KL}(P_i(X_i)\|P_j(X_j))}}{\sum_{j'} \pi_{ij'} e^{-D_{KL}(P_i(X_i)\|P_{j'}(X_{j'}))}}. \quad (3.29)$$

Since the divergence terms in (3.29) have already been calculated in finding the neighbourhood structure, the gradients are relatively inexpensive to compute and often require only a small number of iterations to achieve a minimum. An important adaptation to the mapping is that the dissimilarities D_{KL} can be large depending on the numerical order of the covariance matrices. As such the exponential term in equation (3.27) can quickly tend towards zero, causing dependence on only one neighbour, even when k is chosen to be much larger. To avoid this issue the D_{KL} should be scaled so that the

maximum value of $D_{KL}\left(P(X_i)\|P(X_j)\right)$ is 1. Once the weights, π, have been optimised they are fixed and used to construct the embedding. The reconstruction error is given by:

$$E(Y) = \sum_i KL_{var}\left(Q(Y_i), \sum_j \pi_{ij} Q(Y_j)\right), \tag{3.30}$$

where Q is the embedding distribution. The different notation here is to show that P and Q need not be the same probability distributions, particularly in the case of dimension reduction where the dimensionality of the distribution plays a significant role. As in the case of N-NS the distribution of each embedding Y_i can be restricted to be a Normal distribution with covariance resembling that of the original observation.
Similarly to the weight parameters, minimisation of this error function is not as well-posed a problem as that of standard LLE, hence gradient based optimisation is again required. The gradients with respect to the embedding error in equation (3.30) are given by:

$$\frac{\partial E(\mathbf{Y})}{\partial \mathbf{Y}_i} = \frac{\sum_j \pi_{ij} e^{-D_{KL}(\mathcal{N}(\mathbf{y}_i,\bar{S}_i),\mathcal{N}(\mathbf{y}_j,\bar{S}_j))}(\mathbf{y}_i - \mathbf{y}_j)\bar{S}_j^{-1}}{\sum_j \pi_{ij} e^{-D_{KL}(\mathcal{N}(\mathbf{y}_i,\bar{S}_i),\mathcal{N}(\mathbf{y}_j,\bar{S}_j))}} + U_i, \tag{3.31}$$

with U_i given by the contribution made by the distribution $\mathcal{N}(\mathbf{y}_i,\bar{S}_i)$ in the mapping of $\mathcal{N}(\mathbf{y}_j,\bar{S}_j)$:

$$U_i = \frac{\sum_{j\neq i} \pi_{ji} e^{-D_{KL}(\mathcal{N}(\mathbf{y}_j,\bar{S}_j),\mathcal{N}(\mathbf{y}_i,\bar{S}_i))}(\mathbf{y}_j - \mathbf{y}_i)\bar{S}_i^{-1}}{\sum_j \pi_{ji} e^{-D_{KL}(\mathcal{N}(\mathbf{y}_j,\bar{S}_j),\mathcal{N}(\mathbf{y}_i,\bar{S}_i))}},$$

- i.e. U_i is greater than 0 when Y_i is in the k-nearest neighbours of Y_j. These gradients can be solved by any standard nonlinear optimiser, for instance quasi-Newton or Scaled Conjugate Gradients. Optimisation time can be reduced when Y is initialised by using standard LLE with weights, $\mathbf{w}_i = \pi_i$. This step is, in general, computationally inexpensive.

The LLE algorithm was extended in [62] to incorporate out of sample data by using eigenfunctions to create a kernel in an empirical Hilbert space. However, that extension did not incorporate uncertainties in observations. It also did not allow the imposition of arbitrary distributions in the visualisation space as PLLE does. Despite the changes to the mapping procedure, PLLE still has the main purpose of LLE at heart. LLE looks at

the local weighting of observed neighbourhoods, as does PLLE with distributions P_i. These observations are then reconstructed in a visualisation space using the same neighbourhood weighting scheme in both cases.

3.6 Probabilistic Isomap - PIso

In contrast to the case of LLE, the incorporation of uncertainty into the Isomap framework is straightforward given the previous work. The dissimilarity between observations and their k-nearest neighbours can be given by the Kullback-Leibler divergence. The remaining elements of D can be found using Djikstra's algorithm, providing a connected graph is achieved, similar to the approach of Isomap outlined in section 2.3.1.

The standard Isomap algorithm seeks a topographic mapping by preserving the inner products of observations via MDS. In order to preserve the dissimilarity between distributions in the visualisation space, MDS no longer suffices in creating a topographic map as it does not account for observation uncertainty. In order to preserve these dissimilarities a pseudo-STRESS measure must be used:

$$E_{PIso} = \sum_{i,j} (d_x(i,j) - d_y(i,j))^2 . \tag{3.32}$$

This method essentially reduces to the N-NS case (currently without the RBF network) where the dissimilarity matrix d_x is given by geodesic dissimilarities as opposed to the Kullback-Leibler divergence over all observations. This is seen in the gradients since:

$$\frac{\partial E_{PIso}}{\partial \mathbf{y}_i} = \sum_j 2(d_x - d_y)\bar{S}_i^{-1}(\mathbf{y}_i - \mathbf{y}_j).$$

PIso can also be viewed as a probabilistic extension to the Geodesic Nonlinear Mapping (GNLM) [10] with the un-normalised STRESS error function.

3.7 Probabilistic extension to Laplacian Eigenmaps - PWNM

Extending Laplacian Eigenmaps into a probabilistic framework is again different to the extension of LLE and Isomap. Interestingly, as will become clear, the method no longer relies on the graph Laplacian and is not created using eigen-mapping. It does, however have the same initial dissimilarity and weighting process as well as a similar error function to be minimised. This ensures the ability to learn nonlinear manifolds using weighted neighbourhoods is preserved. Due to these changes to the algorithm the new method will be called Probabilistic Weighted Neighbourhood Mapping (PWNM).

Firstly, a dissimilarity measure operating on distributions must be specified, following which the weight matrix, W, is found; as described in section 2.3.2. For the case that $P(X_i) = \mathcal{N}(X_i, S_i)$, the Kullback-Leibler divergence is an appropriate choice for constructing D. Following this stage only the k lowest elements of D_i are kept and the rest set to zero. The error function minimised by the standard LE algorithm is:

$$E_{LE} = \tfrac{1}{2} \Sigma_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 W_{ij}.$$

Similarly to PLLE this Euclidean Norm can be replaced with another measure. Again here the Kullback-Leibler divergence can be utilised. The error function can be re-written as:

$$E_{PWNM} = \frac{1}{2} \sum_{ij} KL\left(Q_i(Y_i) \| Q_j(Y_j)\right) W_{ij}. \tag{3.33}$$

As with PLLE, this function is no longer solvable via eigendecomposition, therefore gradient based optimisation is used. In the same fashion as PLLE the gradients are given by:

$$\frac{\partial E_{PWNM}}{\partial Y_i} = \frac{1}{2} \sum_j \left(\mathbf{y}_i - \mathbf{y}_j\right) \bar{S}_j^{-1} W_{ij} + U_i, \tag{3.34}$$

where again U_i accounts for the contributions of $\mathbf{y}_i$ in the error of $\mathbf{y}_j$ where W_{ij} is greater than zero:

$$U_i = \frac{1}{2} \sum_j \left(\mathbf{y}_i - \mathbf{y}_j\right) \bar{S}_i^{-1} W_{ji}.$$

As with PLLE, the gradient descent procedure in PWNM can be improved by initialising

Y with standard LE once W has been fixed in the probabilistic framework.

3.8 Overview

In this chapter it has been demonstrated that isotropic uncertainties are insufficient to geometrically capture the nature of high dimensional data. The benefit of utilising elliptical uncertainties, namely full rank covariance matrices to characterise the shape in high dimensions, has been geometrically shown.

The original Probabilistic NeuroScale was extended to account for full rank, elliptical Gaussian distributions. The Shadow Targets updating rule was reformulated in the Gaussian visualisation representation to allow for efficient updating of the RBF network in training. This method was also extended to the multivariate T-distribution where an approximation to the Kullback-Leibler divergence was described allowing for a projection to T-distributed visualisation spaces. It should be noted that despite the degree of freedom, 'ν', essentially being a free parameter, the results in the parametric extention of T-SNE in [63] show that results are better for models where ν is specified and not learned. As a simple rule of thumb the degrees of freedom in the model could be set either to the observation dimension, O, or the number of parameters which make up the means; namely $(N \times P)$. It should be noted that the purpose of this thesis is to describe T-NS and not to define the optimum ν, which should be application specific. Table 3.1 outlines the standard, deterministic cost functions for the algorithms outlined in chapter 2 compared to the proposed extensions from this chapter.

In the next chapter the importance of the underlying distributions generated in the latent visualisation spaces will be explored. PLLE, PIso and PWNM will be incorporated into a generalised RBF framework allowing for interpolation to new observations. Chapters 5 and 6 will show the results of these new algorithms based on vectorial and time series observations respectively.

Standard	Cost Function	Proposed	Cost Function
NS	$\frac{\left(d_x(X_i,X_j)-d_y(Y_i,Y_j)\right)^2}{d_x(X_i,X_j)}$	N-NS	$\frac{\left(d_x(X_i,X_j)-KL(\mathcal{N}(Y_i,\bar{S}_i),\mathcal{N}(Y_j,\bar{S}_j))\right)^2}{d_x(X_i,X_j)}$
NS	$\frac{\left(d_x(X_i,X_j)-d_y(Y_i,Y_j)\right)^2}{d_x(X_i,X_j)}$	T-NS	$\frac{\left(d_x(X_i,X_j)-KL(P(Y_i,\Omega_i,\nu_i),P(Y_j,\Omega_j,\nu_j))\right)^2}{d_x(X_i,X_j)}$
LLE	$\sum_i \lvert\mathbf{y}_i-\sum_{j\in N(i)} W_{ij}\mathbf{y}_j\rvert^2$	PLLE	$\sum_i KL_{var}\left(\mathcal{N}(Y_i,\bar{S}_i),\sum_j \pi_{ij}\mathcal{N}(Y_j,\bar{S}_j)\right)$
Isomap	$\left(d_x(X_i,X_j)-d_y(Y_i,Y_j)\right)^2$	PIso	$\left(d_x(X_i,X_j)-KL(\mathcal{N}(Y_i,\bar{S}_i),\mathcal{N}(Y_j,\bar{S}_j))\right)^2$
LE	$\frac{1}{2}\sum_{i,j=1}^{N}\lVert\mathbf{y}_i-\mathbf{y}_j\rVert_2^2 W_{ij}$	PWNM	$\frac{1}{2}\sum_{ij} KL\left(\mathcal{N}(Y_i,\bar{S}_i)\lVert\mathcal{N}(Y_j,\bar{S}_j)\right)W_{ij}$

Table 3.1: This table summarises the deterministic and proposed probabilistic cost functions for the algorithms outlined in this chapter. All five new algorithms are flexible as to the dissimilarity used in observation space. The links between N-NS, T-NS and PIso are clear. There is also a link between PLLE and PWNM, as there is between LLE and LE.

4 Interpreting Uncertainties In Visualisations

> ‘As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.’
>
> - Albert Einstein

4.1 Introduction

Naturally, given a probabilistic latent space, it is desirable to not only convey a set of points in the visualisation space that describe the observations, but a measure of the uncertainty of these observations which can be interpreted. In this chapter a standard method for how the visualisation space should be analysed incorporating both observation and mapping uncertainties is explored. Observation uncertainties corresponding to the training data (or network centres) in, for instance, N-NS can be visualised. These uncertainties are 2-dimensional distributions and can be combined to

form an 'Uncertainty Surface'. This chapter will formally define the notion of the Uncertainty Surface, upon which the visualised points sit. Following this a method for quantifying the uncertainty generated by the curvature of the mapping function will be outlined, using Fisher Information. This improves the ability of mappings to detect anomalies in the projections. Finally, to use these methods of uncertainty, three of the new probabilistic mappings outlined in Chapter 3 (PLLE, PIso and PWNM) are extended to use RBF networks in order to generate visualisation spaces.

4.2 Uncertainty Surfaces

This section introduces the method used to generate an 'Uncertainty Surface' across the visualisation space. When training the RBF network required for the N-NS and T-NS mappings, a number of centres, either specially selected prototypes or simply multiple randomly chosen representative observations, are used. The network centres characterise the visualisation space and can determine its shape and the way it is interpreted. These centres serve a pivotal role in the nature of the latent representation. For this reason, it is proposed that when the means of the mapped observations are generated in the visualisation space they should sit upon an Uncertainty Surface. This surface is the underlying density of the centres' distribution.

For the case of N-NS, the Uncertainty Surface, $f(Y_{1:M}, \bar{S}_{1:M})$, consists of an M-th order Gaussian Mixture Model (GMM), where M is the number of network centres (or the number of training data observations):

$$f(Y, \bar{S}) = \frac{1}{M} \sum_{l=1}^{M} \mathcal{N}(\mathbf{y}_l, \bar{S}_l), \tag{4.1}$$

or in the case of T-NS:

$$f(Y, \Omega, \nu) = \frac{1}{M} \sum_{l=1}^{M} P_t(\mathbf{y}_l, \Omega_l, \nu_l). \tag{4.2}$$

It has been established that the mappings created by NS vastly improve in quality when as large a number of centres as possible is used in training, i.e. when $M = N$; contrary to normal weight-based models [32]. There are methods for constructing the optimum set

of centres in RBF networks, which has an anology with the prototype selection methods in Dissimilarity Spaces [64]. It was shown in [65] that better classification results could be obtained using a specific subset of data as prototypes. An interesting by-product of the research was that when using less than two hundred prototypes for medium sized datasets (typically 400-1000 observations), random selection of prototypes often achieved better classification results, through a richer feature space, than purpose-chosen prototypes from the popular methods outlined in [65].

4.2.1 Similarities with GTM and GPLVM

In GTM the distribution in the latent space is given by [7]:

$$P(\mathbf{x}) = \frac{1}{K}\sum_{i=1}^{K}\delta(\mathbf{x}-\mathbf{x}_i), \tag{4.3}$$

where the $\mathbf{x}_i$'s are equally spaced on a rectangular grid. The responsibilities of each $\mathbf{x}_i$ form the GTM equivalent to the Uncertainty Surface. The resolution of the equivalent Uncertainty Surface here is governed by the value of K (the size of the latent grid), whereas for N-NS and T-NS, the resolution is governed by the number of centres used in the training of the RBF. Since NS-based methods using RBF networks with the STRESS error do not overtrain, the Uncertainty Surface can appropriately span the full dataset. On the other hand, in GTM it is unlikely that $K = N$ as this limits the interpolation ability of the RBF network, potentially leading to overtraining. The Uncertainty Surface is therefore a more intuitive and useful representation of the observation uncertainties than the latent grid in GTM.

In GPLVM the equivalent plot to the Uncertainty Surface is the posterior probability distribution, $p(X|Y)$. This distribution is determined largely by the fit of the observation likelihood, $p(Y|X)$, which is sensitive to the kernel choice. In addition to this, the posterior is fixed by the prior choice, which in GPLVM is restricted to a product of Gaussians:

$$p(X) = \prod_{i=1}^{N}\mathcal{N}(\mathbf{x}_i|\mathbf{0},I).$$

It is obvious that a product of Gaussians is less flexible than a mixture of Gaussians, of the same order, in the latent space. The GPLVM mapping, like the regression-based RBF network used in GTM, can overtrain. As such the number of training points must be low to avoid this, limiting the interpolation ability of the GP. The GMM distributions used by the five methods discussed in chapter 3 are capable of expressing full rank elliptical covariance structures and not only Isotropic Gaussians. The impact of these two restrictions on the GPLVM are a potential reason why the posterior probability surfaces shown in chapters 5 and 6 are less informative, and sometimes have a negative impact on the interpretability when compared to those of the other methods. In the next section a method for quantifying how informative a projected observation is, when combined with the uncertainty surface, is discussed.

4.3 Mapping Uncertainty

When mapping from observations to the latent space we may wish to convey the of uncertainty imposed upon the visualisation of an observation, X_i. This can be physically explained as the curvature imposed on the mapping process from $X_i \rightarrow Y_i$, or in the probabilistic case $p(X_i) \rightarrow p(Y_i)$. When assessing curvature the natural measure to be used is Fisher Information, as such it will be an integral part of characterising mapping uncertainty. Fisher Information is an approach to measure the degree of order or structure in a system which pre-dates the use of Shannon Entropy [66]. The use of Fisher Information over other entropy-based measures, like Shannon Entropy or the Kullback-Leibler divergence, is not based solely on the curvature, but because it is a local measure [67, p. 35-39] and involves derivatives in its integrand calculation, making it a better descriptor for physical systems than Shannon Entropy [67].
Fisher Information allows for the measurement of the amount of information that an observable random variable, X, carries about an unknown parameter, θ, upon which the probability of X depends. For the 1-dimensional case, the Fisher Information is given by

[2]:

$$I(\theta) = E\left[\left(\frac{\partial}{\partial\theta}\log P(X;\theta)\right)^2 \middle| \theta\right] = \int \left(\frac{\partial}{\partial\theta}\log P(X;\theta)\right)^2 P(X;\theta)dX. \tag{4.4}$$

It can be derived from first principles by considering the information uncertainty relation given by the Cramer-Rao bound. This is outlined in Appendix D. The univariate expression above has been extended to the multiparameter case where $I(\theta)$ is now a Fisher Information Matrix (FIM) with the elements $(I(\theta))_{ij}$ given by:

$$(I(\theta))_{ij} = E\left[\left(\frac{\partial}{\partial\theta_i}\log P(X;\theta)\right)\left(\frac{\partial}{\partial\theta_j}\log P(X;\theta)\right) \middle| \theta\right]. \tag{4.5}$$

Fisher Information was used extensively in [67] as a fundamental methodology to unify physics and to re-derive the fundamental equations of quantum and statistical physics. In particular these results rest on the establishment of the Cramer-Rao bound [68]. The bound states that the variance, or covariance matrix, of an unbiased estimator is given by the inverse of the Fisher Information (or FIM). For an unbiased estimator of θ, $T(X)$, we have that:

$$cov_\theta(T(X)) \geq I(\theta)^{-1}. \tag{4.6}$$

This provides a minimum uncertainty bound which can be implemented in the visualisation mapping. To this end, this added level of uncertainty is automatically integrated into the mapping for interpretation. The Uncertainty Surface, onto which projected means are plotted, is used only to convey observation uncertainty. The FIM for the multivariate Normal distribution is well known to be given by:

$$(I(\theta))_{ij} = \frac{\partial\mu^T}{\partial\theta_i}\Sigma^{-1}\frac{\partial\mu}{\partial\theta_j}, \tag{4.7}$$

where $\Sigma(\theta) = \Sigma$ is independent of the mapping parameters, θ. In N-NS the mean of each Normal distribution in the visualisation space is determined by an RBF network mapping:

$$\mu = \boldsymbol{\phi}W, \tag{4.8}$$

where the parameters of the mapping are the weights contained in the matrix W. The covariance, for example Σ in equation (4.7), is the covariance matrix associated with the visualised mean, $\mathbf{y}_i$, and therefore given by the rank-reduced covariance matrix S_i corresponding to observation X_i. As seen in Appendix A, $\boldsymbol{\phi}_i$ is given by:

$$\boldsymbol{\phi}_i = \phi(d(X_i, C)),$$

such that $\boldsymbol{\phi}$ is an $1 \times M$ vector which is a nonlinear function, ϕ, over the dissimilarities, $d(X_i, C)$. The derivative of μ_i, given in equation (4.3), with respect to the parameters W, corresponding to observation X_i, from equation (4.7) is given by:

$$\frac{\partial \mu_i}{\partial W} = \begin{bmatrix} \boldsymbol{\phi}_i & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\phi}_i \end{bmatrix}, \tag{4.9}$$

where $\mathbf{0}$ is a $1 \times M$ vector of 0's. The matrix in equation (4.9) is of dimensions $2 \times 2M$ for a 2-dimensional visualisation space. Using this expression in equation (4.7) the FIM for observation i, $I(\theta)_i$, can be written as:

$$I(\theta)_i = \begin{bmatrix} \boldsymbol{\phi}_i & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\phi}_i \end{bmatrix}^T \bar{S}_i^{-1} \begin{bmatrix} \boldsymbol{\phi}_i & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\phi}_i \end{bmatrix}. \tag{4.10}$$

The FIM is therefore formed of outer products of the $\boldsymbol{\phi}$ vectors, with dimensions $2M \times 2M$. For the special isotropic case, where $\bar{S}_i$ is given by a scalar, $I_2\sigma_i^2$, $I(\theta)_i$ becomes a block diagonal matrix:

$$I(\theta)_i = \begin{bmatrix} (\frac{1}{\sigma_i^2})\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i & \mathbf{O} \\ \mathbf{O} & (\frac{1}{\sigma_i^2})\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i \end{bmatrix}, \tag{4.11}$$

where $\mathbf{O}$ is an $M \times M$ matrix with all entries 0. In order to characterise mapping uncertainty, a scalar measure is required which describes the FIM of equation (4.10), or (4.11) for the simplified case. This observation-specific quantity is denoted FI_i.

For the elements of the FIM here there are no other free parameters since network centres and therefore input dissimilarities are fixed. D-Optimality [69] is a popular

criterion in optimal design utilising the determinant:

$$FI_i = |(I(\theta)_i)|,$$

with model fit improving by maximising the determinant. In practice it has been found that the determinant of the FIM for uncertainty visualisation mappings is always zero which offers no insight to the relative uncertainty of mapped observations. The reason for this is that $I(\theta)$ is formed by the outer product of the $\boldsymbol{\phi}_i$ vectors. An outer product matrix only has rank unity since for any arbitrary vector $\mathbf{z}$:

$$(\boldsymbol{\phi}_i \boldsymbol{\phi}_i^T)\mathbf{z} = \boldsymbol{\phi}_i(\boldsymbol{\phi}_i^T \mathbf{z}),$$

which is the scalar $(\boldsymbol{\phi}_i^T \mathbf{z})$ multiplied by $\boldsymbol{\phi}_i$. A rank one matrix has only one non-zero eigenvalue and using the fact that:

$$|(A)| = \prod_i \lambda_i,$$

where λ_i is the i-th eigenvalue of a matrix A, it is clear the zero eigenvalues will force the determinant to be zero.

It is therefore preferred to use the A-optimality [69] criterion:

$$FI_i = tr\left(I(\theta)_i^{-1}\right), \tag{4.12}$$

a suitable lower bound for equation (4.6). As above, the matrix $I(\theta)$ is rank one and therefore not numerically stable for inversion. It is proposed that the pseudo-inverse, $I(\theta)^{\dagger}$, can be implemented to circumvent this issue. An alternative is to use a jitter-based inversion, but the resulting matrix will depend heavily on the level of jitter and as such the resulting values of FI_i could be unreliable. In the simplified case, where the observation uncertainty is given by a scalar, the block diagonal nature of the FIM allows this computation to be made more efficient. The inversion of a block-diagonal matrix FIM, where the off-diagonal blocks are matrices consisting only of 0's, is given by:

$$I(\theta)_i = \begin{bmatrix} \left(\left(\frac{1}{\sigma_i^2}\right)\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i\right)^{-1} & \mathbf{O} \\ \mathbf{O} & \left(\left(\frac{1}{\sigma_i^2}\right)\boldsymbol{\phi}_i^T\boldsymbol{\phi}_i\right)^{-1} \end{bmatrix},$$

and as such only one of the diagonal blocks requires inversion. For a 2-dimensional visualisation space FI_i is given by:

$$FI_i = 2tr\left(\bar{I}(\theta)_i^{\dagger}\right),$$

where $\bar{I}(\theta)$ is the block matrix.

On a practical note, this measure is an absolute quantity of mapping uncertainty. Due to the nature of mappings constructed to preserve relative dissimilarities, it is proposed that a relative measure of A-optimality is used. A softmax modification of this uncertainty measure or a simple scaling to maximum unity allows the relative uncertainties to be interpreted, offering a more useful anomaly detection-based quantity. In this thesis the scaling to unity approach has been used to avoid over-penalising lower values. For functions, ϕ, monotonically increasing on $d(X_i, C)$ (for example splines), the values of $\boldsymbol{\phi}$ will be larger for more dissimilar observations, X_i and therefore the trace of the inverse, used in the calculation of FI_i, should be lower than for expected observations. The converse happens for monotonically decreasing functions, ϕ, for example the squared exponential function. Anomalies are however often found to have a higher observed variance than expected data. As such, large levels of mapping uncertainty are indicated by larger values of FI_i. An important addendum to this is the case where the uncertainty, or variance, of observations is shared over all data; as is seen in chapter 7. In this special, but not uncommon, case lower values of FI_i indicate anomalies. With this in mind both large and small values of FI_i should be investigated in visualisations. Later, we will use this mapping uncertainty to quantitatively indicate informative 'surprise' of an observation projected into the visualisation space.

4.4 Feed Forward Visualisation Mappings

Currently this use of mapping uncertainty through Fisher Information has been described only in the case of the RBF network used by N-NS for visualisation. In order to apply the mapping uncertainty to PLLE, PWNM and PIso outlined in chapter 3, these models must be extended to the form of an RBF network. Extensions of the standard LLE, LE and Isomap are given in [62] using Kernel methods and Eigenfunctions. There are two main differences between the RBF approach derived here and that of [62]:

1. In this thesis the mapping functions are provided explicitly.
2. Extension to new points in a mapping does not require new eigen-analysis in an RBF framework as the weights are fixed after training.

There are five parts to the RBF framework for visualisation which must be addressed:

1. What is $d(X_i, C_j)$?
2. What is the nonlinear function or functional ϕ?.
3. How are the weights, W, optimised to find Y from the training data minimising the relevant error function?
4. What is $d(X^*, C_j)$, where X^* is a new observation?
5. How is the mapping uncertainty, FI_i, using Fisher Information for an observation X_i characterised?

We address these five points in the following sections.

4.4.1 RBF PLLE

Of the three new probabilistic methods discussed in section 3, PLLE is the most awkward to fit into the RBF framework of:

$$y_{ik} = \sum_j W_{jk} \phi(d(X_i, C_j).$$

It will become obvious that the 'nonlinear basis function' ϕ is somewhat abused to work with PLLE (and PIso) in the following sections.

1. Since PLLE, like LLE, is only concerned with neighbourhoods (k), $d(X_i, C_j)$ can be given by:

$$d_{ij} = d(X_i, C_j) = \begin{cases} KL(X_i \| C_j) & \text{if } C_j \in \mathbb{N}^k(i), \\ 0 & \text{otherwise} \end{cases}$$

2. Once the dissimilarity matrix D is computed the function ϕ is used to fit the weights. This is written as:

$$\phi(d_{ij}) = \begin{cases} 0 & \text{for } d_{ij} = 0, \\ \pi_{ij} & \text{for } d_{ij} \neq 0. \end{cases}$$

where π_{ij} is found by gradient descent of equation (3.5). Following this stage the Φ matrix is fixed for training.

3. The natural process for determining the network weights, W, is found by following the shadow targets process in section 3. The derivatives of the error in equation (3.30) with respect to the visualised parameters Y are given in equation (3.31). Since $Y = \Phi W$ and $\mathbf{t}_i = \mathbf{y}_i - \eta \bar{S}_i^{-1} \frac{\partial E}{\partial \mathbf{y}}$ (as in N-NS the parameters of W are given by $W = \Phi^{\dagger} T$). Gradient descent modification of the learning rate η can be used until convergence of the error with respect to W.

4. The propagation of a new observation through the network is relatively trivial here:

$$Y_k^* = \sum_j W_{jk} \phi(d(X^*, C_j)),$$

where d and ϕ are used as in the training phase and W is the fixed matrix of network weights.

5. The mapping uncertainty given by the Fisher Information Matrix in equations

(4.7) and (4.10) for the PLLE RBF is:

$$I(\theta)_i = \begin{bmatrix} \boldsymbol{\pi}_i & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\pi}_i \end{bmatrix}^T \bar{S}_i^{-1} \begin{bmatrix} \boldsymbol{\pi}_i & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\pi}_i \end{bmatrix},$$

a sparse matrix of the square neighbourhood weights penalised by the observation uncertainty. In the special case where $\bar{S}_i$ is given by the identity matrix, FI_i is bounded from below such that $1 \leq FI_i$. The equality in the constraint occurs when an observation is characterised entirely by one neighbour. The converse, where FI_i is large, occurs when the weights, π_{ij}, for observation X_i are all equal to $\frac{1}{k}$. Both cases can be thought of as anomalous. In reality it is low values of FI_i that are of interest, particularly when $FI_i = 1$. This can occur in one of two scenarios. Firstly, when an observation, both in terms of its mean and uncertainty, are identical to one of the network centres. Secondly, when an observation is far from all network centres and therefore found numerically to be well characterised by only one centre. Of the two scenarios this extrapolation scenario is expected to be far more likely, under the assumption that exact replicas of observations are not expected. It is assumed that observations characterised entirely by one neighbour will be subject to a higher level of uncertainty than other observations within densely populated neighbourhoods.

4.4.2 RBF PWNM

Due to the use of the nonlinear heat kernel function (commonly called the Gaussian function) in Laplacian Eigenmaps the framework is more naturally paired with the RBF than that of PLLE.

1. The neighbourhood graph is constructed with $d(X_i, C_j)$ similar to that of PLLE:

$$d_{ij} = d(X_i, C_j) = \begin{cases} KL(X_i \| C_j), & \text{if } C_j \in \mathbb{N}^k(i), \\ 0, & \text{otherwise.} \end{cases}$$

2. With the dissimilarity matrix, D, ϕ can be implemented as:

$$\phi = \exp\left(-\frac{D}{2\sigma^2}\right), \sigma^2 = (max(D) \times s),$$

where s is the sigma parameter specified in standard LE, $s \in (1, \infty)$. The Φ matrix is then fixed so that W can be learned.

3. Minimising the PWNM error from equation (3.33) can again be performed with Shadow Targets learning. The weight matrix, W, is computed as for PLLE.

4. In feed forward mode the new observation, X^*, is projected to:

$$Y_k^* = \sum_j \exp\left[-\frac{d(X^*, C_j)}{2\sigma^2}\right] W_{jk},$$

where σ is fixed from the first stage.

5. The mapping uncertainty here is similar to that of N-NS given by equation (4.10). In the special case where $\bar{S}_i$ is given by the identity matrix the bound on FI_i is determined by the choice of s. Writing the elements of $\boldsymbol{\phi}$ as $\boldsymbol{\phi}_j = \exp\left(-\frac{d_{ij}}{\max(d_{ij})(2s)}\right)$ so $\boldsymbol{\phi}$ is bounded from below by zero and from above by:

$$0 \leq \frac{d_{ij}}{2\max(d_{ij})} \leq \frac{1}{2} \Rightarrow 0 \leq \phi_j \leq \exp(-\frac{1}{2}s),$$

where s here is a free parameter greater than 0 (with popular choices being 1 or ∞). For large values of s, ϕ_j approaches 1 and $FI_i \Rightarrow \frac{1}{k}$ where k is the neighbourhood size. The information then becomes the same for all observations and therefore uninformative. For fixed d_{ij}, FI_i grows for decreasing s. With this in mind it is preferable to fix s to be low.

4.4.3 RBF PIso

In Isomap and PIso, geodesic dissimilarities are used to construct the neighbourhood graph which must be reflected in the RBF implementation:

1. Finding $d_{ij} = d(X_i, C_j)$ must be performed in a two step process. Firstly, constructing the k-ary neighbourhood graph:

$$D_{ij} = d(X_i, C_j) = \begin{cases} KL(X_i \| C_j), & \text{if } C_j \in \mathbb{N}^k(i), \\ 0, & \text{otherwise.} \end{cases}$$

Secondly, passing D to Dijkstra's algorithm to find the remaining dissimilarities:

$$D_{ij} = d(X_i, C_j) = \begin{cases} KL(X_i \| C_j), & \text{if } C_j \in \mathbb{N}^k(i), \\ dijkstra(X_i, C_j), & \text{otherwise.} \end{cases}$$

2. Following the standard Isomap algorithm the nonlinearity $\phi(d_{ij})$ should square the dissimilarities: $\phi(d_{ij}) = (d_{ij})^2$.

3. Minimising the PIso (also classical MDS) error function of equation (3.32) turns out to be an almost identical shadow targets process to that of N-NS except with the slightly altered error; now optimising over $\sum_{ij} \left(d_x(X_i, X_j) - d_y(Y_i, Y_j)\right)^2$.

4. The projection of a new observation, X^*, to its mean, Y^*, in the visualisation space requires the calculation of its dissimilarity with respect to the network centres. This should again be done in the two step fashion. Depending on the algorithmic implementation, the computation of the Dijkstra dissimilarity may require the squared dissimilarity matrix to find geodesic distances (for instance the implementation in [20]). The matrix should be constructed as follows:

$$D^* = \begin{bmatrix} D & \mathbf{0} \\ \mathbf{d}(X^*, C) & 0 \end{bmatrix},$$

with the geodesics $d(X^*, C_j), j \notin k$ nearest neighbours of X^* computed. Padding the matrix with a column vector, $\mathbf{0}$, prevents the algorithm from recomputing the neighbourhood dissimilarities using the new observation. The mapping process is then given by: $Y^* = d(X^*, C)W$.

5. The computation of the mapping uncertainty for an observation is given by equation (4.10). This is not bounded from above since d_{ij} has no upper bound, as with N-NS, but bounded from below by zero, provided d_{ij} is.

4.4.4 Mapping Uncertainty with T-NS

T-NS introduced in section 3.4 requires a slight alteration to the mapping uncertainty equation (4.7) as the latent distributions are T-distributed and not Normally distributed. As such, the FIM needs to reflect the difference. Firstly it must be noted that $\Omega = \frac{\nu}{\nu-2}\bar{S}$ where $\bar{S}$ is the covariance matrix relating to the observation. The FIM can be calculated from the KL divergence as the second derivative with respect to the parameters:

$$(I(\theta))_{ij} = \lim_{P' \to P} \frac{\partial^2}{\partial \mu_i \partial \mu_j} KL(P \| P'),$$

which for T-distributions, using the KL approximation from equations (3.16) and (3.17) it is given that:

$$(I(\theta))_{ij} = \frac{1}{2}\left(\frac{\nu+2}{\nu}\right)\left(\frac{\nu}{\nu-2}\right)^{-1} \frac{\partial \mu^T}{\partial \theta_i} \bar{S}^{-1} \frac{\partial \mu^T}{\partial \theta_i},$$

which can be summarised as:

$$(I(\theta))_{ij} = \left(\frac{1}{2} - \frac{2}{\nu^2}\right) \frac{\partial \mu^T}{\partial \theta_i} \bar{S}^{-1} \frac{\partial \mu^T}{\partial \theta_i}. \tag{4.13}$$

This can be implemented to describe the mapping uncertainty for T-NS by calculating FI_i; as in equation (4.12).

4.5 Conclusion

This chapter has introduced two methods for representing uncertainty in visualisation spaces. The use of an Uncertainty Surface allows for simple human interpretation as to how likely an observation is, based upon the network centres used in training. It is important to note that visualisation spaces do not behave like classification targets. For certain tasks when a specific anomaly-detection operation is required, specifically

chosen centres are required. An example of this is the work of [27] in the SONAR domain where the anomalies are, for instance, unseen ship and submarine contacts.

The second measure, mapping uncertainty, utilizes Fisher Information to describe how surprising an observation is compared to the network centres. This measure, by construction, incorporates the observation uncertainty. It was found in [70] that more sensitive target detection of observations in N-NS is indicated by higher FI_i than by the corresponding location of a visualised mean on the Uncertainty Surface. This is caused by the imposed curvature of the mapping function between observation and visualisation spaces. It should be noted that in [32] it is shown that the learning process in NS is more robust, in terms of both interpolative accuracy and reduced curvature, when the Shadow Targets algorithm is used than when the latents, Y, are learned and treated as supervised targets in a regression context.

The extension of PLLE, PIso and PWNM to use RBF networks allows for feed-forward mappings of unseen data. These measures of uncertainty will be used to create an informative probabilistic visualisation space of six distinct datasets in chapter 5, for vectorial observations, and chapter 6, for time series observations.

5 Visualisation of Vectorial Observations

> ‘In mathematics you don’t understand things.
>
> You just get used to them.’
>
> - John von Neumann

5.1 Introduction

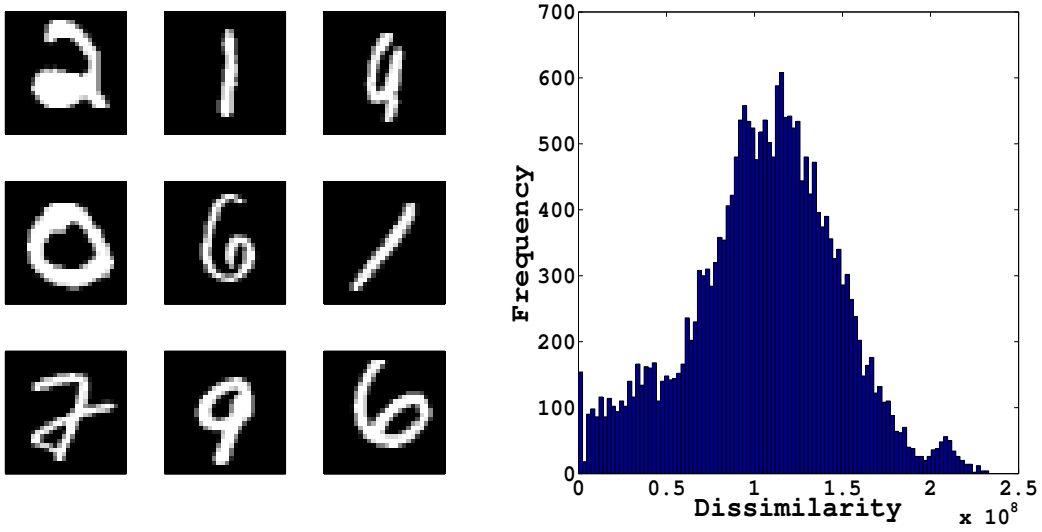
The purpose of this chapter is to generate visualisations of vectorial datasets using the algorithms and uncertainty measures developed in chapters 3 and 4. It will be demonstrated that the extended algorithms can map structures faithfully when creating a dissimilarity space. As a comparison the GPLVM algorithm will be used on the three datasets. However, for this model the points will be treated as deterministic in observation space since GPLVM is incapable of mapping specific uncertainties in its standard form. Following the discussion in chapter 3 on the unsuitability of isotropic Gaussians in representing high dimensional observations GTM is not used as a

comparison in this thesis. The three datasets to be used are:

- the MNist handwritten digits database - an example of real world data with a single shared uncertainty measure [71],
- an artificially generated 4-cluster dataset with cluster specific observation uncertainties,
- an artificially generated punctured sphere dataset where each observation has its own uncertainty.

The visualisation spaces created by each of the five probabilistic methods from chapter 3 (N-NS, T-NS, PLLE, PIso and PWNM) as well as GPLVM will be discussed. The kernel functuion used for GPLVM is a compound kernel combining the commonly used squared exponential kernel, a bias term and additive noise process learned from the data. The quality criterion from section 2.5 will be compared for the five new mappings. GPLVM will not be included in this comparison as it only creates deterministic points in latent space and observes deterministic points in data space. As such, GPLVM is performing a different task to the other mappings. Comparison between GPLVM and the other mappings in any way other than the qualitative visualisations generated is therefore not informative.

In chapters 5 and 6 observations are plotted as circular points, representing the means of the latent distributions. These means sit atop the Uncertainty Surface, f, for that mapping which is evaluated over the visualisation space. f is shown as a heatmap with pink indicating areas of high probability and blue indicating areas of low probability. To indicate the level of mapping uncertainty for an observation, FI_i in the visualisations in chapters 5 and 6, the size of points will be increased to reflect the higher levels of FI_i. These points should be interpreted as being more anomalous than the more certain observations. In order to avoid confusion in chapters 5 and 6 mapping uncertainty will be referred to as ‘mapping surprise’ so that it is not misinterpreted as a reference to the Uncertainty Surface.



(a) Example observations (b) Dissimilarity Histogram

Figure 5.1: (a) Examples of nine images taken from the MNist dataset and (b) histogram of dissimilarities (right). The structure of the characters is clear. However, as seen in the lower-left "7", there are joins and discrepancies in the images. The dissimilarities are spread considering the scale of 10^8 along the x-axis. The distribution over dissimilarities could be approximated by a single truncated Gaussian or Gamma distribution.

5.2 MNist Dataset

The first vectorial dataset used in this thesis is the MNist dataset [71]. The dataset consists of 60,000 training and 10,000 test images of handwritten digits (0 - 9). Figure 5.1a shows nine sample images taken from the dataset. This dataset is traditionally taken as a deterministic pointwise analysis problem to test classifiers. Despite this, there is an intrinsic covariance matrix observable in the data. The images are 28×28 pixels and are therefore treated as a 784-dimensional vector observation. The visualisation spaces in this thesis consist of a very small subset of the data, mapping fifty "0"'s, fifty "1"'s and fifty "6"'s to a 2-dimensional visualisation space.

The uncertainty given by the sample (784-dimensional) covariance matrix is shared across all observations. It should be noted that this is not the typical approach used in the literature and is only used as a proxy to describe the uncertainty, or spread, of observations. Dissimilarity matrices were constructed using the Kullback-Leibler divergence between observations which, due to the shared covariance matrix, reduces to

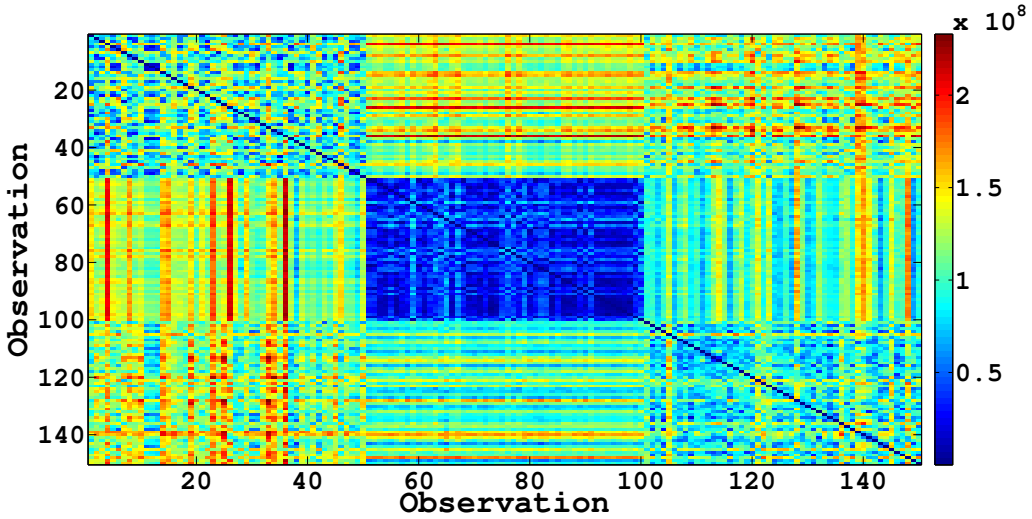


Figure 5.2: Dissimilarity matrix for 150 sample subset of the MNist database incorporating uncertainty. Is is seen that there is a more subtle dissimilarity between "0"'s (observations 1 to 50) and "6"'s (observations 101 to 150) than with "1"'s (observations 51 to 100). The within-class dissimilarities are much lower for "1"'s compared with the other two clusters, most likely since there is often little variation in how "1"'s are drawn compared with the other two characters.

a weighted Euclidean distance:

$$d_{ij} = (\mathbf{x}_i - \mathbf{x}_j)^T \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j). \tag{5.1}$$

A histogram of the dissimilarities is shown in figure 5.1b and the dissimilarity matrix is shown in figure 5.2 showing separation but relative similarities between the three groups. It was established in chapter 2 that the Sammon map produces reliable visualisation spaces in terms of both local and global neighbourhoods. As a benchmark reference, figure 5.3 shows the visualisation generated by a Sammon mapping of the 150 samples to illustrate how these images can be represented in a low-dimensional space. There is a clear separation between the "1"'s (grey) to the "0"'s (white) and the "6"'s (black). However, there is an overlap between the "6"'s and "0"'s due to the circles present in both. There is a greater similarity between "6"'s and "1"'s than "0"'s and "1"'s due to the presence of an unjoined straight line not present in the "0"'s. The equivalent probabilistic maps will now be shown to compare the advantages of using distributions in the latent space.

All of the six visualisations in figures 5.4, using N-NS, T-NS, PLLE, and 5.5, using PIso,

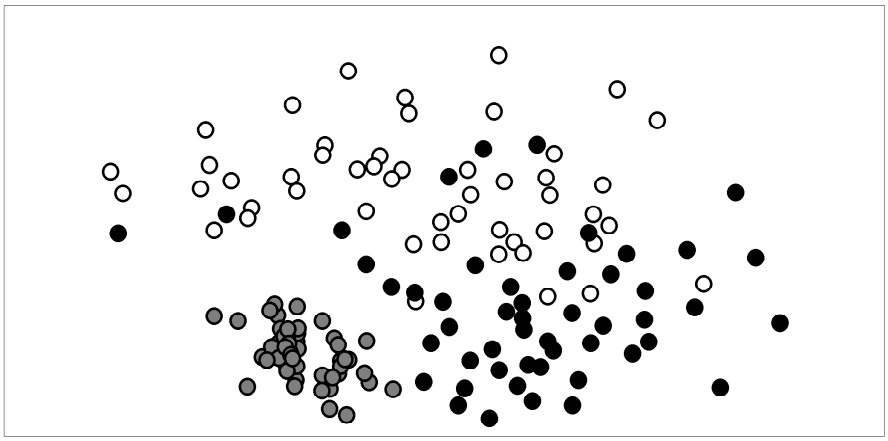
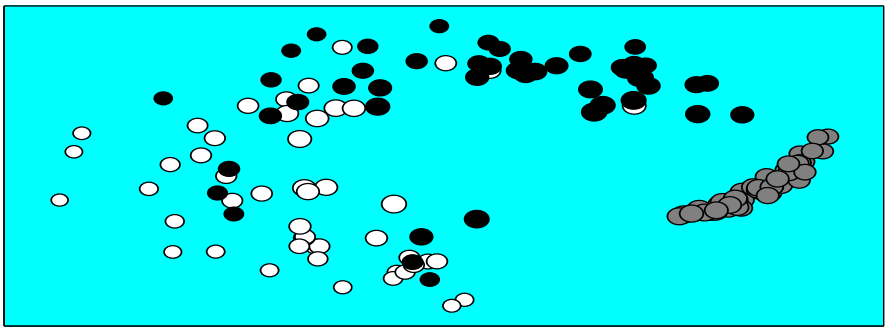
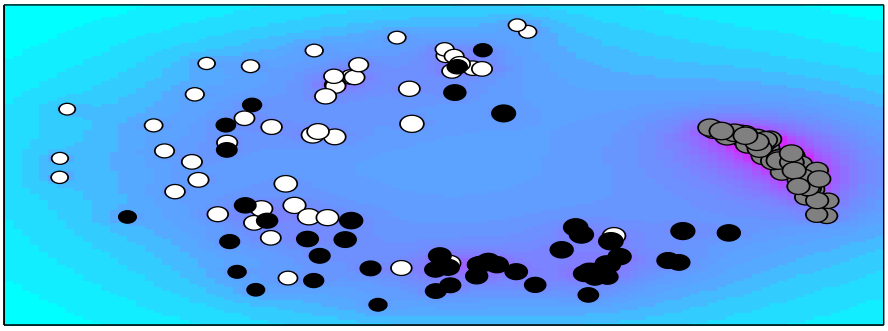


Figure 5.3: Sammon mapping for 150 samples from MNist database accounting for uncertanties in observations. "0"'s are plotted as white, "1"'s as grey and "6"'s as black circles. There is clear separation of the cluster of "1"'s and a level of overlap between the "0"'s and "6"'s due to the similarities between the shapes of the numbers.

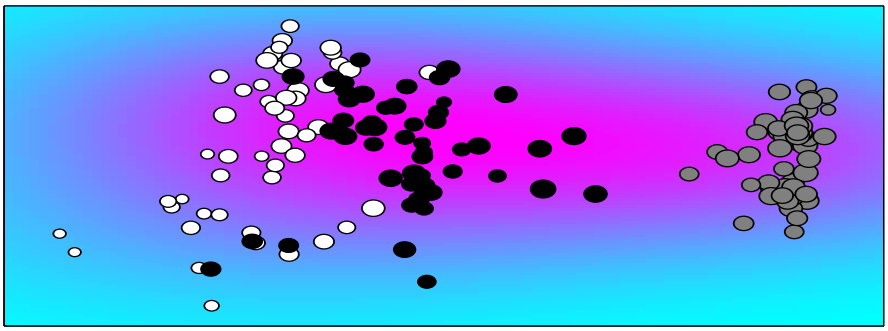
PWNM and GPLVM, distinguish between the three classes, successfully segmenting the "1"'s from the other two clusters. In N-NS and T-NS (figures 5.4a and 5.4b respectively) an identical overall structure is observed where the clusters of "1"'s and "6"'s are closer than the "0"'s and "1"'s. Both latent spaces have the same outliers from the "6"'s contained within the general cluster of "0"'s. The mapping surprise highlights the "1"'s as being the most anomalous group and all other observations occuring within a relatively small range of uncertainties. The PIso visualisation (figure 5.5a) is similar to that of N-NS as it optimises a similar objective function. A connected graph was found for $k = 8$ neighbours so the parameter was fixed to this. The use of geodesics as dissimilarities has created a further separation between the "1"'s and the other two clusters. These two clusters have become more spread but remain overlapping with the most uncertain observations belonging to the class of "6"'s. As with PIso, the PLLE mapping was generated using $k = 8$ neighbours with different consequences. The cluster of "1"'s is much more spread out than in the other mappings and there is a clear separation between the "1"'s and "0"'s thanks to the intermediary cluster of "6"'s. As with the NS-based methods the "1"'s appear to have the highest level of mapping surprise which indicates their location is subject to observation inaccuracies. On the other hand, the outliers from the "0" class in the lower left have a low level of mapping



(a) N-NS

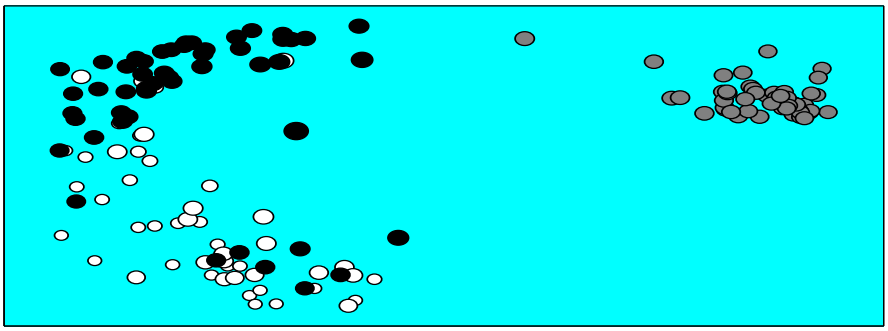


(b) T-NS

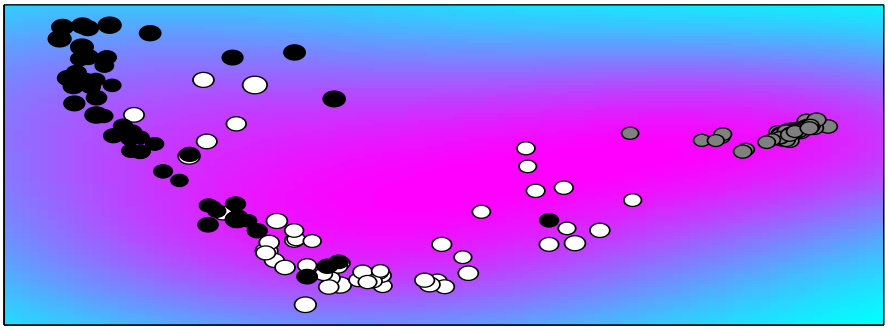


(c) PLLE

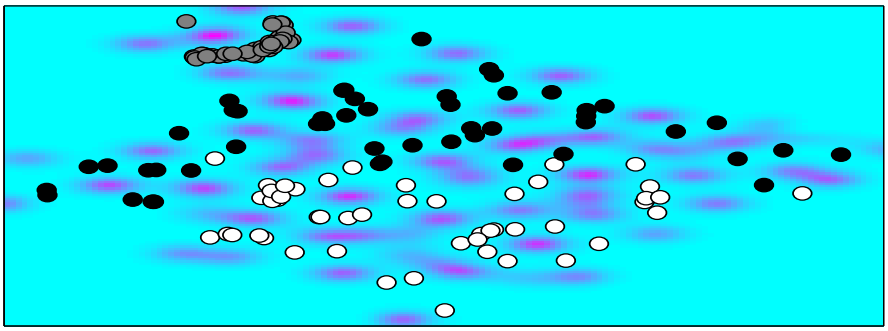
Figure 5.4: Visualisations of the MNist data using (a) N-NS , (b) T-NS and (c) PLLE. The observations are plotted as circles, coloured by class, with size given by the relative level of mapping surprise. These points sit upon the Uncertainty Surface density map with pink indicating regions of high probability and blue, low probability. All three visualisations separate classes with an overlap between "0"'s and "6"'s as in the Sammon map of figure 5.3. N-NS and T-NS generate identical visualisations with "6"'s mapped closer to "1"'s than "0"'s. Due to the relatively high precision of each observation, the Uncertainty Surface in the N-NS mapping appears flat. The T-NS Uncertainty Surface shows that "1"'s appear in higher probability regions than "6"'s and "0"'s. In PLLE "6"'s are the most probable observations and "1"'s are, in general, more anomalous, appearing largely in the purple region. The mapping surprise level for these visualisations is approximately constant, except for the "0" and "6" outliers in the on the left of the mappings.



(a) PIso



(b) PWNM

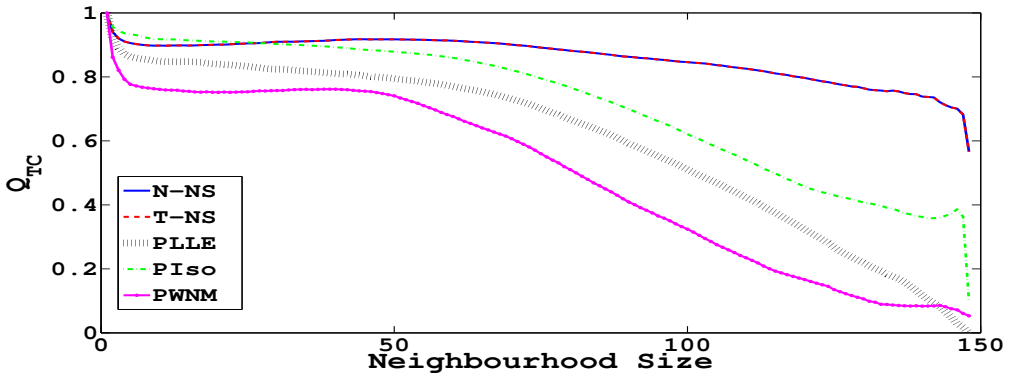


(c) GPLVM

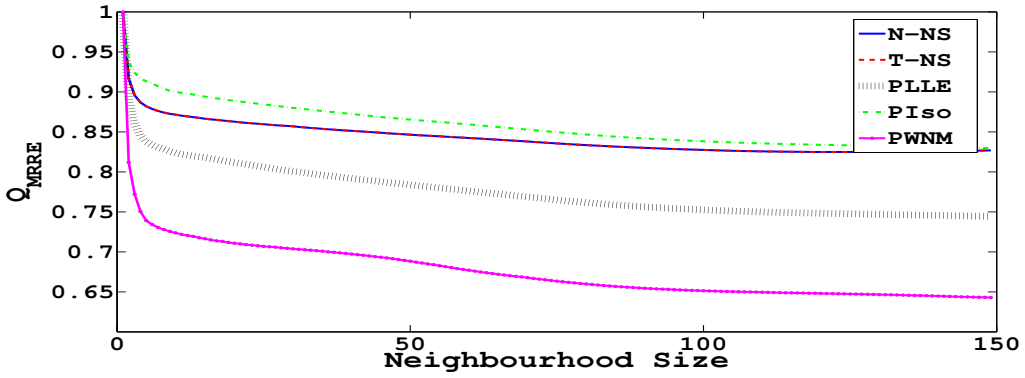
Figure 5.5: Visualisations of the MNist data using (a) PIso, (b) PWNM and (c) GPLVM. The structure of the PIso mapping and its' Uncertainty Surface are almost identical to that of N-NS in figure 5.4a. The PWNM visualisation space appears to show a 'V' shaped manifold where "6"'s are seen as anomalous in terms of both the location on the Uncertainty Surface and relative mapping surprise (shown by the larger points than the other two classes). The centre of the Uncertainty Surface (pink) appears unpopulated, but the shape is correctly summarised by the 150-order GMM in the visualisation space. The GPLVM mapping is similar to the Sammon map of figure 5.3 with the same separation and overlapping clusters. The posterior probability surface is uninformative showing areas of high probability between observations in separated patches. This is an indicator of poor interpolation of the observation space by the kernel function.

surprise indicating a level of precision in their locations. An altogether different structure is found by PWNM (again with $k = 8$ neighbours) in a 'V' shape. As with PLLE the cluster of "1"'s is separated from the "0"'s by the "6"'s along the 'V' manifold. However, here the outliers from the "6" cluster have been placed away from the "0"'s off the manifold with a high level of mapping surprise. The GPLVM mapping performs better than PLLE, finding a tighter cluster of "1"'s and some discrimination between the two other classes. However, the Posterior Probability Surface, the GPLVM equivalent to the Uncertainty Surface, is 'fractured' with multiple regions of isolated high-confidence areas with no supporting data points.

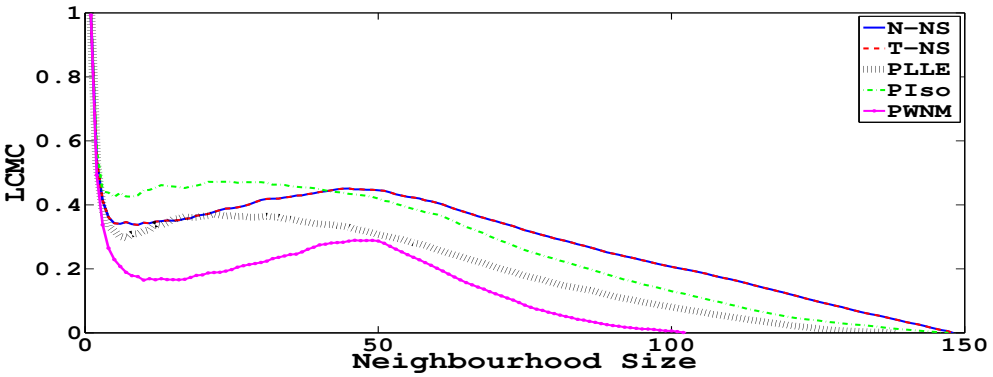
The inherent covariance matrix for the observations is reduced from $\mathbb{R}^{784\times784}$ to an uncertainty matrix in $\mathbb{R}^{2\times2}$ using SVD rank reduction described in chapter 3, with 35% of the variance from observation space preserved in this process. The inverse of this low rank matrix, known as the precision matrix, indicates that the uncertainty surrounding each latent mean is small. For this reason, the Uncertainty Surfaces for the N-NS (figure 5.4a) and PIso (figure 5.5a) appear to be flat and uninformative. The distributions in this case reduce almost to a series of delta peaks for each observation. The Uncertainty Surface in the case of the T-NS mapping is more informative due to the penalisation of the precision matrix by the degrees of freedom, set here to three (chosen for the number of classes). This surface states that given a new observation it is most likely to be a "1" and sit within the higher probability area (pink). Two smaller high density parts of the surface are present in the cluster of "6"'s, having a tighter distribution than that of the spread of "0"'s. The latent spaces generated by PLLE and PWNM have more informative Uncertainty Surfaces since the initialised points prior to optimisation are set using LLE and LE respectively. These two initialisations use constraints to set the relative scale of points (described in chapter 2) and as such the points in PLLE and PWNM visualisations are closer in terms of absolute distance than the NS-based methods. Since the covariances of each latent distribution are the same for each mapping (except for the penalised T-NS case) the closer proximity of the means of these distributions in PWNM and PLLE, in terms of absolute distances, causes the combined Uncertainty Surface to appear more spread than that of N-NS. The PLLE Uncertainty Surface indicates there



(a) Q_{TC}



(b) Q_{MRRE}



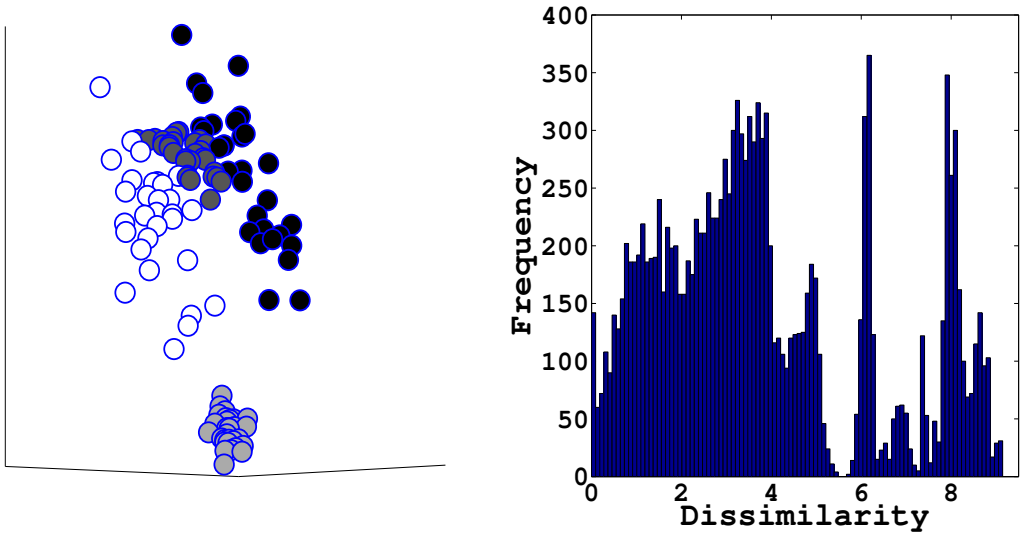
(c) LCMC

Figure 5.6: Quality criteria for visualisations: (a) Q_{TC}, (b) Q_{MRRE}, (c) LCMC as a function of neighbourhood size. PWNM has the worst performance for the dataset, achieving low neighbourhood rank preservation and trustworthiness. Visualisations generated by N-NS, T-NS and PIso are on the other hand the best performing with good overall relative rank preservation (Q_{MRRE}). The drop in terms of trustworthiness and local continuities occur beyond neighbourhoods of 50 since other classes are included in the neighbourhoods. Local continuity is poor for neighbourhoods of 100 and over since the actual neighbourhoods are not preserved; only neighbourhood rankings. PLLE achieves the average performance of all the mappings considered with neighbourhood preservation similar to the global methods of N-NS, T-NS and PIso. This is expected since it enforces neighbourhood preservation through its' cost function. On the other hand, the trustworthiness for larger neighbourhoods is similar to PWNM.

are anomalies in the class of "0"'s and that the most certain class is "6". This seems intuitive since the algorithm trains on the structures of the circles in "0"'s and the lines of "1"'s and "6"'s can be thought of as a combination, making them theoretically the best understood class. The contrary occurs with PWNM (figure 5.5b) where "6"'s are outliers compared to the more probable "0"'s and the most probable "1"'s. An interesting consequence of the equal weighting of the (overspecified) Uncertainty Surface seen here is that the highest probability regions of the surface are relatively unpopulated. The GPLVM equivalent of the uncertanty surface is the underlying posterior probability distribution (figure 5.5c) which indicates isolated islands of higher probability between observations. This information is more confusing than the flat Uncertainty Surfaces. So much of the unpopulated latent space appears to have a high probability of observation, preventing anything from being inferred regarding the distributions of the latents.

The quality criterion plots (as described in section 2.5) are shown in figure 5.6. Of the mappings shown, PIso and N-NS/T-NS perform the best, in terms of the rank based criteria, for creating trustworthy visualisations. The results for the three criteria are largely the same, penalised dissimilarities (geodesics in PIso and the use of degrees of freedom in T-NS) generate mappings where neighbourhoods are better preserved. The PLLE mapping is a middle ground of the five methods. Beyond neighbourhoods of 50 points each of the algorithms seems to suffer from some level of performance decay. This is because neighbourhoods of greater than 50 points will definitely include more than one observation from a separate cluster. The quality improvements in terms of LCMC between neighbourhoods of 10 and 50 is due to the fact that the classes are better reconstructed in terms of all neighbours, whereas some of the smaller inter-class neighbourhood clusters are not appropriately grouped.

The probabilistic mappings developed in chapter 3 have performed well in learning the structure of the MNist data. The visualisation spaces are similar to the Sammon map shown in figure 5.3 with greater between-class separation. This is due to the dissimilarity measure here being a form of weighted Euclidean distance as opposed to the isotropic Euclidean distance. It should be noted that perfect separation between classes is not expected since not all "6"'s and "0"'s belong to their respective clusters in observation



(a) Data-space (b) Dissimilarity Histogram

Figure 5.7: (a) The four artificial clusters from section 5.3 in 3-dimensional observation space and (b) histogram of dissimilarities. Three of the clusters are closely located; one (dark grey) situated between two more spread clusters (white and black). The fourth cluster is entirely separated from the group (light grey) with a lower level of class spread. A successful visualisation should keep the light grey cluster separated from the other overlapping clusters. The close proximity of the three clusters in observation space ensures that the dissimilarities between these points is low with larger dissimilarities caused by the separated cluster.

space. This is seen in the dissimilarity matrix and Sammon mapping of the observations.

5.3 Four Clusters Dataset

An artificially generated set of four 3-dimensional clusters was created where each of the four groups has a unique uncertainty. This could be typical of the errors experienced in measurement systems where uncertainties can be impacted on a global scale by, for instance, location or time. The dataset consists of four clusters of 30 observations, each sampled from a separate Gaussian distribution with mean and full rank covariance matrices known. The scatterplot of the original data is shown in figure 5.7a.

There are two widely spread clusters (black and white) with a tighter central cluster (dark grey) and a completely separated cluster (light grey). Each cluster is sampled from a Gaussian distribution in 3-dimensional space with a class-specific covariance matrix. The 120 observations are then treated as a Gaussian distribution with observed mean and

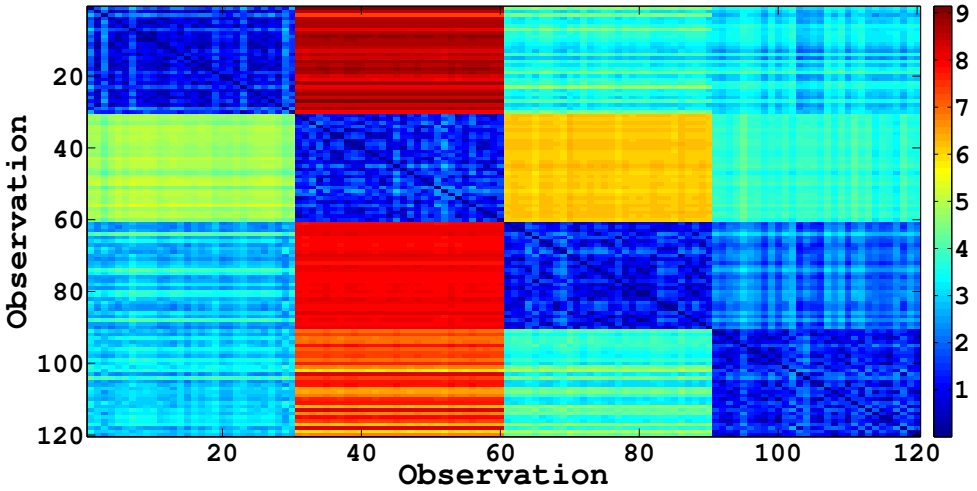


Figure 5.8: Dissimilarity matrix for the four clusters dataset incorporating uncertainty. The separated cluster (observations 30 to 60) is significantly more dissimilar to the other three clusters than they are to one another. Clusters 3 and 4 (observations 60 - 90 and 90 - 120 respectively) are closely located in dissimilarity space as they are in close proximity in observation space.

known class covariance matrix, i.e.:

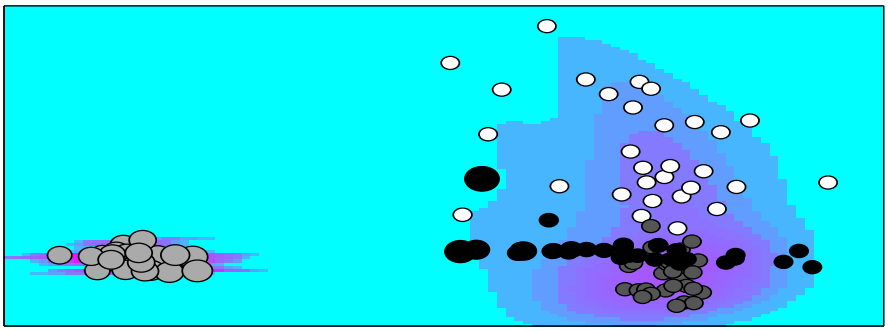
$$P(X_i^j) = \mathcal{N}(X_i, S_j),$$

where each observation, i from class j, is characterised by the Gaussian distribution with covariance matrix, S_j, from class j. The measure used to compare the dissimilarity between the uncertain observations is again the Kullback-Leibler divergence. It should be noted that the dissimilarity is not symmetrised as it was in [33] by design. There is no particular reason to impose a metric space on observations, particularly when they are often better characterised by non-metric distances such as those used in Isomap. The work in [64] showed that better classification, and therefore discriminative power, was found by not imposing these standard spaces using instead, for instance, Psuedo-Euclidean or Krein spaces. The dissimilarity histogram (figure 5.7b) shows that there is a large concentration of low dissimilarities, caused by the small within class dissimilarity contributions and close proximity of three of the clusters. The dissimilarity matrix in figure 5.8 shows that three of the clusters are closely located in dissimilarity space as well as observation space.

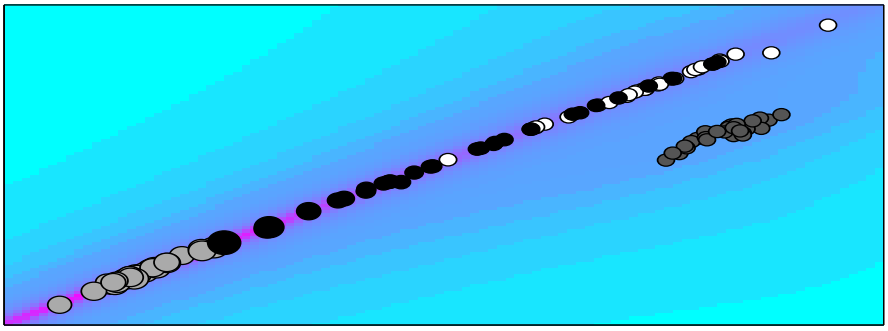
It is assumed that the covariance matrices characterising each cluster are known a-priori (methods to estimate these exist, for instance, the GP-based approach of [72]). This may seem like an unlikely scenario, but this could be the case in a measurement-specific application, for instance the seasonal variations in the uncertainties of weather measurement devices. Using the Kullback-Leibler divergence to compare observations, the visualisations in figures 5.9 and 5.10 were generated.

A connected graph was achieved using $k = 8$ neighbours so the neighbourhood sizes for PLLE, PIso and PWNM were fixed to this value. There are many similarities between the N-NS (figure 5.9a), PLLE (figure 5.9c), PIso (figure 5.10a) and PWNM (figure 5.10b) visualisations shown. The distinct separation of the light grey cluster is as expected from the original plot in figure 5.7a. The treatments of this distinct cluster, as being contained in a region of high probability from the Uncertainty Surface, indicates the accuracy of the mapping of this cluster. The three remaining clusters are located almost identically in the PWNM and PLLE visualisations (black - dark grey - white). The order is switched with the N-NS and PIso mappings (dark grey - black - white). The mapping surprise is relatively uninformative for the PWNM and PLLE latents. This was expected since all of the observations were generated using similar, relatively precise uncertainties, ensuring the relative dissimilarity and covariance is small. The T-NS mapping found using $\nu = 4$, since there are four clusters, appears as a linear relationship except for the dark grey cluster (central in the observation space). Even with the linear relationship there is separation between the light grey cluster and the black cluster. The level of overlap between the black and white clusters here is comparable with that of N-NS, despite the self-imposed linear manifold limitation. The GPLVM visualisation in figure 5.10c separates the clusters in a similar way to PLLE and PWNM. However, the Posterior Probability surface is indicating the unpopulated areas as high probabilities of observations which seems unbelievable.

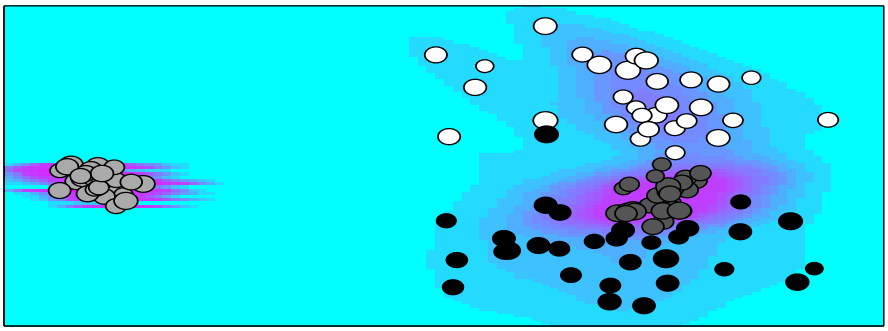
Figures 5.11a, 5.11b, 5.11c show the Q_{TC}, Q_{MRRE} and $LCMC$ respectively for the mappings of the clusters dataset. In terms of trustworthiness and continuity, the mappings are largely the same, except for the superior performance of PWNM. The Q_{TC} decays for neighbourhood regions of greater than 90 observations. This is likely due to



(a) N-NS

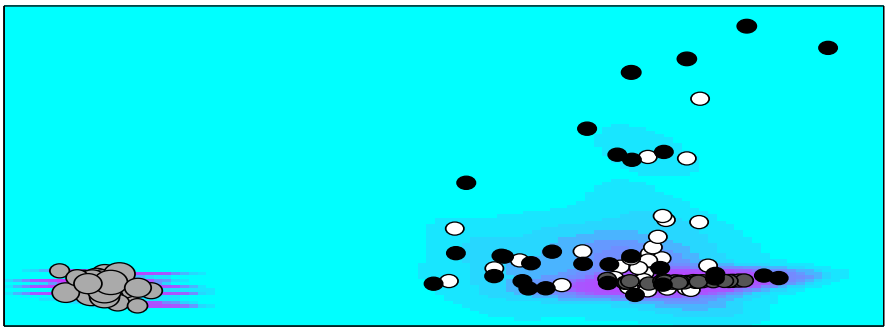


(b) T-NS

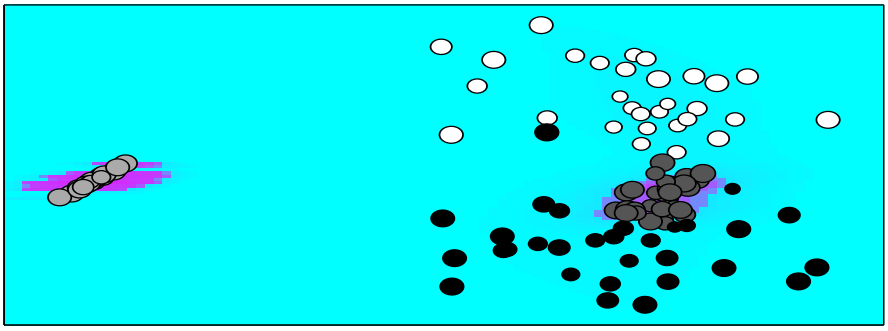


(c) PLLE

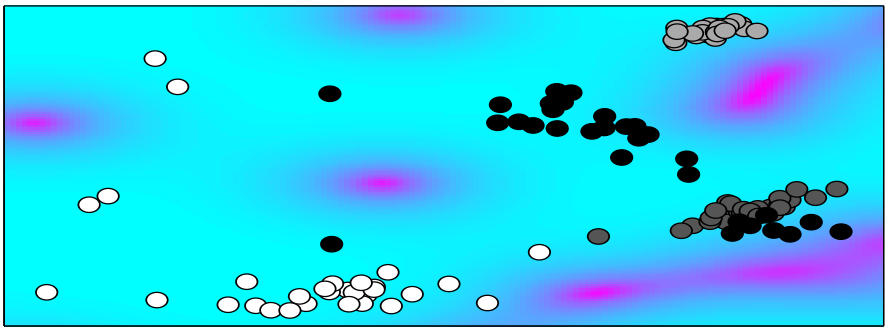
Figure 5.9: Visualisations of the four clusters data generated by (a) N-NS, (b) T-NS and (c) PLLE. N-NS and PLLE group the clusters in close proximity in observation space, with greater levels of spread given to the groups with the largest degree of spread in observation space (white and black). All three of the above place the separated cluster (having the highest level of precision) on the highest density area of the Uncertainty Surface. T-NS finds a largely linear structure with the central overlapping cluster separated. This cluster is situated in the lower probability region. The most anomalous observations come from the separated clusters in N-NS and T-NS. The mapping surprise for PLLE is largely the same, showing no surprising observations. The neighbourhood size of $k = 8$ is therefore large enough to weight neighbourhoods in the 30 observation clusters without achieving equal weightings.



(a) PIso

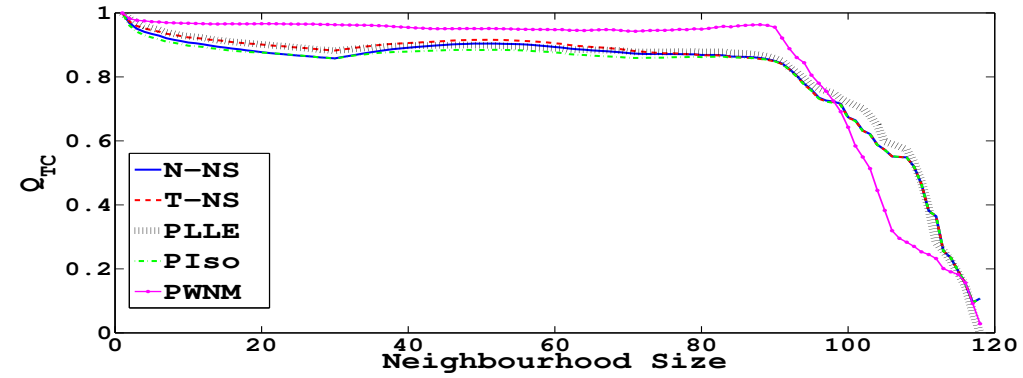


(b) PWNM

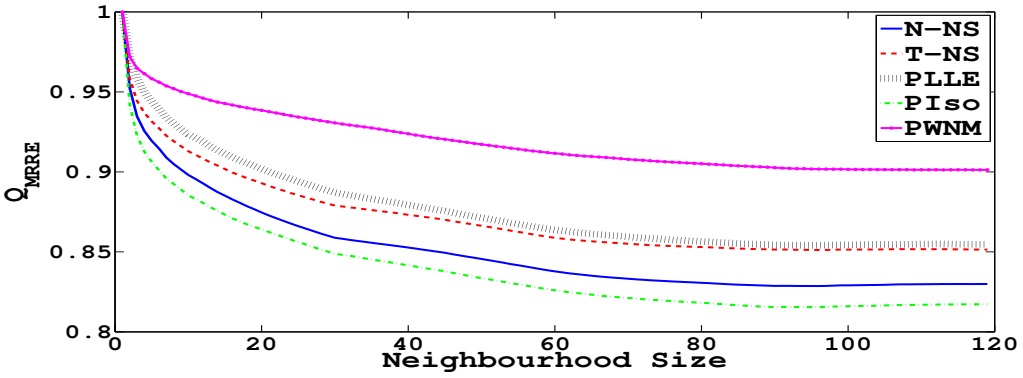


(c) GPLVM

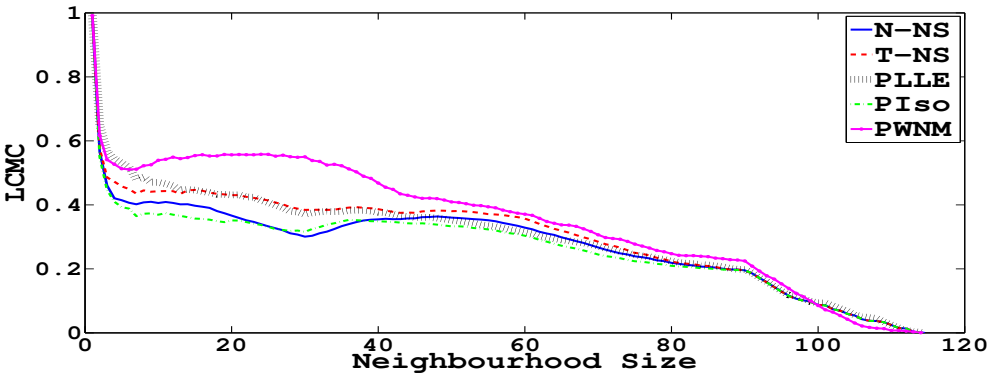
Figure 5.10: Visualisations of the four clusters dataset using (a) PIso, (b) PWNM and (c) GPLVM. PIso and PWNM map the separated cluster far from the others, whereas GPLVM leaves a smaller gap between it and the black cluster. In the PIso visualisation the black cluster is spread far, overlapping with both the dark grey cluster (as in observation space) and with the white cluster (unlike in the observation space). The dark grey and light grey clusters are mapped to high probability areas on the Uncertainty Surface, as with PWNM. The mapping surprise in PIso highlights the separated cluster as being anomalous, caused by the large geodesic distances between the black clusters' points in the far right of the space. The mapping surprise is less spread in PWNM but the cluster separation and groupings appear faithfully preserved. The GPLVM latent space has prevented the dark grey cluster from overlapping with the white cluster as it did in observation space and it has disconnected high probability islands of certainty away from data points.



(a) Q_{TC}



(b) Q_{MRRE}



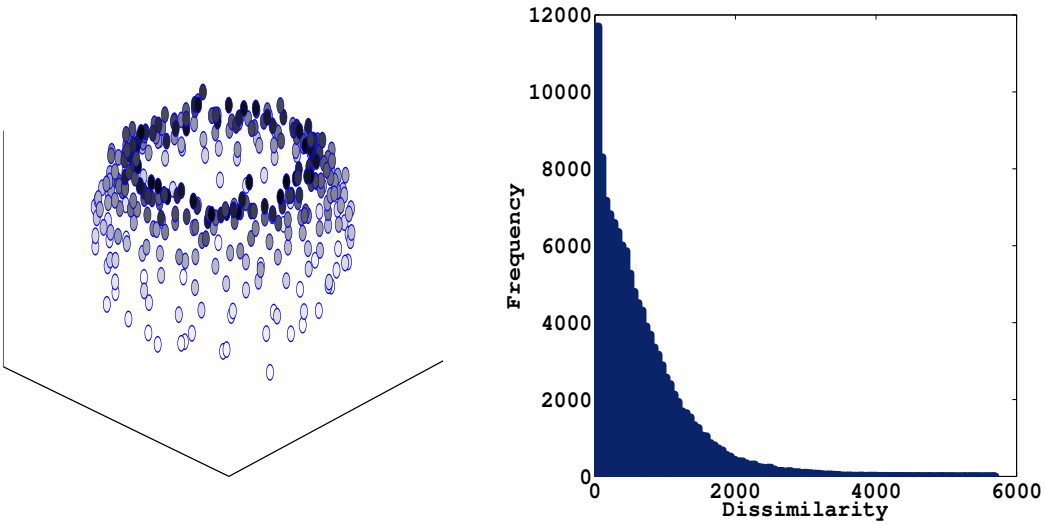
(c) LCMC

Figure 5.11: Quality criterion for visualisations: (a) Q_{TC}, (b) Q_{MRRE} and (c) LCMC. PWNM outperforms the other mappings in terms of neighbourhood rankings and the trustworthiness is higher for neighbourhoods less than 90. There is a decay in Q_{TC} for all visualisations beyond this point as it concerns the relationship between the three connected clusters and the separated cluster. The worst performance is seen in N-NS and PIso due to the loss in continuity between the overlapping clusters which are forced to go from dark grey to black to white instead of being centered at the dark grey cluster. The Q_{MRRE} results are, however, better than the performance on the MNist dataset and appear constant beyond neighbourhoods of 60 relating to two adjoining clusters. Overall, PWNM and PLLE have generated the most reliable visualisation spaces.

the fact there are four clusters each consisting of 30 points with three of those in close proximity, causing a loss in T and C when the final separated cluster is considered. The visualisations are considered truthful with a Q_{TC} of almost 0.9 for three quarters of the dataset. The Q_{MRRE} results (figure 5.11b) paint a similar picture with all mappings performing well in preserving the data ranking of observations. There is a steady drop in performance of neighbourhood sizes between 3 and 60 before approaching a constant rank error. As with Q_{TC}, PWNM performs the best for this dataset; followed by PLLE. These methods use only local graph structures to embed observations which will have benefitted the clustered nature of the dataset. PIso and N-NS perform the worst in terms of all measures, possibly due to the global focus of the training procedure which has changed the topological ordering of points. The geodesic distances in PIso appear to have disproportionately extended the dissimilarities between some observations, causing neighbourhood links to be broken. The results for the LCMC (figure 5.11c) shows steady performance decays for neighbourhoods greater than three as well. PWNM boasts the best results for neighbourhoods of between 8 and 60 observations, neighbourhoods of greater than 60 become relatively indistinguishible for the methods used.
The quality measures indicate the best mapping was produced using the PWNM algorithm. The latent space is visually appealing, showing the four clear clusters appropriately with a representative Uncertainty Surface. The mapping surprise performs, for the most part, intuitively highlighting the separated cluster as the most anomalous.

5.4 Punctured Sphere Dataset

The Punctured Sphere dataset is an artificially generated structure in 3-dimensional space shown in figure 5.12a using the *mani* toolbox [73]. It is a popular test for topographic mapping techniques as many struggle with appropriately mapping the sparse sampling region compared to the densely populated open top. The hole at the sphere top also poses a problem and can cause different mappings for geodesic versus Euclidean based dissimilarity mappings, for example in [15]. The sphere used here was generated with 350 observations in a deterministic way. These points were then subjected to



(a) Data-space (b) Dissimilarity Histogram

Figure 5.12: (a) The punctured sphere observation space and (b) histogram of dissimilarities. Plotted points are coloured from observation 1 (white) at the sphere base to 350 (black) at the open top of the manifold. The open top of the sphere structure is clear with the densely-sampled region of black points here. The uncertainties imposed on the manifold make some points appear to be perturbed and separated from the smooth structure. The histogram of dissimilarities shows that many of the observations are close in dissimilarity space as many are concentrated towards the top of the densely populated structure. The larger dissimilarities are caused by the sparsely sampled observations at the base of the sphere.

data-specific uncertainties from randomly created covariance matrices such that each observation is a full-rank elliptical Normal distribution with observed mean. The covariances were generated as:

$$S_i = AA^T + 0.1tr(AA^T)I_3,$$

where A is a 3×3 matrix where entries are randomly generated numbers in the interval [0,0.5]. This interval was chosen such that the overall structure of the manifold is not lost by these perturbations. The right hand term above ensures that the matrix is numerically stable for inversion. A Tikhonov regularization scheme could also have been used in the case the matrix is not stable for inversion. As with the previous two examples in this chapter the mappings are performed with the uncertainties assumed known.

The dissimilarity between observations is again taken to be the Kullback-Leibler divergence between the observed Normal distributions. The histogram of dissimilarities

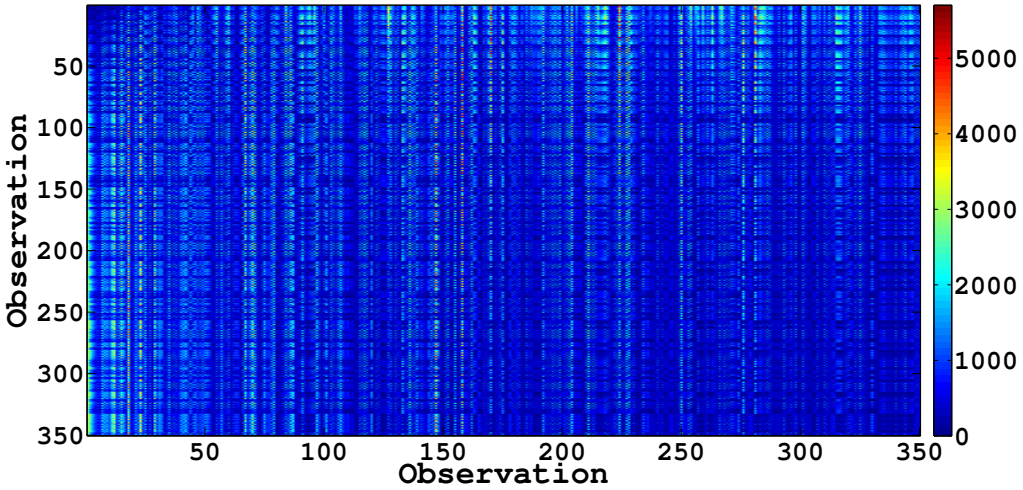
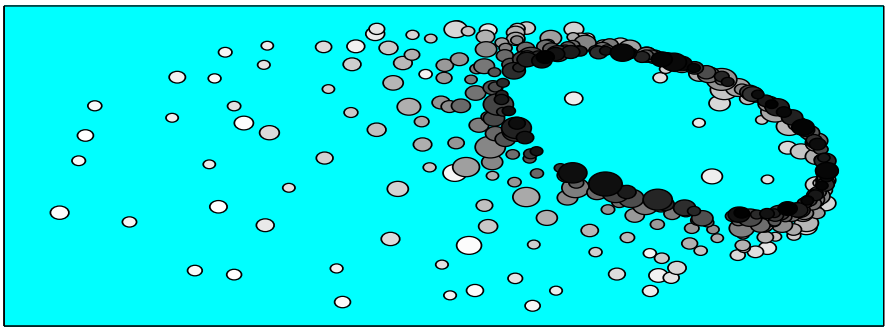


Figure 5.13: Dissimilarity matrix for the uncertain punctured sphere dataset. Low values for the Kullback-Leibler divergence are seen in the bottom right and top left of the dissimilarity matrix, relating to the observations at the open top of the sphere. The larger dissimilarities are found at the observations from the sphere base (observations 1 to 50) with some anomalies appearing throughout the dataset, e.g. the dissimilarities with respect to observation 149 are higher than expected due to an unusual covariance structure. The matrix is not symmetric as would be expected since the Kullback-Leibler divergence is not symmetrised, as in section 5.3. It may not be clear, but the diagonal elements of the matrix (the self-dissimilarities) are all zero. The larger dissimilarities seen in the horizontal lines are numerical artefacts from the inversion of some of the matrices, having larger values than the rest of the matrices.

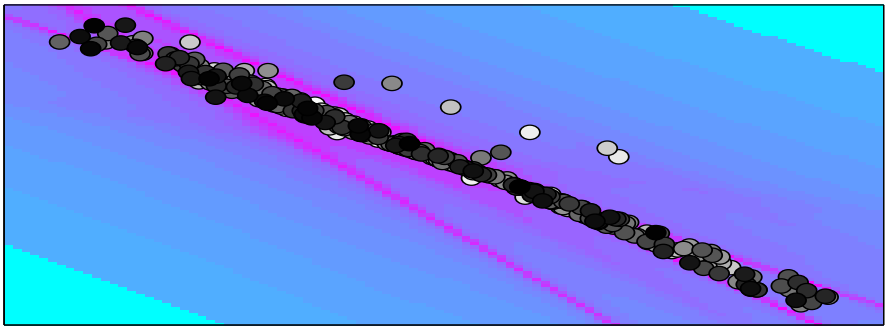
in figure 5.12b emphasises the densely populated region at the top of the sphere. The larger dissimilarities are caused by the sparsely populated base of the sphere. This is also seen in the dissimilarity matrix of figure 5.13 where there are few observations with high levels of relative dissimilarity.

The visualisation spaces generated from the above described sphere are shown in figures 5.14 and 5.15. A fully connected graph was created by using $k = 8$ neighbours and therefore used for PIso, PLLE and PWNM. The degrees of freedom is fixed to $\nu = 3$ in T-NS as the observations are in 3-dimensional space.

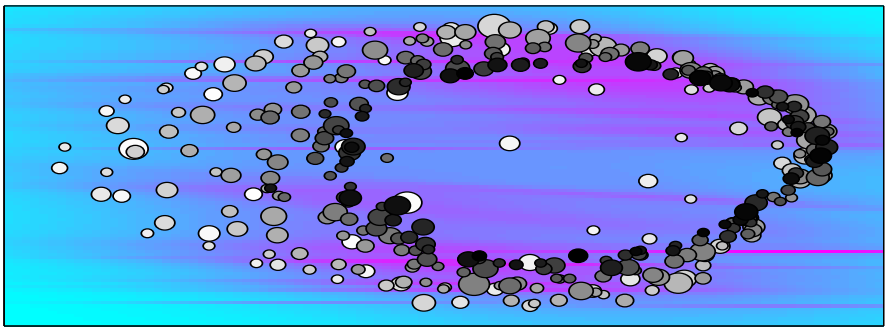
It is clear that there are many similarities between the mappings of N-NS (figure 5.14a), PLLE (figure 5.14c), PIso (figure 5.15a). The sphere has been compressed from above at an angle in a similar way to the Sammon mapping in [15]. The sparser sampled sides of the structure are peeled away on the left hand side with larger separation than that of the top of the original structure. The seven observations from the bottom of the structure are



(a) N-NS

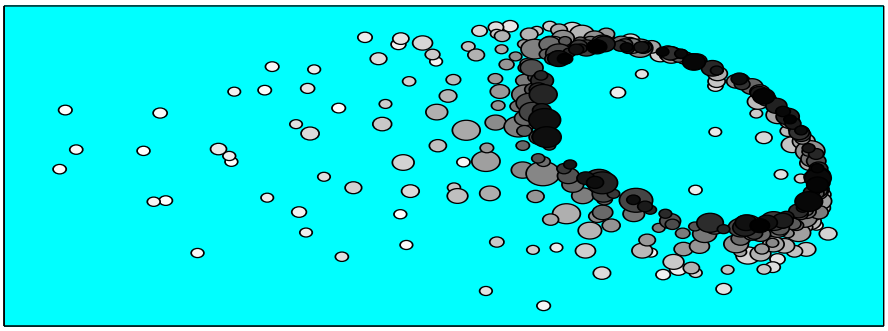


(b) T-NS

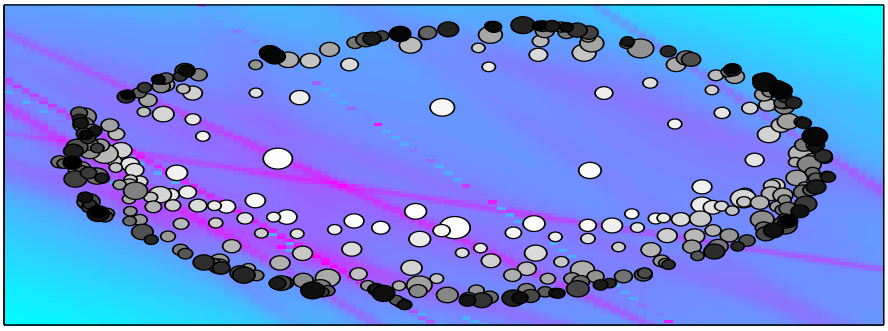


(c) PLLE

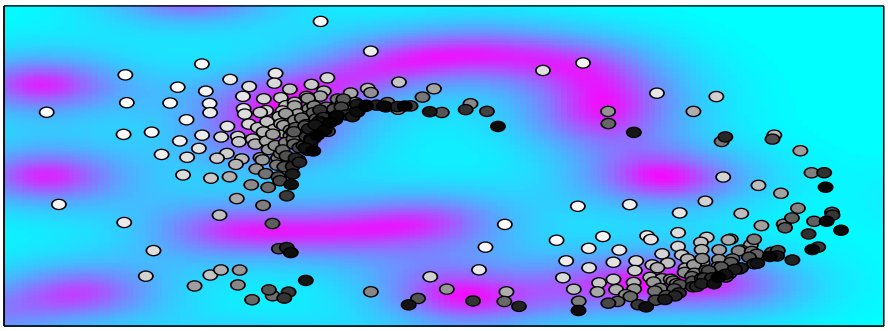
Figure 5.14: Visualisations of the uncertain punctured sphere dataset using (a) N-NS, (b) T-NS and (c) PLLE. The N-NS visualisation appears to be a side view of the sphere with the bottom observations torn to the left side of the visualisation space. The Uncertainty Surface is flat as with the MNist dataset because of the relatively high precision of each observation. The mapping surprise highlights some of the observations from the side of the structure as being anomalous due to them being located off of the surface of the sphere in the observation space. T-NS has found an entirely linear structure to represent the data, tearing the manifold along the side. This introduces a discontinuity and impacts the mapping quality in figure 5.17. The PLLE visualisation is a more top-down view than the N-NS, highlighting similar anomalies. The Uncertainty Surface intuitively highlights more densely populated areas as being expected. This is caused by the LLE initialisation.



(a) PIso



(b) PWNM



(c) GPLVM

Figure 5.15: Visualisations of the uncertain punctured sphere dataset using (a) PIso, (b) PWNM and (c) GPLVM. The PIso mapping has preserved the topological ordering of the dataset with observations from the base of the sphere pulled to the far left of the visualisation space. The mapping surprise highlights the same anomalies as the N-NS mapping and the Uncertainty Surface appears flat. The PWNM visualisation unfolds the sphere from the open top, placing the densely sampled top on the higher probability region of the Uncertainty Surface. Higher mapping surprise is seen in the sparsely sampled base of the structure. The mapping surprise is distorted by the uncertainties of a small number of observations. The GPLVM latent space squashes the structure in a similar way to PIso but separates along the middle. The posterior probability surface is again misleading, finding both high and low probability regions in the densely populated regions.

placed within the centre of the N-NS and PIso manifolds. Interestingly, the PIso and N-NS mappings place the latent means in almost identical relative positions. The PLLE mapping is found by squashing the sphere from above with the left hand side points being spread, leaving the right hand side points in microclusters with the higher top points. The Uncertainty Surfaces from N-NS and PIso are uninformative. This is caused by the large relative dissimilarities between observations which ensures the means are separated in terms of absolute positions. This separation is at a level where the precision of the uncertainty matrices appears as a set of delta peaks along the surface. On the other hand, the PLLE mapping often places the means closer together in terms of their absolute positions since the cost function does not relate directly to the observation space. The Uncertainty Surface for PLLE appropriately described the densely populated areas in latent space as higher probabilities (pink). It is apparent that the covariance matrices of the latent distributions are not simply isotropic, due to the visible horizontal striations on the Uncertainty Surface. This emphasises the benefits of choosing the SVD approach for embedding a covariance matrix in a 2-dimensional space over the determinant-based method described in section 3.3. The determinant-based embedding of S_i would result in large isotropic uncertanties which are not present in the original covariance matrix, replacing the striations with circles of equivalent size. T-NS (figure 5.14b) again minimises its STRESS measure by tearing the structure open directly down the side and finding a linear latent relationship between the observations. The latent means are often placed atop of one another and the Uncertainty Surface is uninformative. With this in mind, the visualisation is considered to be poor for this dataset. PWNM (figure 5.15b) tears the structure open from the open top creating a continuous circle of the top-most observations with the sparser sampled sides contained within this circle. The denser areas of the visualisation space are contained within higher probability areas of the Uncertainty Surface with the horizontal lines similar to those in PLLE. The mapping surprise highlights similar points in N-NS, PIso and PLLE as being unusual, often the points from the side where the sparser sampling from observation space begins. Interestingly, PWNM highlights the central points using the Fisher Information measure, making them larger, which, one would naturally assume, are the most dissimilar

observations. The GPLVM mapping (figure 5.15c), based on deterministic points as observations, seems to tear the structure in half prior to enforcing a squashing motion. The points in the two halves are clustered appropriately but this will have a significant impact on the trustworthiness and rank error of the mapping. This illustrates that the noisy structure is not a simple manifold to visualise, as the deterministic version in [15] is. The posterior distribution in the latent space is again difficult to understand with high point density areas occuring in low probability spaces and low point density areas occuring in high probability spaces. As with the T-NS mapping, the tearing motion is hard to interpret and the underlying latent distribution makes the GPLVM visualisation a poorer mapping of the dataset than most of the other probabilitic mappings above. Given the entirely linear structure found by T-NS, the experiment was re-run increasing ν gradually. The mapping results improved until $\nu = 35$. Beyond this point the mapping becomes algebraically similar to N-NS and therefore the choice of latent distributions should be Gaussian instead of T-distributed. Figure 5.16 shows the mapping of the punctured sphere with $\nu = 35$ degrees of freedom. A linear structure is still present, but the curvature is also imposed when the top-most observations from the sphere are considered. This mapping was re-run with different initialisations, in an attempt to test if this was a poor local minima, without change. The Uncertainty Surface is informative in a similar way to that of PWNM, with higher probabilities over the denser regions. Mapping surprise considers the bottom-most observations as anomalous, but also some of the top-most observations at the start of the puncture. This mapping, although not as visually appealing as the others, is a vast improvement upon the original T-NS mapping. The quality criterion for the punctured sphere dataset are shown in figures 5.17a, 5.17b and 5.17c. The results are shown for the above mappings including the $\nu = 35$ mapping of T-NS for comparison. The performance for Q_{TC} decays for neighbourhoods greater than 60 for all mappings. The results for N-NS are best, with PIso behaving similarly until the general performance drop. The T-NS ($\nu = 3$) mapping is the poorest in terms of all criterion, however for $\nu = 35$ the mapping is only slightly worse than the N-NS and PIso despite the largely linear reconstruction of many observations. This steady decay for all mappings indicates a good global reconstruction (better than the MNist dataset

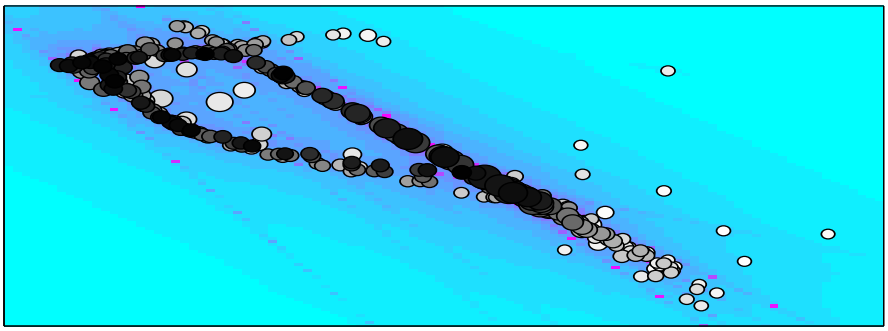
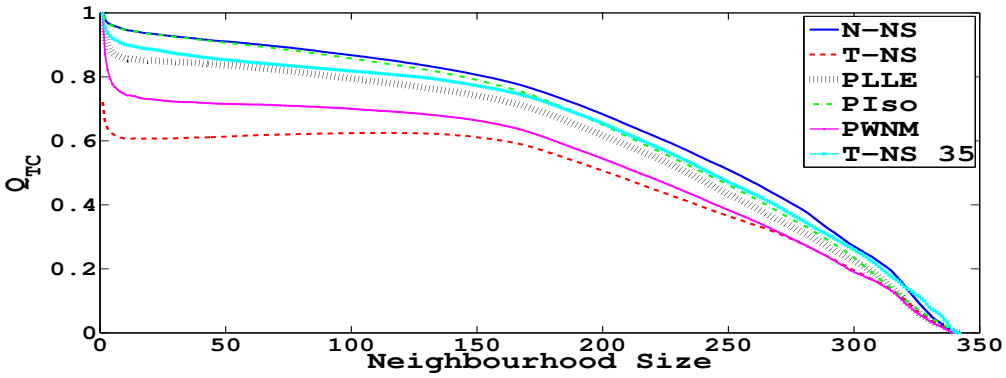


Figure 5.16: T-NS mapping of the uncertain punctured sphere dataset with $\nu = 35$. The mapping is more representative of the observation space with a more circular structure than the original mapping in figure 5.14b. The mapping surprise finds the observations at the base of the structure anomalous as is expected. The more informative Uncertainty Surface indicates nearly all observations occur in the high probability region.

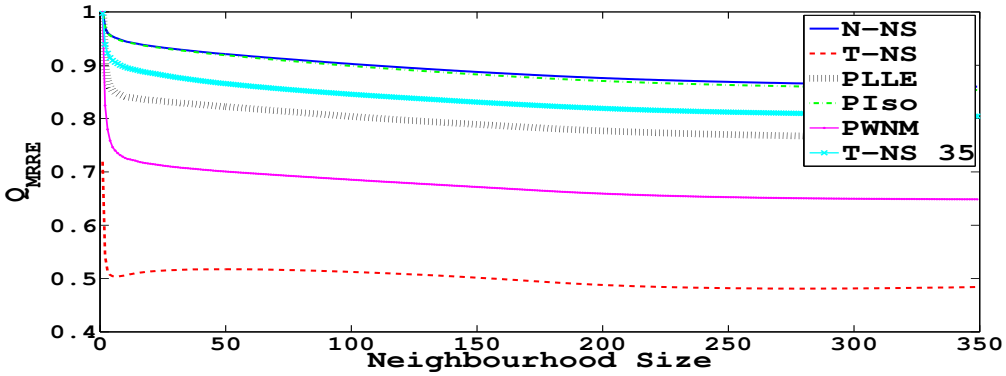
reconstruction for instance). The Q_{MRRE} and *LCMC* criterion behave the same for each of the mappings, except that the performance for Q_{MRRE} decays slowly, almost to a constant, past neighbourhoods of size 20 and the *LCMC* decays quickly beyond neighbourhoods of size 25. These results are also an improvement on the MNist dataset despite the increased difficulty caused by the multiple covariance matrices and larger dataset for the punctured sphere structure. Interestingly, the mapping generated by PWNM is similar to that described in [15] as being a best case scenario for the dataset, found by unfolding the sphere from the open top. However, the quality criterion rank the performance of this mapping worse than NS-based methods, resembling the Sammon map of the original dataset shown in [15]. The N-NS and PIso visualisations perform the best in terms of the quality criterion.

5.5 Overview

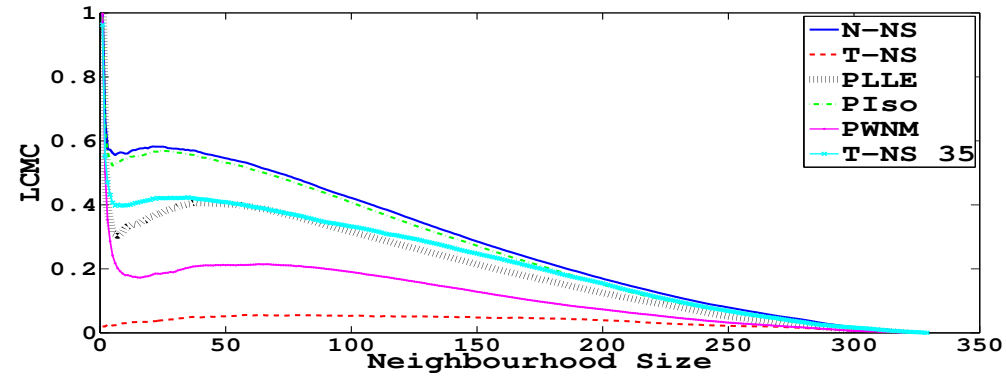
This chapter has implemented the probabilistic algorithms developed in chapter 3 with the uncertainty measures of chapter 4 on three vectorial datasets. The datasets consist of uncertain observations initially with one global uncertainty measure for the MNist dataset. The same algorithms are then implemented on the four clusters dataset with class-specific covariance matrices. The final dataset, the punctured sphere, involved



(a) Q_{TC}



(b) Q_{MRRE}



(c) *LCMC*

Figure 5.17: Quality criterion for visualisations of the uncertain punctured sphere dataset: (a) Q_{TC}, (b) Q_{MRRE} and (c) LCMC. PIso and N-NS perform almost identically achieving high rank preservation (Q_{MRRE}) and better trustworthiness (Q_{TC}) than all other mappings. T-NS with $\nu = 35$ performs almost as well, despite part of the visualisation being approximately linear. T-NS with $\nu = 3$ achieves poor results as expected. PLLE ahieves the middle results for the group again.

observation-specific covariances in the mappings. Overall, the mappings perform similarly for all three tasks with T-NS being the most unusual. The choice of the neighbourhood size, k, make PLLE, PIso and PWNM more awkward to set, but PIso seems to perform as well as N-NS with connected graphs. The visualisation spaces

Dataset	Best	Worst
MNist	N-NS / T-NS	PWNM
Four clusters	PWNM	PIso
Punctured Sphere	N-NS	T-NS($\nu = 4$) / PWNM

Table 5.1: Best and worst performance of visualisation quality criterion for vectorial datasets

generated are similar, but N-NS does the best at conveying different levels of mapping surprise, indicating good data interpolation. PIso can overestimate the dissimilarities by using geodesic distances. GPLVM generates interesting visualisation structures but the posterior probability maps (the GPLVM version of the Uncertainty Surfaces) have a negative impact on the interpretation of the visualisations when compared to the RBF counterparts. The lack of mapping surprise here offers the other five algorithms an additional benefit over visualisations generated using GPs.

Table 5.1 shows the best and worst performance results for the datasets used in this chapter. The best visualisations of the proposed methods appear to be generated using N-NS and PIso, however the quality of these visualisations for the four clusters dataset is the worst, as shown in figure 5.11. PLLE consistently achieves the average performance with relatively useful and informative Uncertainty Surfaces. This is an interesting result since in [17] LLE performs poorly compared to many other algorithms when tested on datasets such as those used in this chapter. A T-distributed latent space in PLLE may improve these surfaces, for instance like the T-NS visualisation of the MNist dataset. On the other hand, the mapping surprise is relatively uninformative for PLLE compared with N-NS and PIso for these datasets. It should be noted that T-NS is introduced in this thesis as an alternative to N-NS and not necessarily as an improvement. As such, the paramter choice of ν should be application specific and no grounded methods for the choice are proposed in this thesis. It is clear that no single mapping outperforms the others in all cases. This is a general result comparable with that of classifiers in machine learning, such that no single topographic visualisation algorithm is 'best' in all scenarios. The merits and pitfalls of each individual algorithm must be taken into account when choosing which visualisation algorithm to use.

In future work the sensitivity of the mapping surprise measure in PLLE will be tested on

more diverse datasets. The next chapter introduces the concept of Residual Modelling, a novel approach for generating dissimilarity matrices from time series observations.

6 Visualisation of Time Series: The Method of Residual Modelling

> 'The shortest path between two truths in the real domain passes through the complex domain.'
>
> - Jacques Hadamard

6.1 Introduction

Unlike the previous chapter, uncertainties in time series cannot be treated simply as observations with a mean and covariance matrix. The purpose of this chapter is to show how time series observations, even deterministic, can be transformed to where they can be described in terms of a dissimilarity matrix. A new technique, outlined in [28] and [70] which is named Residual Modelling in this thesis, states that the most important part of an observed time series is not the deterministic signal, but the changes in residual distributions over time.

The assumption of Residual Modelling is that data observations are the result of an underlying, unobserved generator. This generator is typically stochastic and, during the observation stage, the subject of an unknown noise process. Assuming that the noise process, and therefore the generator, can be estimated is optimistic and unrealistic for most data scenarios. As such, it is proposed that when searching for anomalies in time series we can characterise how unusual an observation is based upon the fluctuations in the residual model, comprising both incorrectly estimated signal and additional noise components. To paraphrase John Wheeler; 'the generator is in the fluctuations'. This runs counter to traditional signal processing. The advantage of Residual Modelling is that it does not require an accurate signal model and is not reliant on any unrealistic assumptions as other areas are, for instance the Gaussian Process Time Series model [74].

The intended application of this work is for human visualisation of complex data for anomaly detection. This is particularly important in certain domains, for example; defence, such as the detection of submarine signatures in SONAR [27]; healthcare, with the detection of patients' critical care anomalies in time series [75]; and other integrated critical systems. As more of these systems are becoming automated, problems such as high false alarm rates and reliability arise. With this in mind, the human interpretation of data is vital. As such, humans should be able to visualise and utilise as much information as possible from data, placing them at the heart of the decision making process.

6.2 Residual Modelling

Core to the principle of Residual Modelling is the fact that an observed noisy time series can be described in terms of a deterministic signal and the residual then described through a noise distribution. In traditional time series modelling, for example in [76], an observation is described as an Auto-regressive Moving Average (ARMA) model. The AR components, of delay length m, are first fixed and Gaussian distributions are used as the MA components. AR components of a specified time delay (model order) are fit using the Yule-Walker equations to minimise the mean-square error (MSE). In the case

where the model order is unknown the partial auto-correlation function (PACF) is used. The PACF fits models of different time delays to a set of training data and the model order minimising the MSE is selected.

Traditional linear models are often incapable of appropriately characterising and interpolating real-world observations. As such, the AR model can be replaced with a nonlinear AR (NAR) model. The coefficients should be fit by a suitably adaptive deterministic nonlinear interpolator. RBF networks and Support Vector Machines (SVMs) are popular choices for this task. One typically thinks of SVMs in a two-class classification setting, however they can be set in a regression context in the same 'maximum margin' framework for instance in [77] ([2] provides an introduction to SVM regression). Despite often outperforming RBFs in classification tasks, SVMs are not as well suited to the regression required by Residual Modelling. The gradient descent optimisation over the nonconvex SVM cost function can cause an SVM to fit the same observation differently depending on parameter initialisation. This can lead to a significant change in residuals which will create an artificial anomaly. Therefore regression SVM's are not suitable base models for approximating the time series generator. As such the direct pseudo-inverse optimisation of RBF networks makes them well suited to this task.

The State Space model for observations, $\mathbf{x}_t$, of dimensions $1 \times K$, can be written as:

$$\mathbf{x}_t = \boldsymbol{\theta}_t W + \boldsymbol{\varepsilon}_t, \tag{6.1}$$

where $\boldsymbol{\theta}_t$ is a vector consisting of a nonlinear functional over the delay matrix of observations. The delay matrix is denoted $X_{t-m:t-1}$, a matrix where the rows consist of a delay of $\mathbf{x}_t$ from $t-m$ to $t-1$, namely $[\mathbf{x}_{t-m}, \ldots, \mathbf{x}_{t-1}]$. The length of the delay, m, is known in this chapter as the model order. As such, the dimensions of $X_{t-m:t-1}$ are $(m-1) \times K$. W in the above equation are the weights, and $\boldsymbol{\varepsilon}_t$ is a $1 \times K$ vector, relating to a noise process. $\boldsymbol{\theta}$ is determined by a nonlinear function:

$$\boldsymbol{\theta}_t = \mathbf{f}(X_{t-m:t-1}), \tag{6.2}$$

which is approximated by some interpolation tool. Traditionally the $\boldsymbol{\varepsilon}_t$ term is incorporated into the nonlinearity $\mathbf{f}$ such that the process becomes a NARMA model, however that is not a requirement here. NARMA models often suffer from basic assumptions such as Gaussian noise and the sensitivity to the choice of '**f**'.

Once the observations are concatenated into a delay matrix then $\mathbf{f}$, $\boldsymbol{\theta}$ and $\mathbf{W}$ can be fit using a variant of the PACF. In the RBF case, $\boldsymbol{\theta}_t = \mathbf{f}(X_{t-m:t-1}) = \phi\|X_{t-m:t-1} - C\|_2$ where C is the matrix set of network centres and ϕ a nonlinear basis function, for example a spline or squared exponential function. The weights, W, in equation (6.1) are then optimised as $W = \Theta_t^{\dagger} X_{m:t}$, with $X_{m:t}$ representing $t-m$ training observations in K dimensions and Θ is the matrix set concatenating $\boldsymbol{\theta}_i$ for $i = m:t$. The dimensions of $\boldsymbol{\theta}_t$ are $1 \times M$, for M network centres, and W is of size $M \times K$. Repeating this process for different delay lengths, m, performs the same role as the PACF and allows for m to be fixed.

With the deterministic signal model characterised, the noise process can be assessed. Since $\boldsymbol{\varepsilon}_t$ is the residual noise process it seems natural to look at the squared errors:

$$E\left[\boldsymbol{\varepsilon}_t^2\right] = \left(\mathbf{x}_t - \Theta_t W\right)^2, \tag{6.3}$$

but the outliers may distort the distribution of $\boldsymbol{\varepsilon}_t$. As such, it is preferable to look at the absolute error:

$$E\left[\boldsymbol{\varepsilon}_t\right] = \left|\mathbf{x}_t - \theta_t W\right|. \tag{6.4}$$

It is obvious that $P\left(E\left[\boldsymbol{\varepsilon}_t\right]\right)$ will, in general, be non-Gaussian. In order to visualise the observed time series in a topographic way, a dissimilarity matrix must be constructed. The approach we take is to base the dissimilarity calculatation for time series around the 'distances' between the distributions. $P\left(E\left[\boldsymbol{\varepsilon}_t\right]\right)$ can be estimated in one of three ways depending on the observations:

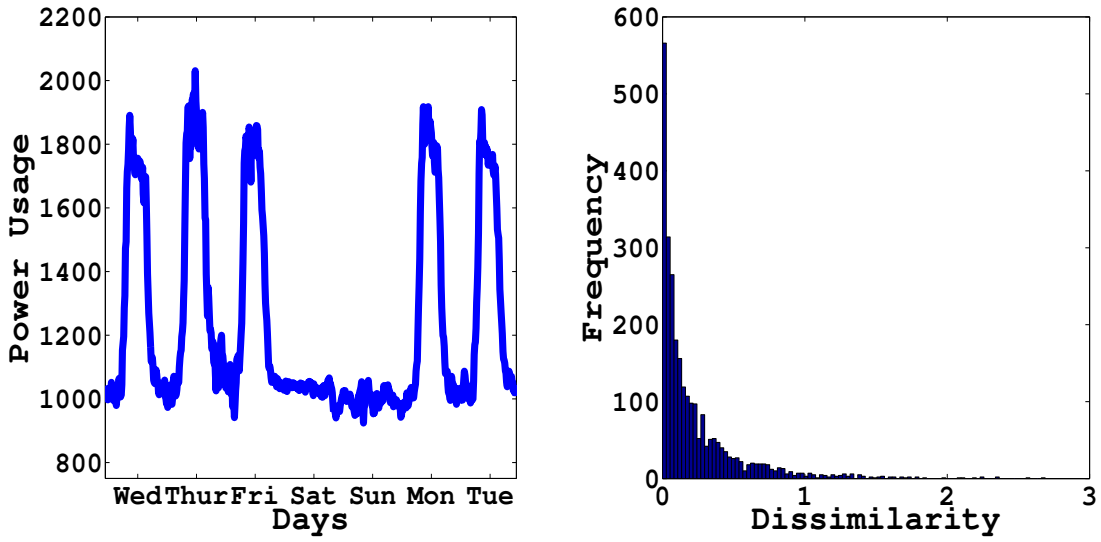
1. Univariate observations with Gamma noise, with no prior knowledge of background noise process,

2. Univariate observations with application driven noise model,

3. Multivariate observations with noise proxy (the reason for using a proxy instead of a distribution will become clear in section 6.4).

The Gamma distribution is chosen for the standard univariate observations since $E[\boldsymbol{\varepsilon}_t]$ is, by construction, greater than zero and should, for a suitably interpolated time series, be distributed close to zero. This shows why a Gaussian distribution is not appropriate and in general a Gamma distribution fits this framework. The noise proxy in the third case is used instead of attempting to estimate the residual distribution and will be properly justified in section 6.4. The following section will demonstrate the first case, using data taken from Dutch power consumption from 1997. The third case will then be used to visualise pre-seizure EEG data. Finally, the second case will be tested on real world SONAR data with a physically motivated compound mixture model. The SONAR application is an extension to the standard Residual Modelling framework since for the Dutch Power and EEG datasets we are searching for anomalies in a single signal. In the SONAR domain there are multiple signals to search across for anomalies and as such the relative changes in the predicted generator, as well as in the residuals, must be analysed. In its current form the GPLVM cannot generate a latent space from arbitrary dissimilarity matrix observations. In order to generate comparative visualisations, the dissimilarity matrices formed for the above datasets will be used to create a vector observation space, $\bar{Y} \in \mathbb{R}^N$, i.e. the dimensionality of the embedded points is the number of observations, N. The Sammon map can take these dissimilarity matrices as input, d_x, and generate a set of pseudo-observations, $\bar{Y}$ of dimension N, in a lossless way. Following this phase, the pseudo-observation vectors, in the artificial high dimensional space, can be visualised using GPLVM. As with chapter 5, GTM is deemed unsuitable for mapping high dimensional observations, particularly when the pseudo-observations are of dimension N. Other methods do not construct a latent distribution and as such do not provide an informative comparison to the proposed methods.

The figures in this chapter are to be interpreted as those from chapter 5 were. Each observation is plotted as a circular point, in this chapter shaded from observation 1 (white) to the final observation (black). The points sit upon the Uncertainty Surface



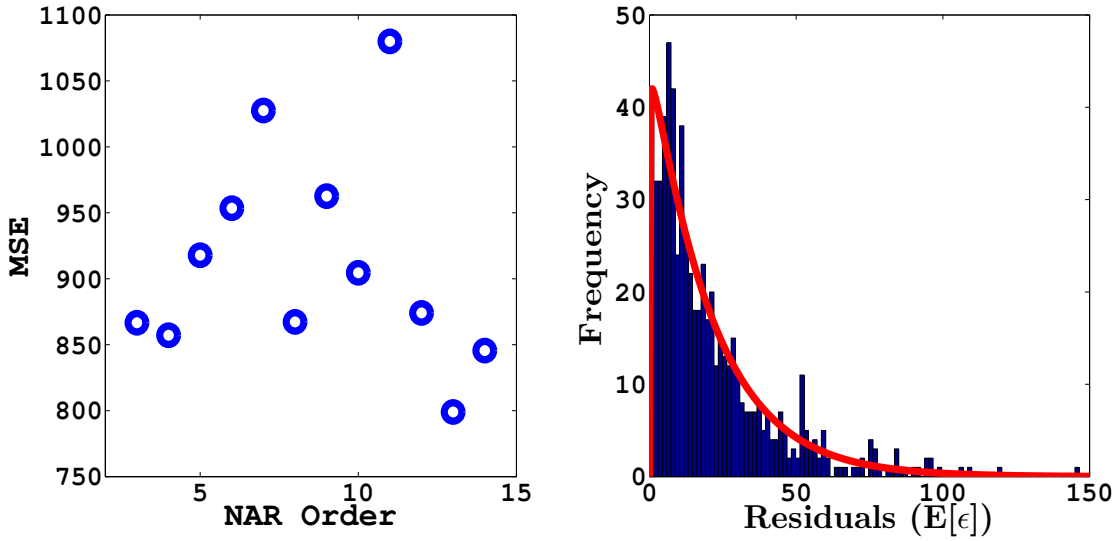
(a) (b)

Figure 6.1: (a) Sample week of the Dutch Power dataset signal and (b) the histogram of dissimilarities. The sample signal is taken from the second week in January showing typical power consumption for the weekdays (Monday to Friday) with fluctuating levels of lower consumption over the weekend. The weeks start on a Wednesday for this analysis since January 1st 1997 was a Wednesday. The histogram of dissimilarities shows the observed week-long signals are closely located in dissimilarity space, largely concentrated between 0 and 0.5.

heatmap with pink indicating high probability and blue low probability regions. The size of each point is given by the mapping uncertainty, FI_i, defined in chapter 4 for each of the models. Again, to avoid confusion between mapping uncertainty and the Uncertainty Surface, the term 'surprise' will be used instead of mapping uncertainty, FI_i.

6.3 Univariate Time Series: Dutch Power Data

The dataset taken from [78] consists of electrical power consumption measurements for 1997 in Holland. There were 96 measurements per day taken for the 365 days in the year. The visualisations will be used to identify weeks (672 samples) where the power consumption is anomalous. Figure 6.1a shows a typical week's power consumption; the second week in January. Using the framework of Residual Modelling we first need to fit the dynamics with a model, estimate the residual distribution function, and then use the Kullback-Leibler divergence between distributions to visualise dissimilarities between each week's power consumptions.



(a) (b)

Figure 6.2: (a) Nonlinear PACF coefficients for the Dutch Power dataset and (b) sample residuals for one week of observation data with Gamma distribution fit plotted in red. The MSE-minimising NAR order is 13. The residual histogram for the sample week shown in figure 6.1a is modelled with a Gamma distribution, achieving a satisfactory fit.

The deterministic signal is fit with an NAR model by an RBF network with 30 centres (similar models were trained with up to 100 centres with no change in PACF) and an '$r\log(r)$' nonlinearity. The nonlinear PACF errors are shown in figure 6.2a for $m = 2$ to 14. The MSE minimising NAR order over the typical week training period from figure 6.1a is 13, fixing m. This value does not seem to have any significant physical meaning. Following this, the weights from the state space model are fixed. Each week's observations are propagated through the RBF network such that the set of residual samples for each week, ε_t can be found. A typical week's residual ($E[\boldsymbol{\varepsilon}_t]$) histogram is shown in figure 6.2b. We assume the residuals are sampled from a distribution function which we need to identify and estimate. A Gamma distribution:

$$Gam(z|\alpha,\beta) = \beta^{\alpha} z^{\alpha-1} \left(\frac{1}{\Gamma(\alpha)}\right) \exp(-z\beta),$$

is used to characterise the residuals. This is plotted as the red line for the histogram in figure 6.2b.

The parameters of the distribution are re-estimated from the residuals each week.

The dissimilarity between observations is given by a dissimilarity between these Gamma

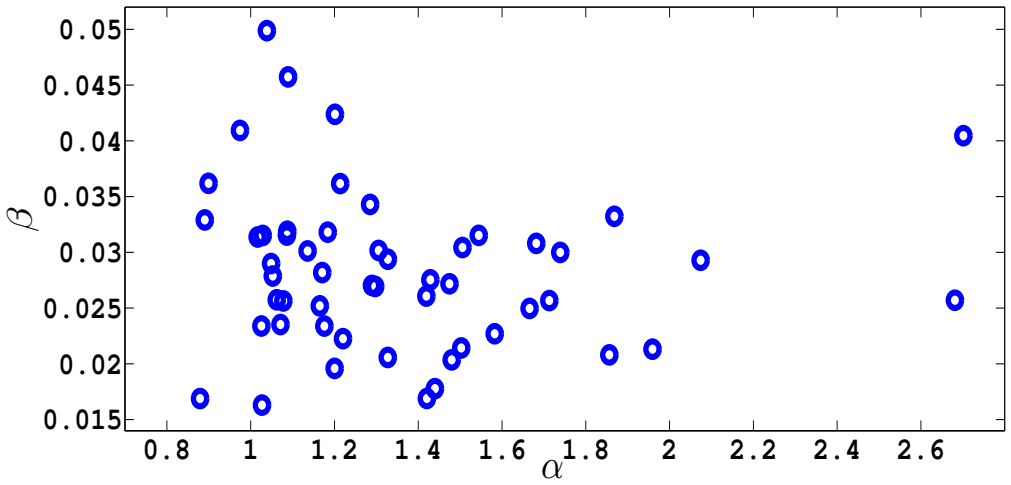


Figure 6.3: Plot of the values of α against β fit to the Dutch Power dataset. There is a clear correlation between the two variables in addition to several outliers which will be identified in the dissimilarity framework of Residual Modelling.

distributions. The Kullback-Leibler divergence is the standard choice [79]:

$$KL_{Gam}\left(P_i(\alpha_i,\beta_i)\|P_j(\alpha_j,\beta_j)\right) = \begin{aligned}&(\alpha_i-\alpha_j)\Psi(\alpha_i)-\log\left(\Gamma(\alpha_i)\right)+\log\left(\Gamma(\alpha_j)\right)+\\&+\alpha_j\left(\log(\beta_i)-\log(\beta_j)\right)+\alpha_i\left(\frac{\beta_j-\beta_i}{\beta_i}\right),\end{aligned} \tag{6.5}$$

where α_i, β_i represent the Gamma parameters for the residuals of week i. The resulting dissimilarity matrix is shown in figure 6.4.

Each week has an inherent observation uncertainty; given by the variance of the Gamma distribution:

$$Var(z) = \frac{\alpha}{\beta^2},$$

which will be used to generate isotropic Gaussians in the 2-dimensional visualisation spaces. A neighbourhood size of $k = 5$ creates a fully connected graph and is therefore fixed for PLLE, PIso and PWNM. The resulting visualisation spaces are shown in figures 6.5 and 6.6.

Separately from this work, in [80], the HOT-SAX anomaly detection algorithm identifies three irregularities:

- Week 20, Ascension Thursday on 8th May,
- Week 13, Easter Sunday with adjoining bank holiday,

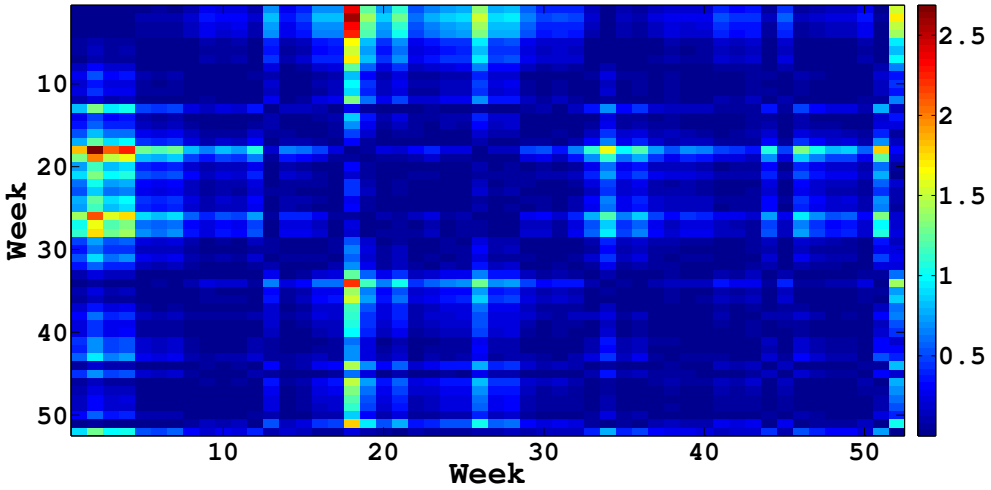
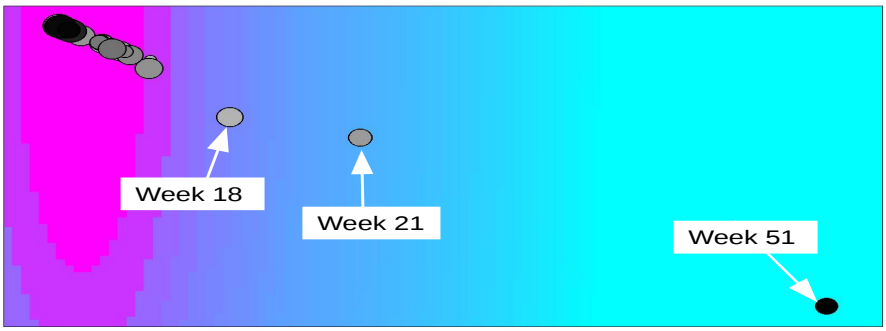


Figure 6.4: Dissimilarity matrix for the Dutch Power dataset. Weeks 13,20 and 51 are highlighed as moderately anomalous. In addition to this, weeks 18, 19, 21, 26 and 27 are indicated as very dissimilar from the remainder of the dataset. The matrix is not symmetric since the dissimilarity measure over the residuals, KL_{Gam}, is not symmetric.

- Week 51, Christmas period.

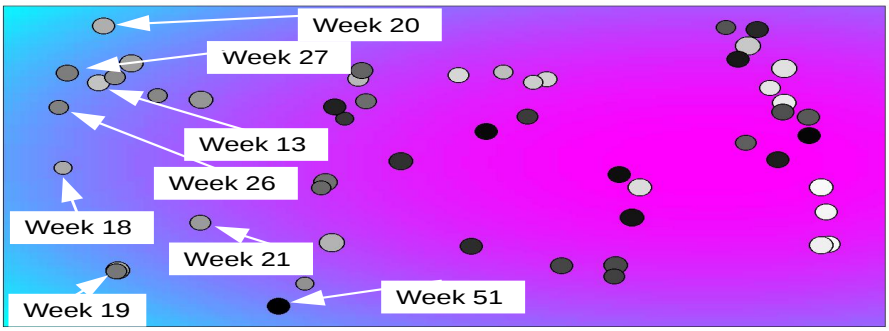
The visualisations identify these three periods and also indicate other weeks as more anomalous due to a lower weekend power usage in addition to the low power consumption over the non-working weekdays above.

The N-NS and T-NS visualisations of the power consumption data in figures 6.5a and 6.5b respectively have a general cluster where the most visible points are the later weeks (black). Weeks 13 and 20 are outside of the main cluster in a microcluster. Beyond these points are the other anomalous weeks as mentioned above (weeks 18,19,21,26,27) with week 51 (Christmas) contained in the low probability region. PIso (figure 6.6a) performs similarly to N-NS and T-NS but the geodesic distances seem to overestimate the dissimilarities for weeks 34 and 46, placing them in the low probability regions seemingly without cause. In PLLE (figure 6.5c), there is much less clustering, however all weeks that are anomolous are contained in the light blue region, including weeks 13,20 and 51 as those listed above. The lack of a general cluster makes this representation less visually appealing than the others, assuming that anomaly detection is the purpose of the analysis, however, all observations can be seen as opposed to being placed atop one another as in N-NS. PWNM (figure 6.6b) finds an approximately quadratic relationship between the observations. Structures of this sort are typical in LE



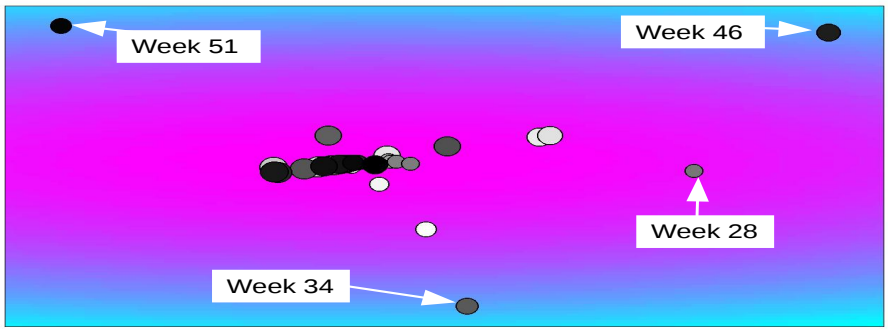
(a) N-NS

(b) T-NS

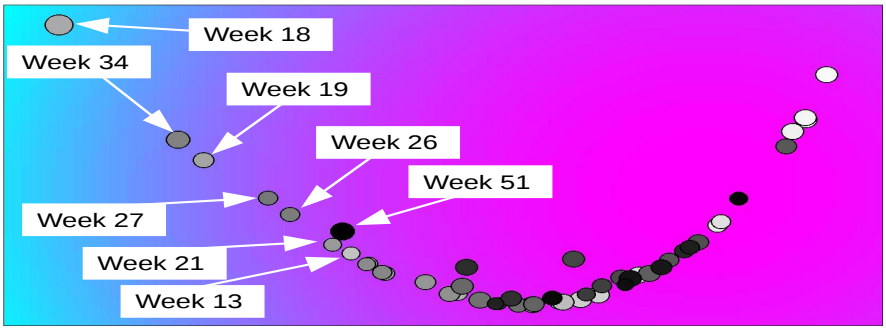


(c) PLLE

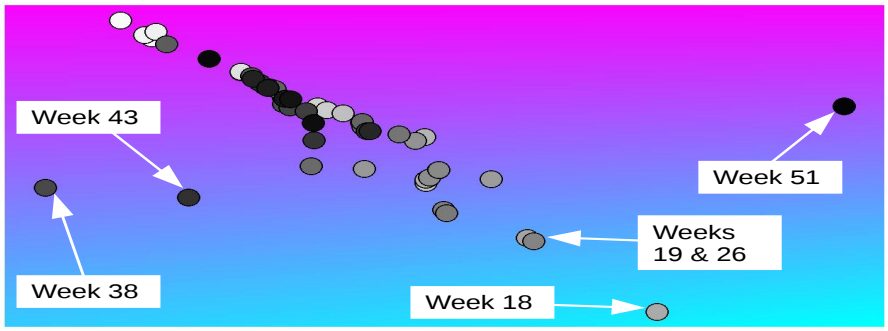
Figure 6.5: Visualisations of the Dutch data using N-NS, T-NS and PLLE. N-NS finds an approximately linear relationship over the observations with weeks 18,19,21,26,27 and 51 being identified as significantly dissimilar from the general cluster of observations. The surprise is higher for these weeks and the Christmas period of week 51 is mapped to the low probability region on the far right in N-NS. T-NS finds a similar mapping to N-NS except that weeks 16,23 and 32 are removed from the general cluster and located in the top-left of the visualisation space. These weeks are given the highest levels of surprise for no apparent reason. PLLE (mapped with $k = 5$ neighbours) provides a more spread mapping than N-NS and T-NS, with the same weeks as N-NS identified as anomalies and located in the light blue region. The level of surprise appears to be fairly constant for all observations, indicating that no single weeks are poorly described by their neighbours in data space.



(a) PIso



(b) PWNM



(c) GPLVM

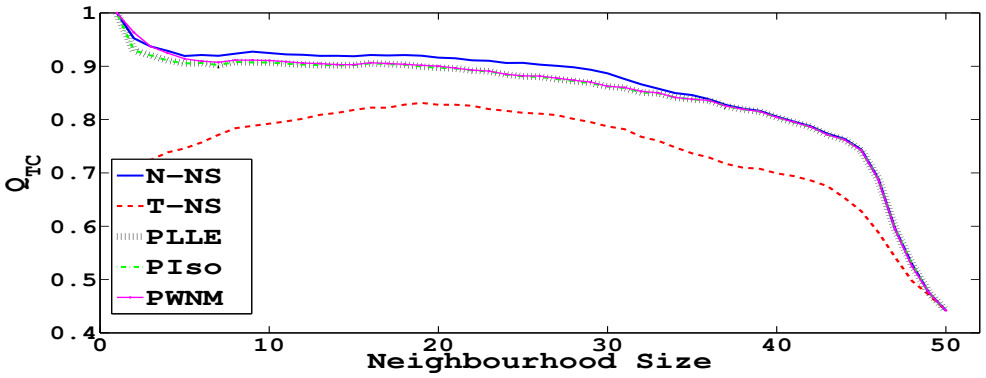
Figure 6.6: Visualisations of the Dutch data using PIso, PWNM and GPLVM. The data representation of PIso is poorer than that of all other new probabilistic mappings. Weeks 46 and 34 are placed in low probability regions on the upper-right and bottom of the visualisation space respectively. The anomalies indicated in the dissimilarity matrix of figure 6.4 are not well separated from the general cluster of weeks and given a low level of surprise. PWNM finds an altogether different mapping with an approximately quadratic relationship between observations. The anomalous weeks indicated by both low probability locations on the Uncertainty Surface and higher surprise agree with the dissimilarity matrix and N-NS, T-NS and PLLE. The GPLVM latent space identifies the weeks identified by the dissimilarity matrix as less likely than the more predictable observations. Interestingly, the Christmas period is seen as more likely than many of the more 'normal' weeks; contrary to the results of the dissimilarity matrix.

and PWNM visualisations due to a local reconstruction of the squared exponential curve. The most dissimilar observation is identified as week 18 with the other listed weeks as well as 13,20 and 51 contained within the purple region of lower probabilities of observation. The GPLVM visualisation (figure 6.6c) finds a similar set of anomalies and uncertainties as PWNM in addition to other weeks such as 43 and 38. These two weeks are also identified as unusual in the purple region on the left, again without apparent cause. The Christmas period is separated from the cluster on the right side. Each visualisation appears to be a useful representation of the data highlighting anomalies differently, but largely in agreement on which weeks' consumptions are unusual.

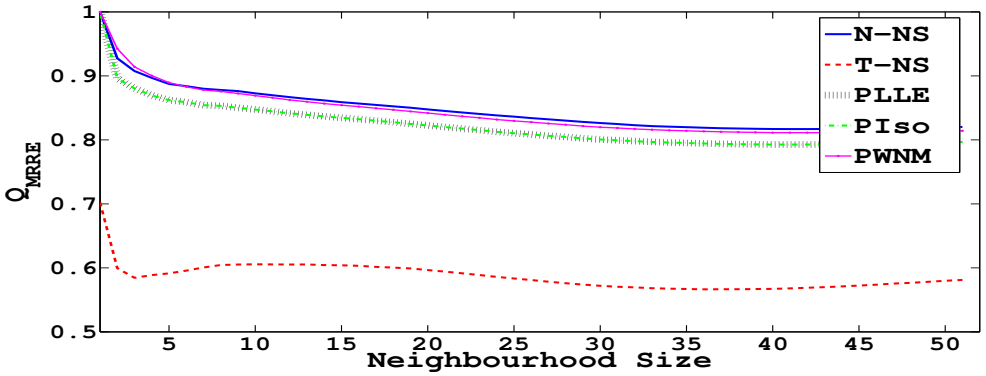
The Q_{TC} results in figure 6.7a show the mappings are largely the same, except those for T-NS, in terms of trustworthiness, achieving a good mapping up until a neighbourhood size of 44. The decay for the last eight weeks is caused by the mapping of the eight anomalous weeks described above; 18,19,21,26,27 found with Residual Modelling and 20,13,51 found by HOT-SAX as well as Residual Modelling. N-NS slightly outperforms the other algorithms for medium sized neighbourhoods. The similar mapping of T-NS performs poorly compared to the other algorithms. The visualisation space appears similar to that of N-NS but the observations placed above the general cluster are not contained in the same microcluster as in N-NS, causing increases in the number of neighbourhood leavers. This is also likely the cause for the drop in Q_{MRRE} seen in figure 6.7b. The other algorithms perform well, achieving a relatively constant rank error from neighbourhoods of size 20 and greater. The same situation is seen in the LCMC results from figure 6.7c except that the quality decays steadily for neighbourhoods greater than 10. Overall N-NS outperforms the other algorithms in terms of visualisation quality, whilst achieving an informative representation of the data. The surprise levels are relatively flat, but the anomalous weeks are still highlighted.

6.4 Multivariate Time Series: EEG Seizure data

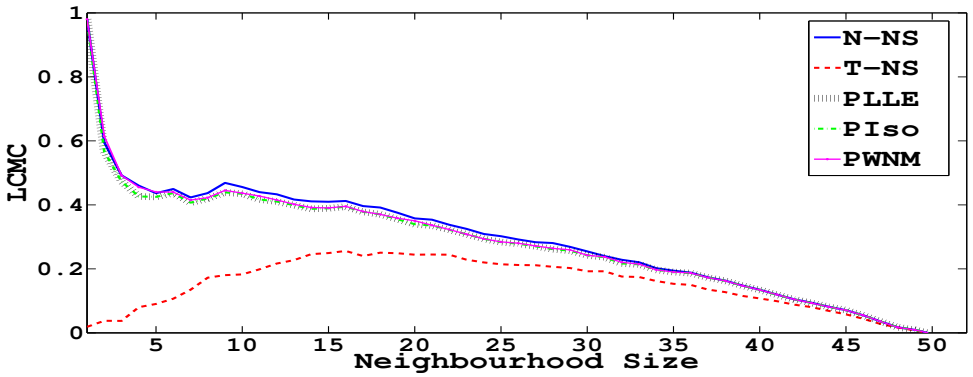
Electroencephalography (EEG) recordings were taken from [81] for the purpose of multivariate Residual Modelling. Three separate hour long segments of the dataset for



(a) Q_{TC}

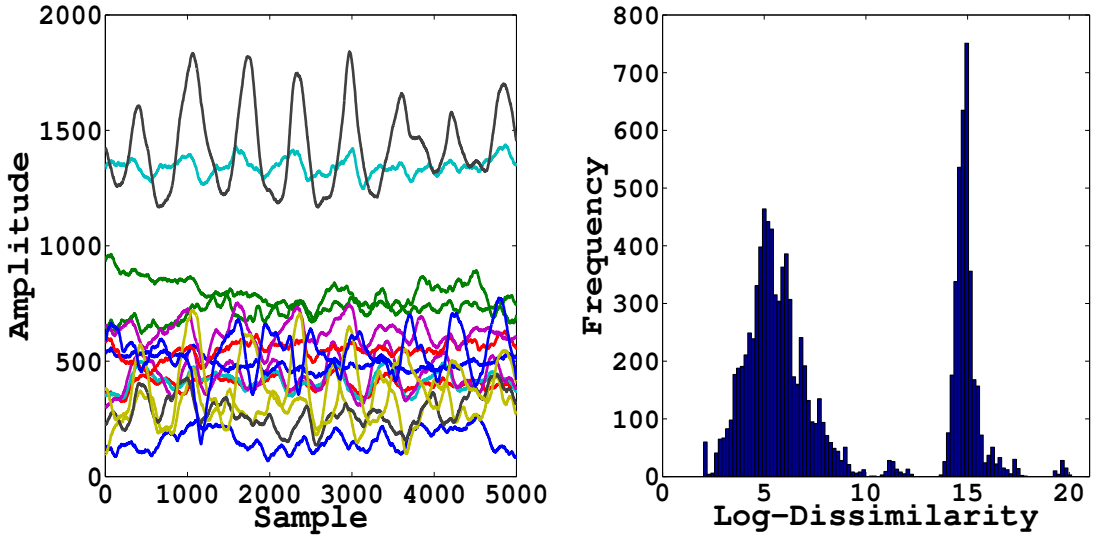


(b) Q_{MRRE}



(c) *LCMC*

Figure 6.7: Quality criterion for visualisations of the Dutch Power data: (a) Q_{TC}, (b) Q_{MRRE} and (c) LCMC. The trustworthiness of the mappings is high for the dataset until the decay for the final eight neighbours of the dataset. These neighbourhoods definitely include all weeks defined as anomalous by the Residual Modelling process. As such, their placement in the visualisation space is a difficult task. Q_{MRRE} is approximately constant beyond neighbourhoods of 20. The LCMC decay is steady beyond neighbourhoods of ten, likely due to the close proximity of many observations mapped to a general cluster as in N-NS, T-NS and PIso. T-NS performs worst over the dataset, most likely due to the placement of the points above the general cluster. The best global performance is seen in the N-NS visualisation, however the PWNM performance is similar to that of N-NS with better Q_{TC} and Q_{MRRE} for neighbourhoods smaller than five. These results are interesting given the very different visualisations generated.



(a) (b)

Figure 6.8: (a) Sample of observed signals for the EEG dataset and (b) histogram of dissimilarities. The signals for each of the 15 dimensions are plotted as different colours and show a varying dynamic structure over each observed dimension. The dissimilarities between each observed minute are multimodal with most of the dissimilarities concentrated between 2.5 and 10 and a tighter cluster centered at 15 relating to the comparison between all normal and anomalous observations.

one patient are used for visualisation in this section. The first hour segment was used to train the Residual Modelling RBF interpolator and the RBF networks for visualisation. The remaining two segments are a separate uneventful hour and a pre-seizure hour of observations. The pre-seizure data consists of the sensor recordings for 65 to 5 minutes prior to the seizure episode. The purpose of the dataset is to develop new early warning seizure prediction algorithms. The recordings were made using a 15 lead EEG with a frequency sampling rate of 5000 samples per second. Successful visualisations will highlight the third segment as being anomalous, preferably early on in the recording. An example of the time series taken from the initial training segment is shown in figure 6.8a. The 15-dimensional time series is fit with a nonlinear vector auto-regressive (NVAR) process using an RBF network. The nonlinearity used is again '$r\log(r)$' with 150 centres; training the weights on 20 minutes worth of data. Models were also fit with 250 and 500 centres with no significant change in MSE or in PACF coeffecients, except for the increase in computational complexity. The nonlinear PACF coefficients are shown in figure 6.9 and a NAR order of five is chosen to minimise the MSE. The parameters of the

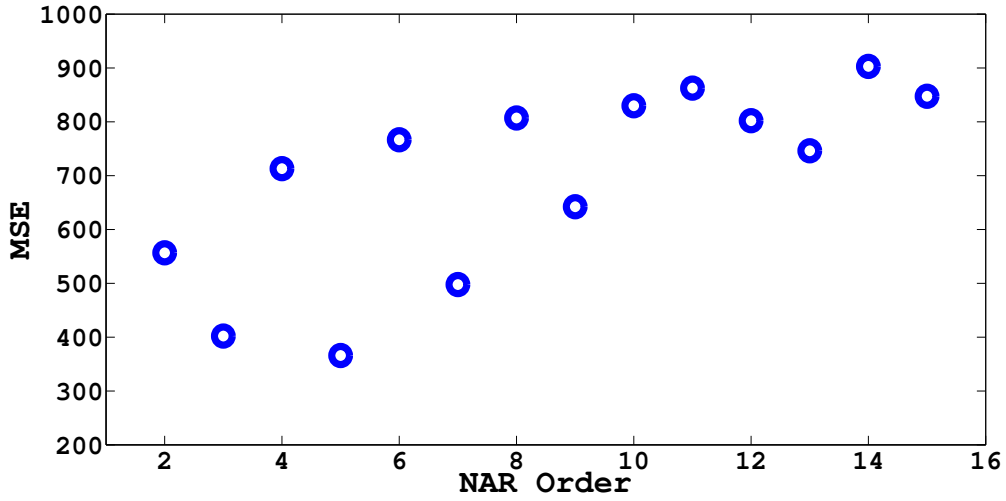


Figure 6.9: Nonlinear PACF errors for the EEG dataset. An RBF NAR order of five minimises the MSE over the training dataset. Larger NAR orders cause increasing MSE which indicates overfitting.

RBF are then fixed. The residuals, $\boldsymbol{\varepsilon}_t$ can be defined as in equation (6.4), however specifying noise distributions in a multivariate case is not the same as for the univariate case in the preceding section.

Estimating probability distributions in high dimensions is not a straightforward task, as described in [82], therefore a proxy for a noise model was introduced in [28]. Since anomalies in the model fit behave differently for different dimensions the residual sample covariance matrix, S, contains the required information:

$$S = \frac{1}{(N-m)} (X - \Theta W)^T (X - \Theta W), \tag{6.6}$$

where Θ is the concatenated matrix set of $\boldsymbol{\theta}$ vectors, W the weight matrix and X the matrix set of observations. The dimensions of X is 4995×15 for each second (given by the frequency, 5000, less the delay length, 5), Θ is 4995×150 (given by the 150 centres) and W is 150×15 (given by the 150 centres and the 15-dimensional observations). Each multi-sensor observation can now be described by a stationary, or quasi-stationary, covariance matrix, S, of dimensions 15×15. For the EEG data, the NVAR models each minute by a non-overlapping window of 5 time series samples per dimension, creating 60 covariance matrices per hour long data segment. For this learning task the RBF extensions from chapter 4 will be tested. The first normal dataset will be used to train the

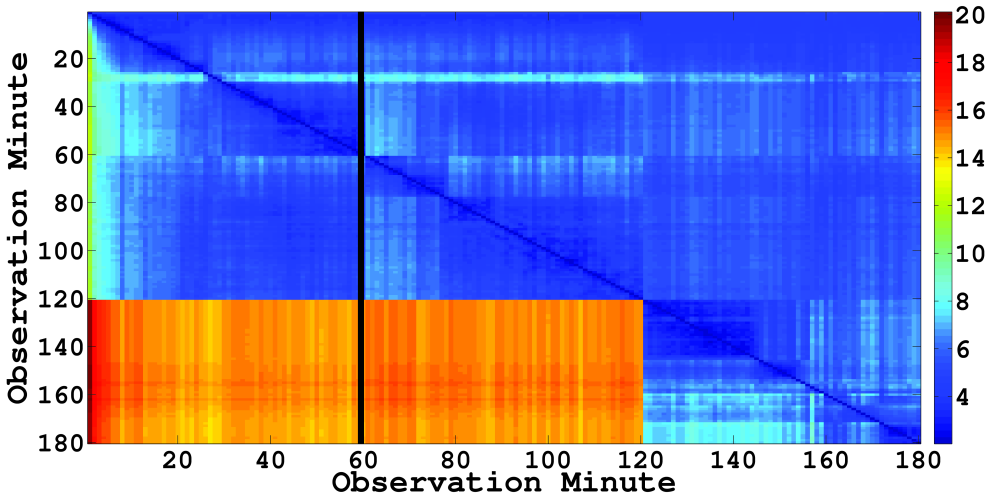
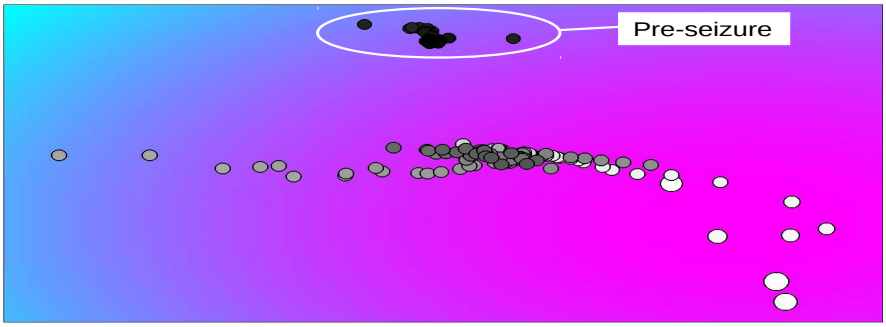


Figure 6.10: Dissimilarity matrix for the EEG dataset. The data used in the RBF interpolation over new data is the leftmost 60×180 thin matrix, separated by the black line. Large dissimilarities are observed when comparing the dissimilarity between the normal segments (observations 1 to 60 and 61 to 120) and the pre-seizure segment (observations 121 to 180).

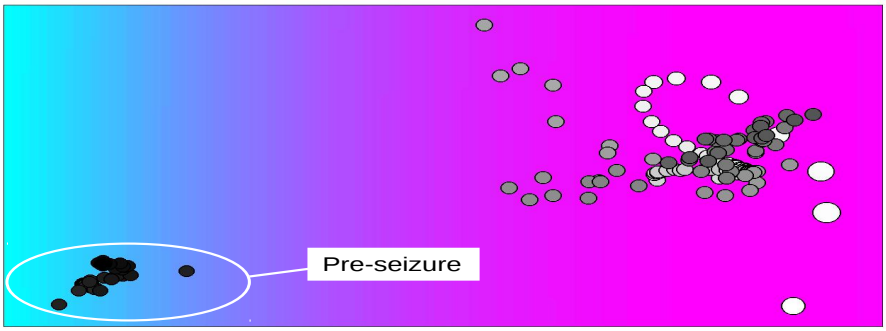
visualisation RBF networks then, using these mappings in feed-forward mode, the other normal and pre-seizure segments will be interpolated (or potentially extrapolated). Visualisation of the three sections reduces to creating an RBF mapping for visualising the relative dissimilarities between the 60 covariance matrices. Following this, the next 120 covariance matrices from the two test data sections were propagated through this network. The dissimilarity between covariance matrices can be computed as in equation (3.10). For this dataset the dissimilarties are concentrated in entirely disjoint regions of dissimilarity space, presenting a problem for visualisation algorithms. These disjoins can force the mappings to sit multiple latent means atop of one another. To avoid this, a logarithmic transform can be imposed upon $d(X_i, X_j)$. The resulting histogram is shown in figure 6.8b and the dissimilarity matrix is shown in figure 6.10.

The neighbourhood graph is connected at $k = 8$ and the degrees of freedom are indicated by the observation dimensionality ($\nu = 15$), fixing these values for the relevant mappings. Visualisations generated from the training set (plotted as white circles) with test sets (light grey for normal and dark grey/black for pre-seizure) superimposed for comparison are displayed in figures 6.11 and 6.12.

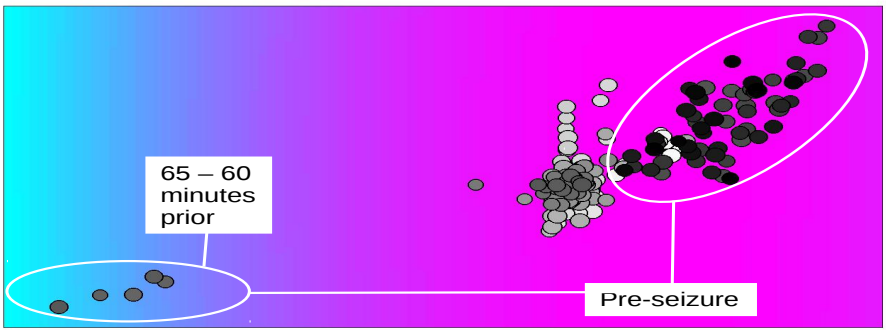
All mappings place the original normal training data in the high probability areas of the



(a) N-NS

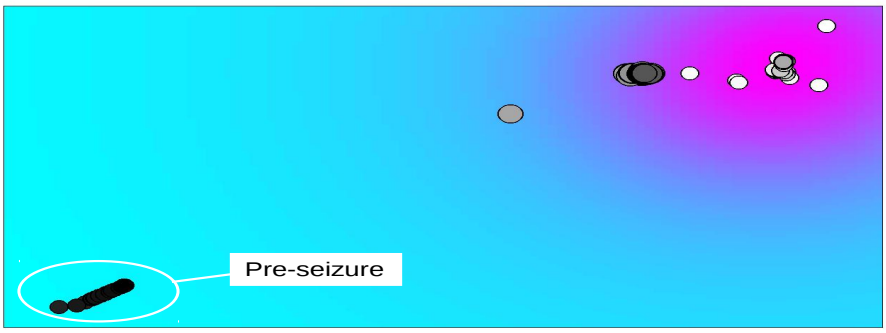


(b) T-NS

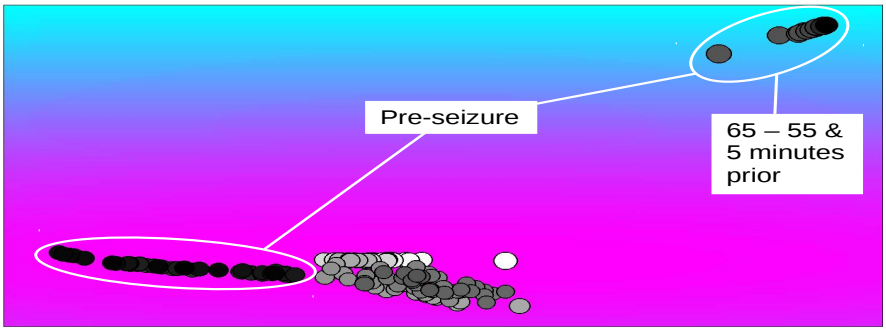


(c) PLLE

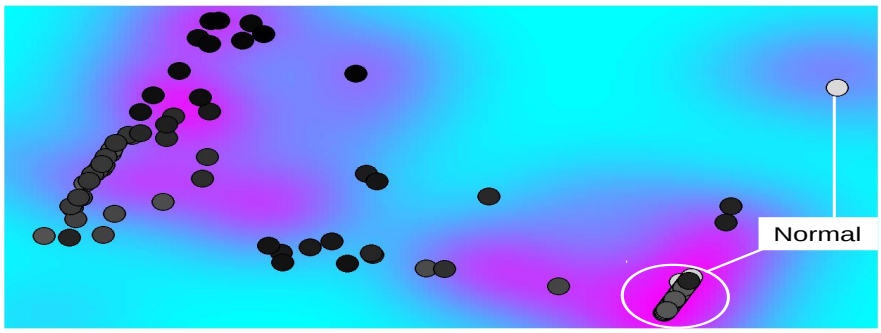
Figure 6.11: Visualisations of the EEG data using N-NS, T-NS and PLLE. The points represent one minute of data with 60 white (representing trained 'normal' data), 60 light grey (representing interpolated 'normal' data) and 60 dark grey/black (represnting interpolated pre-seizure data). N-NS and T-NS both map the normal datasets to higher probability areas of the Uncertainty Surface and identify the pre-seizure segment as anomalous. The surprise in both cases highlights the initial observations as these have precise covariance structures and average levels of dissimilarity with respect to the RBF centres used for the mapping. The PLLE mapping clusters the two normal segments and maps 65-60 minutes prior to the seizure to the low probability region of the Uncertainty Surface. The pre-seizure segment is mapped with larger spread however, it is placed in the higher probability region of the visualisation space.



(a) PIso



(b) PWNM

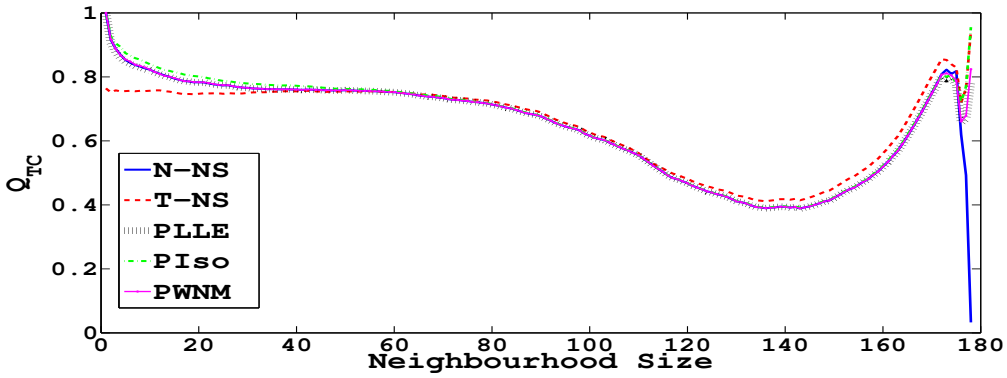


(c) GPLVM

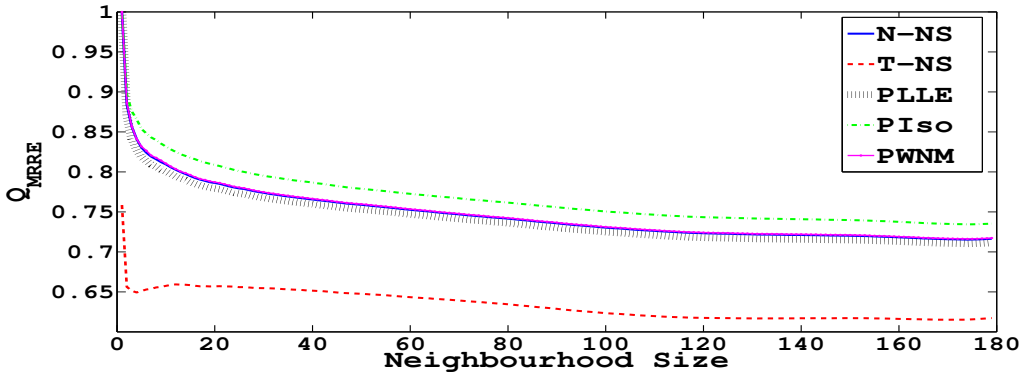
Figure 6.12: Visualisations of the EEG data using PIso, PWNM and GPLVM. PIso maps the three EEG data segments to tight distinct clusters, often placing multiple observations atop of one another. The highest surprise is found by the second normal segment, contrary to all other visualisations. The pre-seizure segment is all mapped to the low probability region which, although potentially correct for the dissimilarity matrix, the anomalies within the segment are not clearly identifiable. PWNM finds a higher level of similarity between the two normal data segments than the general pre-seizure segment. Minutes 65-55 and 5 preceding the seizure are also mapped to the low probability region. These observations are also highlighted as surprising. The GPLVM latent space has tight clustering of the first two segments, similar to PIso, with a single unexpected outlier and a much larger degree of spread given to the pre-seizure segment.

Uncertainty Surface, as expected. There is some separation from the main cluster of training data to the first set of observations, made obvious in the N-NS, T-NS and PWNM mappings of figures 6.11a, 6.11b and 6.12b respectively. The normal test set is placed within the higher probability areas (pink/purple) in all mappings, close to the training data as expected. T-NS and N-NS find more unusual structures than the clusters in the other mappings as the observations appear to be moving in a circular (T-NS) or curved (N-NS) trajectory. N-NS, T-NS and PIso map the pre-seizure data to an entirely disjoint low probability region as desired. PLLE and PWNM map the pre-seizure data to a separate region of the high probability area on the Uncertainty Surface, visually identifiable as different to the 'normal' data segments. The larger separation between anomalies in PIso is caused by the geodesic distances artifically increasing the dissimilarities. The surprise indicates the initial observations as unusual, possibly due to unusual patient activity in their 'normal' EEG segment or to overtraining in the NVAR signal model. PWNM is the only mapping which identifies the anomalies as having a high surprise, particularly the 65-55 and 5 minutes prior to the seizure. On the other hand, the other mappings find the anomalies to have low surprise, but sit in the low probability area of the Uncertainty Surface. All of the mappings created can be considered as good visualisations as they are informative and highlight the pre-seizure data, despite the challenging interpolation task that has been used. The GPLVM mapping was trained on all observations and as such is not performing the same interpolation task of the other mappings. The tight clustering of the two normal segments (apart from the unusually placed observation in the top right) are somewhat similar to the clustering effect seen in PIso. The pre-seizure data is spread throughout the latent space, largely mapped to high probability regions of the posterior probability surface. This spread and latent probabilities of the pre-seizure data run contrary to the self-similarities of the dissimilarity matrix, in particular the lower left corner of figure 6.10.

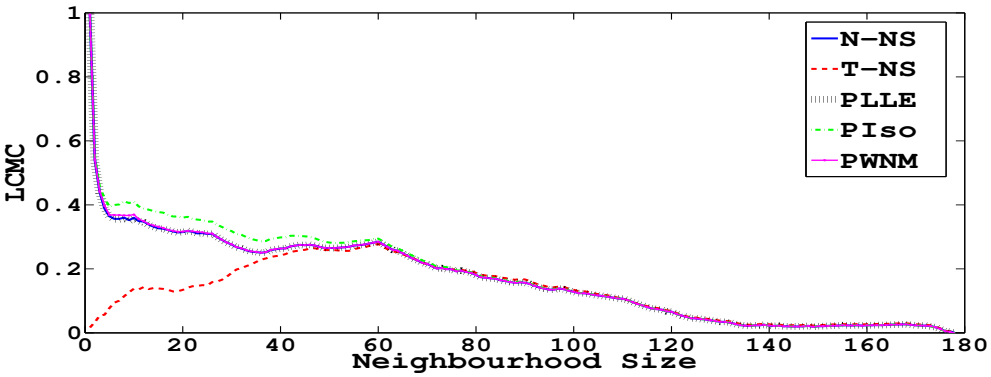
The Q_{TC} rankings for the visualisations shown in figure 6.13a show almost identical performance for all algorithms over all neighbourhoods. T-NS has a lower trustworthiness over local clusters however, the interpolation to new points in neighbourhoods greater than 60 is slightly better than the other mappings. N-NS has a



(a) Q_{TC}



(b) Q_{MRRE}



(c) *LCMC*

Figure 6.13: Quality criterion for visualisations: (a) Q_{TC}, (b) Q_{MRRE}, (c) LCMC. The quality criterion for the EEG dataset presents some unusual results not seen on other datasets in this thesis. The trustworthiness of the mappings is good for the within segment quality (up to 60 neighbours) and decays up to neighbourhoods of 140 following which it increases up to 180 for all mappings except N-NS. Within this increasing section of Q_{TC}, T-NS outperforms the other mappings. Surprisingly, in terms of Q_{MRRE} and LCMC, PIso performs the best despite the tight clusters and uninformative visualisation space. Considering the mappings were generated by training the RBF networks on the first 60 minutes of observations and extrapolating to the other two segments, the quality of the visualisations is overall very high.

rapid decay when all observations are considered, indicating the final neighbours are not well preserved in the mapping. The trustworthiness of all mappings over the dataset is surprisingly high, comparable to other datasets considered in this thesis. PIso outperforms the other mappings in terms of Q_{MRRE} (figure 6.13b), all of which achieve a steady decay in rank error from a neighbourhood of 10 onwards. Despite being worse than the Q_{MRRE} on other datasets, the results are still good when the interpolation of unseen data is taken into account. The T-NS mapping performs much worse than the other mappings, indicating the circular trajectories have not caused a good rank reconstruction of the observations. The same results are apparent in the LCMC criterion (figure 6.13c) with a rapid decay from neighbourhoods greater than 60.

This dataset provides an example of where the visualisation space and quality measures must be used in conjunction to justify which mapping is 'best'. Given the above mappings and similar performance of the mappings other than T-NS and PIso, the PWNM mapping offers the most informative representation of the data. Due to the good trustworthiness and rank-based results it can be concluded that the GPLVM mapping is a poorer representation of the observations than the probabilistic methods introduced in this thesis. The quality criterion for the proposed algorithms show the robustness of the learned mappings. There is no significant performance decay beyond the training data sizes (60) when compared to other datasets and as such the RBF extensions from chapter 4 appear to be reliable.

6.5 Univariate Time Series & Noise Model: SONAR dataset

The simple univariate framework from section 6.3 is not always applicable to real-world observations. The cause of this is the assumed Gamma distribution over residuals. In [83] samples were compared in a dissimilarity framework using their histograms, but this can be very sensitive to outliers. Kernel density estimation can be used to smooth out this effect, however the kernel-based distributions are often still too simple, or too generalised to optimise for the characterisation of residuals in, for instance the SONAR domain. GMMs provide a flexible framework for describing arbitrary distributions and

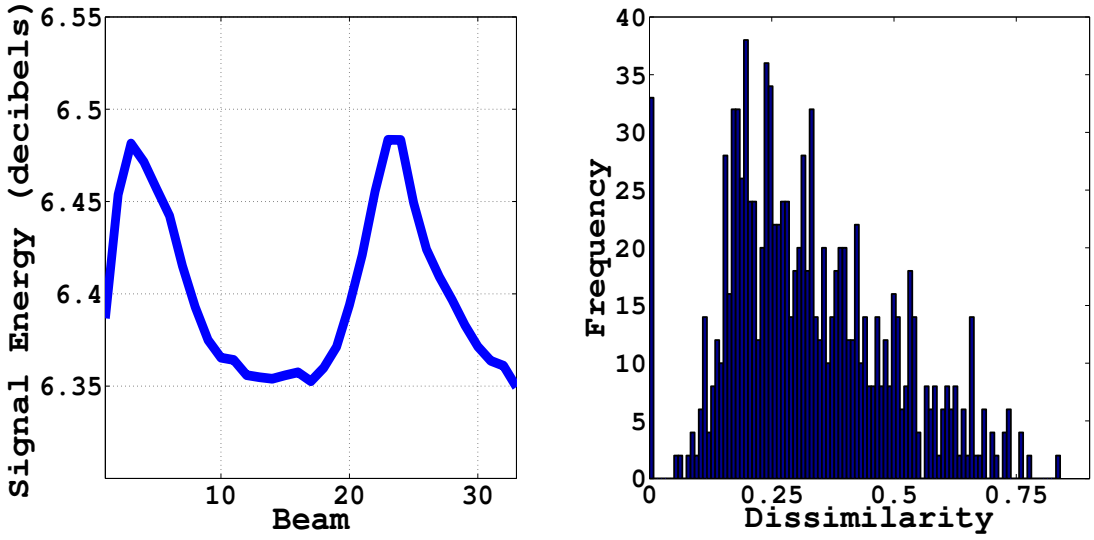
can be compared using equation (3.26), but the order of the GMM is not always straightforward to choose. Dirichlet-based schemes can assist in tuning the model order, but these can require Gibbs sampling [84], such as in [85], or a MAP approximation [86]. These points aside, it is preferable in the Signal Processing field to use a realistic noise model wherever possible. This section will outline one such model for the SONAR domain.

The SONAR dataset used in this section was supplied by the Defence Science and Technology Laboratory (Dstl). It consists of a 32-hydrophone recording from a line array in Portland bay. These 32-hydrophone time series are then linearly beamformed, creating 33 beams (artificial sensors), as is typical in RADAR and SONAR systems, for analysis. The sampling rate is 394 Hz and the weather conditions were all fine with only wave backscatter and some surface ships present. During the recording a speedboat travelled parallel to the array repeatedly with another ship intermittently present. In this thesis the time series analysis and visualisation will only consider one second of data, but the methods used can consider longer time segments or extrapolate to continuous data analysis as required. In order to summarise the activity across the beams, figure 6.14a shows the total signal energy over the period of time we will analyse:

$$Energy(K) = \sum_{t=1}^{T} (x_t^K)^2,$$

where x_t^K represents the observed time series in beam K at time t. From this plot it is clear there are two contacts centred in beams 3 (unknown ship) and 21 (DSTL exercise ship). These two contacts have similar observed energies and similar, but not identical, signal characteristics.

In the SONAR domain individual beams are visually compared by human operators using time-frequency plots called LOFARgrams (a rotated Spectrogram) in order to analyse and localise separate contacts. In keeping with this single-beam analysis framework, the observed time series are treated as 33 univariate observations to be visualised instead of a 33-dimensional observation as in section 6.4 (this approach was



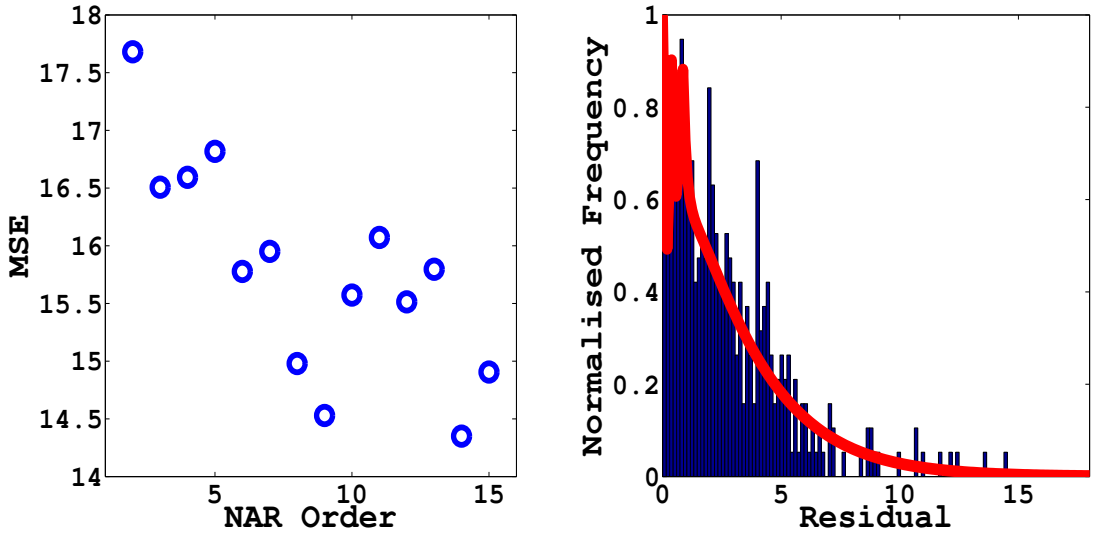
(a) (b)

Figure 6.14: (a) The SONAR signal energy in observation space and (b) histogram of dissimilarities. The high signal energies in beams 1 to 5 and 21 to 25 indicates that two contacts are present. The histogram of dissimilarities shows a wide spread of observations since $0 \leq d_{ij} \leq 2$ with only the self-dissimilarities (d_{ii}) being zero.

implemented in a deterministic visualisation scheme in [28]). Following the first step of the Residual Modelling process, a signal model is fit to the time series prior to the residual characteristication. A nonlinear model is required to properly characterise the time series and as such a NAR model is fit to the data. The NAR RBF uses 30 centres with a thin plate spline, '$r^2 \log(r)$', nonlinearity. The nonlinear PACF orders are shown in figure 6.15a showing 14 as the optimal order. The nonlinearity was chosen as it outperformed the '$r \log(r)$' basis function on training data. An example of the residuals' histogram for beam 21 is shown in figure 6.15b.

A fifth-order GMM is a suitably flexible model to characterise these residuals, however a more realistic noise model from [28] will be introduced to describe the residuals. The known physical noise sources, and their respective probability distributions, can be combined to better represent the residuals from a background additive noise perspective. A fifth-order compound mixture model is used for this task, consisting of:

1. Extraneous signals - Laplace distribution. Any prominent signals not well fit by the NAR model should appear in a small area of residual space and as such the sharply peaked Laplace distribution, with parameters given by '$Laplace(x|\mu, b)$', is



(a) (b)

Figure 6.15: (a) The SONAR nonlinear PACF and (b) histogram of dissimilarities with noise mixture model fit plotted in red. The MSE is minimised by an RBF NAR order of 14, with more complex models achieving a better fit over the training data. The fit of the noise model to the histogram of residuals is satisfactory with the effects of outliers smoothed out. The sharp peaks in the zero to one range are modelled tightly by the model due to the Laplace and Normal distributions.

an appropriate choice.

2. Clutter - K and Rayleigh distributions. It is well etablished in the SONAR literature that K, with parameters given by '$K(x|\nu,l)$', and Rayleigh, with parameters given by '$Rayleigh(x|\sigma)$', distributions are suitable for describing background clutter such as biological and environmental effects in the underwater environment [87].

3. Rain - Gamma distribution. Typically in the literature the Poisson distribution is used to describe rain [88], however the other distributions in this mixture are continuous and therefore the Gamma, with parameters given by '$Gamma(x|\alpha,\beta)$', which behaves similarly to the Poisson, is chosen.

4. Remainder - Normal, with parameters given by $\mathcal{N}(x|m,s)$. Any leftover residual elements can be fit with a Normal distribution.

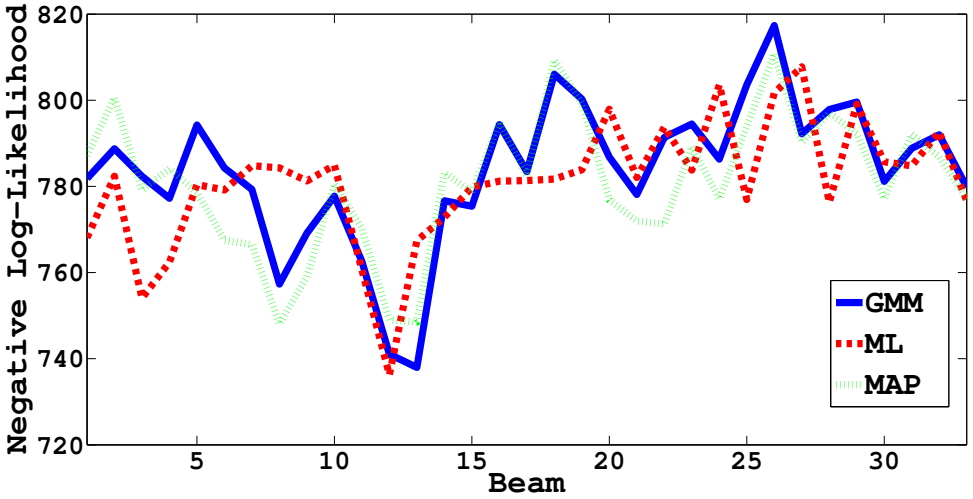


Figure 6.16: Negative log-likelihoods for the mixture model fit to one second of SONAR data. The ML fit is on average better than that of the GMM with a lower standard deviation. The MAP fit of the noise model is an improvement on the ML ensuring a smoother likelihood fit over adjacent beams.

The overall compound mixture model is given by:

$$P(x|\lambda) = \pi_1 Laplace(x|\mu,b) + \pi_2 Rayleigh(x|\sigma) + \pi_3 K(x|\nu,l) + \pi_4 Gamma(x|\alpha,\beta) + \pi_5 \mathcal{N}(x|m,s),$$

using λ to denote the set of hyperparameters. This model can be implemented in a Maximum Likelihood (ML) framework or, provided appropriate priors are specified, a more robust Maximum-a-Posteriori (MAP) scheme can be employed. In this thesis the mixture components are fit using a hybrid version of gradient descent. This is required since optimising the parameters of the K-distribution using gradient descent often leads to unrealistic parameters [89]. In order to remedy this, the parameters of the K distributions are first fit to the residuals in a Bayesian scheme introduced in [90] before fitting the remaining parameters, mixture weights and hyperparameters, in the MAP mixture case, using gradient descent over the negative log-likelihood. Appendix E describes the gradient descent procedure for the MAP scheme of the 24 parameters. Figure 6.16 shows a comparison between the negative log-likelihood of the GMM, ML and MAP of the mixtures fit.

The ML and MAP mixtures fit the residuals better than the GMM. The MAP fit is smoother and more reliable than the ML. The mixture coefficients, the distributional

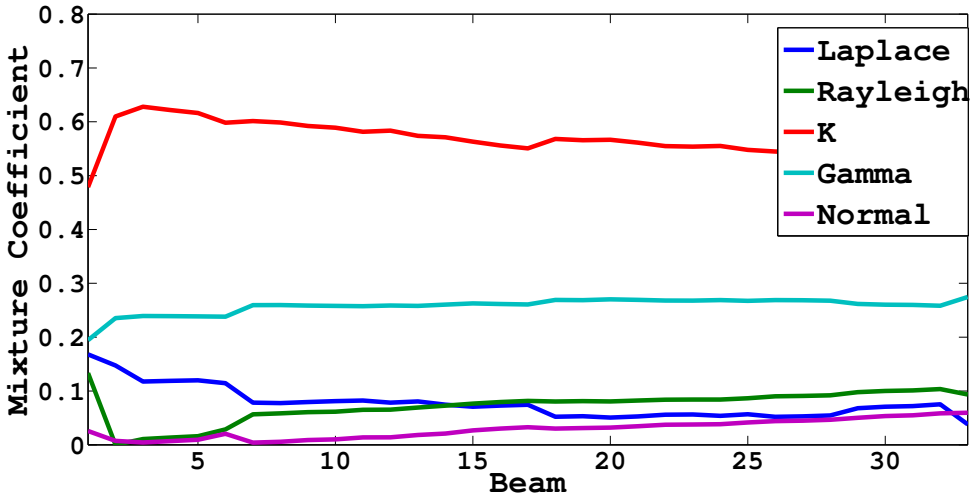


Figure 6.17: Mixture weights for the mixture model fit to one second of SONAR data. The weighting for the K-distribution is highest since the other weather conditions typically modelled by the other mixture components were not present in the dataset. The weighting of each distribution changes smoothly over the beams as expected.

weights, for the five probability distributions are shown in figure 6.17. All parameters here are fit with SCG and as such are subject to local minima. The local minima in the optimisation can force the distributions to swap mixture weights without necessarily changing the overall shape of the distribution, typically for the Gamma and K-distributions as they can have similar shapes. K-distributions are useful for characterising wave backscatter and as such is the dominant distribution for this DSTL dataset. It should be noted that the purpose of this noise model is not to infer the weather condition based on the residual distribution.

The residuals for each beam are compared using a suitable dissimilarity measure over the mixture models. The K-distribution is difficult to involve in many dissimilarity measures and other combinations, such as the integrals over Laplace and Rayleigh products, are analytically intractable. This prevents even variational bounds, for instance over the Kullback-Leibler divergence as in equation (3.26). Without the possibility of a variational approach, a sampling scheme must be considered to solve the integrals required for dissimilarity measures. As with variational techniques, a similar problem arises since inversion of the K-distribution required for many sampling schemes is not analytically tractable either. Fortunately, a biased sampling approach can be implemented, evaluating the mixtures at regular intervals as if it were a deterministic

function. The SONAR noise mixture undergoes biased sampling to generate a 1000-dimensional probability vector for each beam's mixture model. Instead of standard entropy measures, the similarity between mixtures is more important in terms of their overlap. In order to compare the overlap between distributions, the Bhattacharyya distance is used:

$$d_{BH}(i,j) = \sum_{l} \sqrt{q_i^l q_j^l}, \tag{6.7}$$

where q_i^l represents dimension l of the biased sampling vector, $\mathbf{q}_i$, for beam i. In SONAR there are a wide variety of observable signals which may cause spurious values of $d_{BH}(i,j)$. This could cause the Residual Modelling process to highlight beams with any contact present to be seen as anomalies. On the contrary, they may be unimportant merchant vessels, for instance. In the one-contact case this is useful, however as multiple contacts are often present the NAR signal model must be incorporated into the dissimilarity framework to avoid this effect. Signals in the SONAR domain are often periodic and as such analysed in the Fourier domain to distinguish the difference between contacts. The squared Euclidean distance between the Power Spectral Densities of the predicted (NAR) signals is useful for segmenting different types of targets [27]:

$$d_{PSD}(i,j) = (\mathbf{s}_i - \mathbf{s}_j)^T (\mathbf{s}_i - \mathbf{s}_j), \tag{6.8}$$

where '$\mathbf{s}_i = |\int_0^T \mathbf{x}_t^i \exp\{-j2\pi t\} dt|$' (with j the imaginary number $\sqrt{-1}$) is the Fourier vector corresponding to the observed time series x_t^i. The two dissimilarity measures from equations (6.7) and (6.8) can be combined once normalised (to be at maximum $d(i,j) = 1$) so that one does not far outweigh the other:

$$d_{SONAR}(i,j) = (\lambda_d) d_{PSD}(i,j) + (1-\lambda_d) d_{BH}(i,j), \tag{6.9}$$

where λ_d is a dissimilarity weighting parameter greater than zero, set to 0.5 in [28]. The histogram of dissimilarities is shown in figure 6.14b and the beam dissimilarity matrix in figure 6.18. Large relative dissimilarities are observed in the groups associated with the signal energy plot (beams 1-5 and 21-25 containing contacts). It is clear that the contact

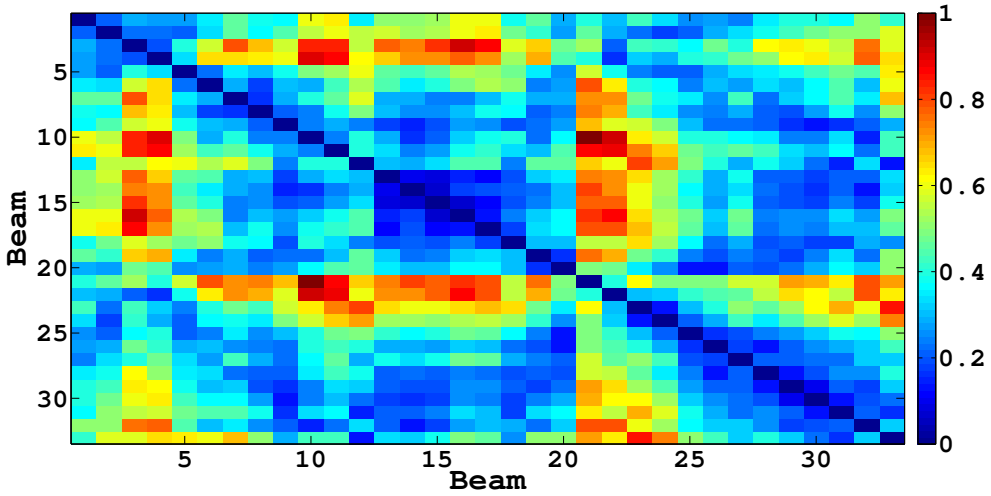
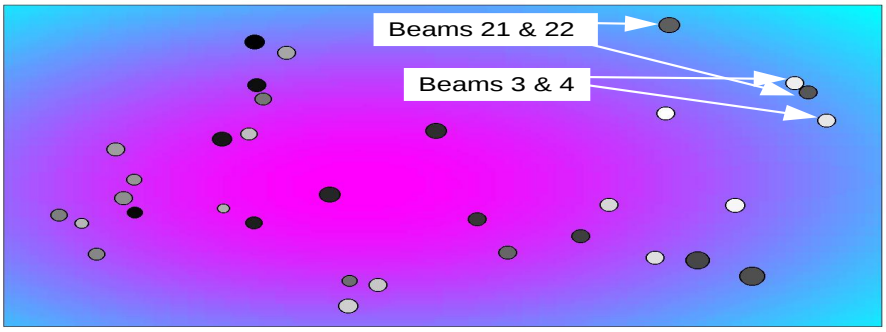


Figure 6.18: Dissimilarity matrix for the SONAR dataset. The dissimilarity matrix is symmetric due to the symmetric pairwise dissimilarity measures used. The beams identified in the Signal Energy plot (figure 6.14a) as containing contacts (1 to 5 and 21 to 25) are significantly dissimilar from the other beams. The two contacts are similar but not identical.

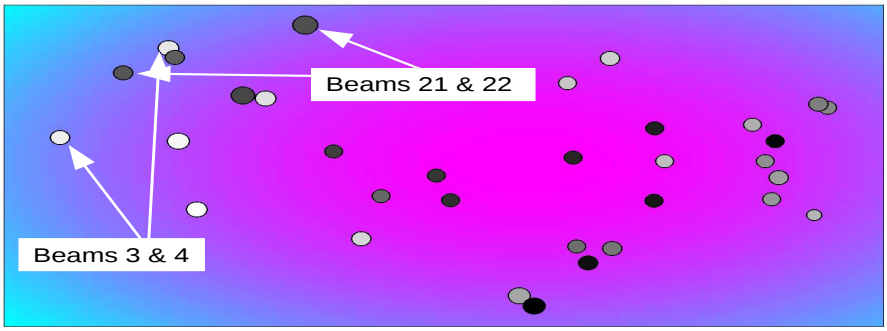
centred in beam 21 is not identical to that centred in beam 3, due to the low, but nonzero, entries in the dissimilarity matrix.

The visualisations of the SONAR dataset are shown in figures 6.19 and 6.20. Connected graphs were achieved for $k = 5$; fixing the parameter for PIso, PLLE and PWNM. This is a realistic neighbourhood parameter in the SONAR domain since a signal from a reasonably quiet target, such as a submarine, would be expected to be contained within five beams, with larger and louder contacts often visible in larger sections of the array. Since the observations are made in a 33 beam beamformed array, the degrees of freedom, ν, in T-NS is fixed to 33.

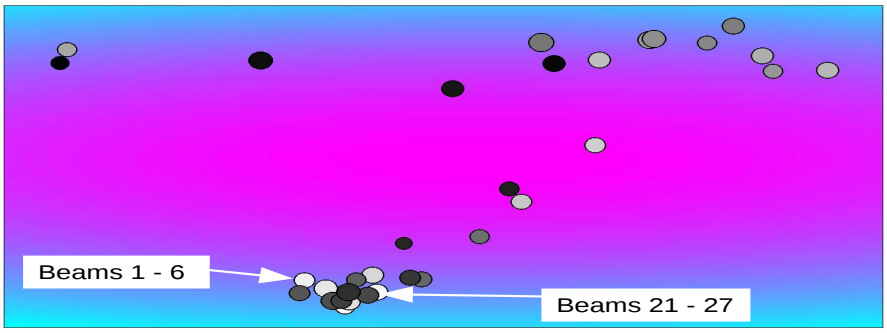
The N-NS visualisation (figure 6.19a) places the bulk of beam observations in a cluster of high probability, with beams 3 and 4 on the border of a low probability region (light blue) and the contact centred in beams 21 and 22 in the low probability area of the Uncertainty Surface. The surprise highlights this as an anomaly compared to the other observations. As such, this is considered a good representation of the data. T-NS (figure 6.19b) performs similarly to N-NS, isolating beams 3,4,21 and 22 as anomalous by locating them in the lower probability region. These observations are given a slightly higher level of surprise than the noise-only beams. PLLE (figure 6.19c) clusters both



(a) N-NS

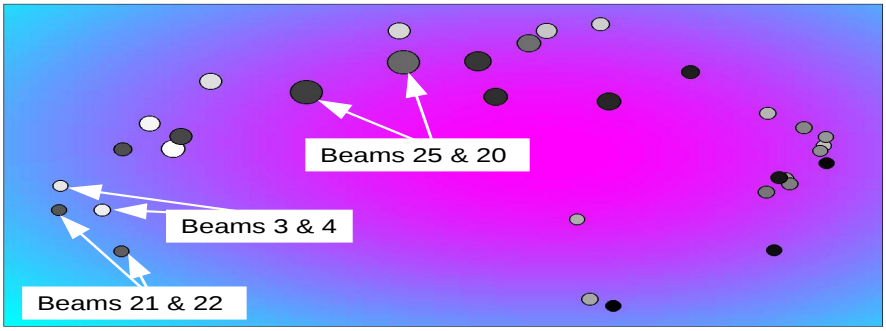


(b) T-NS

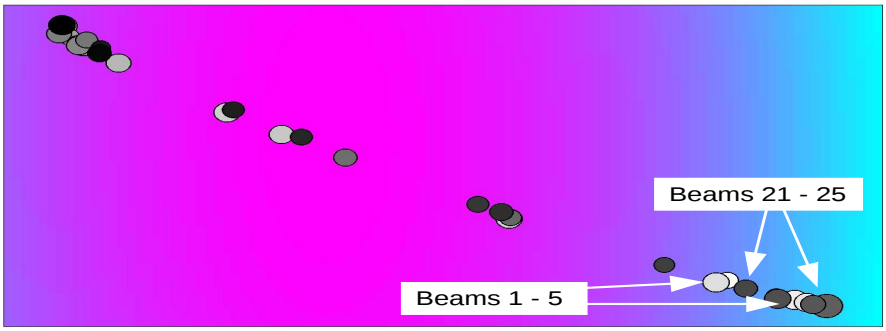


(c) PLLE

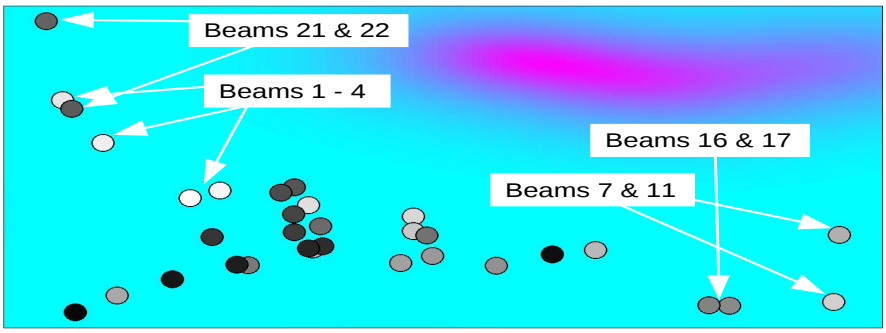
Figure 6.19: Visualisations of the SONAR data using (a) N-NS, (b) T-NS and (c) PLLE. The N-NS visualisation highlights the beams containing contacts as unusual, locating them on the right side of the low probability region. The surprise indicates these beams as more anomalous than the noise-only beams. T-NS finds a very similar mapping to N-NS. PLLE clusters the beams with contacts in the bottom of the visualisation space. The noisy beams are more spread and mapped to a slightly higher probability region of the Uncertainty Surface. The surprise levels are relatively flat, indicating a good linear reconstruction of observations by their neighbours. This clustering is a useful representation for creating summaries of the dataset, however the spread of beams as in N-NS and T-NS is not reflected here.



(a) PIso



(b) PWNM

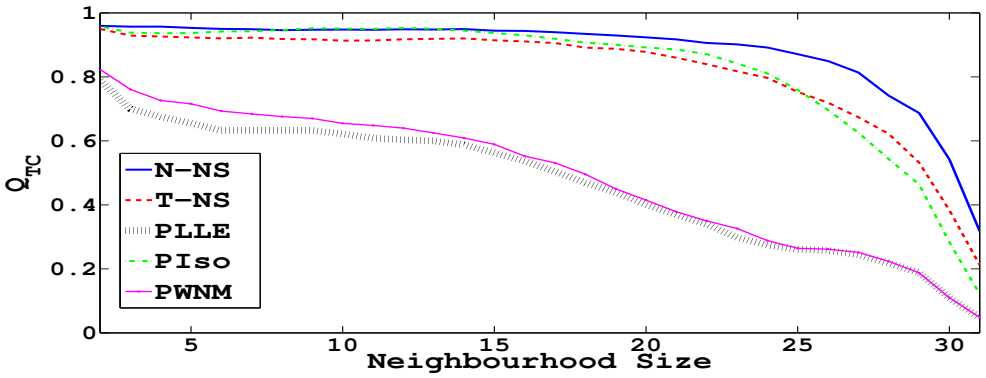


(c) GPLVM

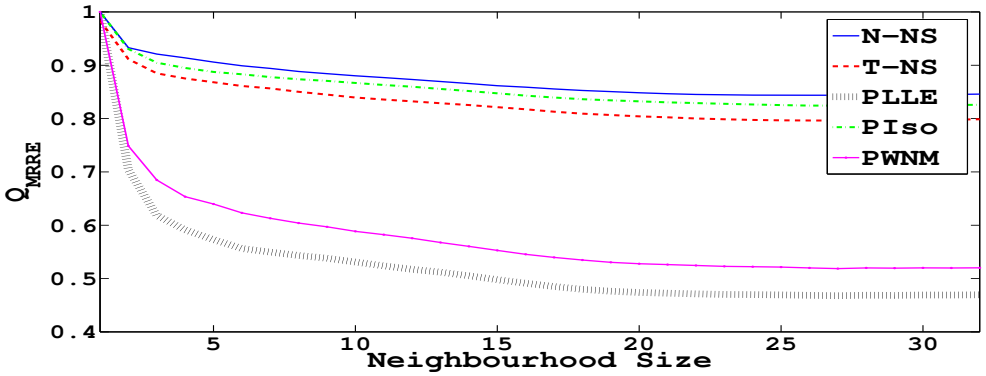
Figure 6.20: Visualisations of the SONAR data using (a) PIso, (b) PWNM and (c) GPLVM. PIso maps the observations to a curved structure with contacts placed on the left hand side and noisy beams on the right hand side. The beams with contacts are placed in the lower probability region with a low level of surprise. This is caused by the large geodesic dissimilarity between these observations resulting in a low FI_i. The PWNM mapping clusters the noisy beams and maps the interesting beams on a trajectory towards the low probability region of the Uncertainty Surface. The surprise indicates these beams as anomalous as one would expect. The GPLVM mapping locates the noisy beams in the centre and lower left and right corners. The other beams are separated from the cluster, but not easily indicated as anomalous. The posterior probability surface is misleading, assigning high probabilities to regions where there are no observations.

contacts and noise only beams together in the low probability area. The noise-only beams are less clustered but occur in a higher probability region than the signal-containing beams. The surprise here is again uninformative. PIso finds a similar visualisation to T-NS but with a more continuous, curved latent shape. The signal-containing beams leave the higher probability areas of the Uncertainty Surface on the left hand side. Higher surprise is given to beams 20 and 25; the edges of the second contact. PWNM maps the observations to a general cluster of noisy beams and locates the signal-containing beams into the low probability region. The neighbourhood size has caused the two separate contacts to be considered as almost identical, placing the means in close proximity to one another. Higher surprise here is given to beam 21, where the second contact is centred. This mapping is an interesting summary of the observation space separating signal and noise characteristics in an intuitive way. The GPLVM mapping is very spread out with the noise-containing beams in the centre and lower parts of the latent space. The remaining points represent the beams observing the contacts. The clustering here is representative of the dissimilarity matrix. However, the posterior probability surface shown is confusing and unbelievable, given that there are no observations contained within the high probability regions. It is also not clear which points represent unusual observations.

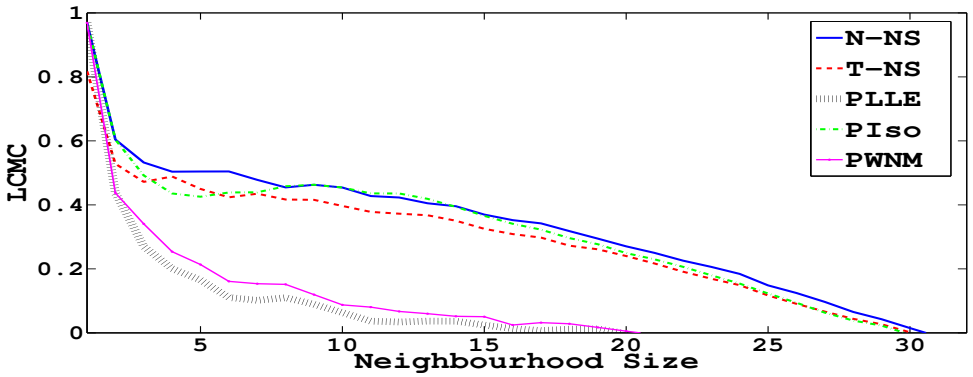
Figure 6.21 shows the quality criterion for the above visualisations of the SONAR dataset. With similar latent spaces the N-NS, T-NS and PIso results are very similar, outperforming PLLE and PWNM in all mapping criteria. The results for these three mappings indicate that the visualisation spaces are very good recreations of the observed data and therefore reliable. The large clusters in PLLE and PWNM have lowered the trustworthiness and rank errors compared to the other mappings. This is, however, an example of where the visualisations themselves must be taken into account as well as the quality criteria, since these representations are more useful for the sole purpose of anomaly detection and decision making processes involving humans.



(a) Q_{TC}



(b) Q_{MRRE}



(c) *LCMC*

Figure 6.21: Quality criterion for visualisations: (a) Q_{TC}, (b) Q_{MRRE} and (c) LCMC. The trustworthiness of the mappings is lower for PLLE and PWNM due to the clustering of contact-filled and noisy beams respectively. The performace for N-NS is better than the other mappings with a very high result for both Q_{TC} and Q_{MRRE}. The decay in LCMC for N-NS, PIso and T-NS is at a far lower rate than for other datasets used in this thesis. This is in part due to the smaller dataset used here with only 33 observations, but also due to the better separation between observations in the dissimilarity matrix. T-NS performs surprisingly well on this dataset, potentially due to the high degrees of freedom ($\nu = 33$) which allows the mapping to be less constrained than in the small ν case and generate similar visualisation spaces to N-NS. The geodesic dissimilarities in PIso allow the separation between contact-filled and noise-filled beams to be greater than in N-NS, damaging the mapping quality.

Dataset	Best	Worst
Dutch Power	N-NS	T-NS
EEG	PIso	T-NS / PLLE
SONAR	N-NS	PLLE

Table 6.1: Comparison of mapping quality criteria for time series datasets. The best mappings are found by N-NS in two of the datasets, with PIso achieving similar mappings. T-NS and PLLE achieved the worst mappings on the Dutch and EEG datasets. PLLE also achieved far worse mapping results for the SONAR dataset but the T-NS mapping was improved on this dataset, potentially due to the higher degrees of freedom used.

6.6 Overview

In this chapter the framework of Residual Modelling has been introduced for transforming observed time series into a dissimilarity framework. These time series, even when deterministic, can be visualised whilst incorporating the model and observation uncertainties.

Three distinct datasets from different fields of signal processing have been analysed with the Residual Modelling methodology. Anomalies in the Dutch Power and SONAR datasets were highlighted as expected. Additional outliers were identified in the Dutch Power dataset which were not found by the popular HOT-SAX anomaly detection algorithm. The multivariate EEG dataset was successfully analysed, predicting seizure behaviour up to one hour in advance.

From the dissimilarity matrix output of the Residual Modelling process, the probabilistic mappings from chapter 3 were used to visualise the data. The visualisation spaces were all informative, trustworthy and preserved rank well. Table 6.1 shows the best and worst mapping performance for the three datasets used in this chapter.

Surprisingly, having been the consistently average mapping quality for vectorial datasets PLLE is the worst algorithm for time series visualisation. The reason for this is the subject of further research, however it may be the case that dissimilarity matrices require larger neighbourhoods than those which create a fully connected graph.

The next and final chapter shows how the thesis' central tenet of visualisation-through-dissimilarity, can be used as a 'deep-learning' architecture. This

new method, the Cascading RBF, circumvents several problems of the traditional approach, but with a performance better than the current world-best.

7 Cascading RBFs

> ‘With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.’
>
> - John von Neumann

7.1 Introduction

Classifiers in Machine Learning such as RBFs or Multi-layer Perceptrons (MLPs) have been used successfully as black-box tools. The generalisation properties and accuracy made them particularly popular in the 1980’s and 1990’s. In the late 1990’s relational kernel models such as the support vector machine (SVM) were favoured over Artificial Neural Networks (ANNs) such as MLPs and RBFs. Their results were due to the expansion of the data space using the kernel trick outlined in, for example, [2]. This involves expanding the dimension of observations processed by these algorithms as they perform operations on dissimilarity (kernel) matrices instead of on the observations. The

concept of stacking MLPs was attempted in the literature, for example in [91], but always resulted in poor local minima. In 2006 Hinton and Salakhutdinov [92] revived the idea of 'deep learning' where each layer of a stacked, many multiple-layer architecture is pre-trained using a dimensionality reduction technique known as the Auto-Encoder (AE) [93]. In this chapter a similarly motivated approach to training a deep classifier consisting of RBFs is used. The cascaded approach uses a topographic form of dimension reduction instead of the AE approach. This new functional mapping is then tested on the MNist handwritten digits database in a classification task.

7.2 Background

We first consider the two mainstream approaches of deep learning based on the MLP and the Convnet, before introducing the new Cascaded RBF model.

7.2.1 Deep MLPs

The current approach [94] to constructing a deep MLP consists of stacking multiple AEs and a final single layer classifier. The AE relies on the use of an MLP network, typically expressed as:

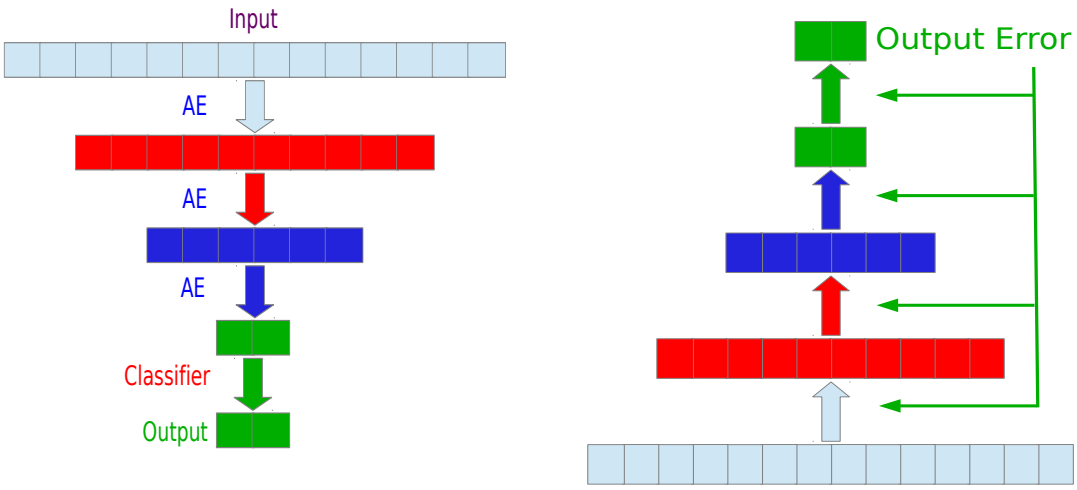
$$Y = W^2\phi(XW^1),$$

where W^1 and W^2 are learned through gradient descent of the network error. ϕ is a nonlinear activation function, typically the hyperbolic tangent function (tanh). X are the observations and Y the targets. The AE is given by the relation:

$$Y = \mathbf{f}(X), \;\; \hat{X} = \mathbf{f}(Y),$$

$$Y = W^2\phi(XW^1), \;\; \hat{X} = W^4\phi(YW^3) = W^4\phi(W^2\phi(XW^1)W^3), \tag{7.1}$$

where W^l is a weight matrix for layer l, X are observations and $\hat{X}$ the reconstructed version of X. This architecture is depicted in figure 7.1a. The weights are optimised by



(a) Pre-Training Phase (b) Back-propagation Training

Figure 7.1: Training procedure for deep MLPs. The inputs are propagated through multiple AE layers before a final layer classifier is trained to map from the final AE layer to the observed targets, creating the network outputs (left). Once pretraining is complete, the misclassification error between the network outputs and observed targets is backpropagated to each of the weight matrices in each AE layer and the classifier layer (right).

minimising the squared error:

$$E_{AE} = \sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2, \tag{7.2}$$

using gradient descent. It is clear from section 2 that this is a nonlinear extension to PCA. As such, the AE is incapable of creating a topographic map since relative dissimilarities are not preserved. The Y layer is used as a feature space descriptor of the observations. In order to prevent the weights from approaching the identity matrix, modifications to the AE such as weight decays, the Denoising AE [95] and Sparse AE [96] have been implemented.

The Y feature space can then be used as input for a classifier, in a two layer deep MLP, or another AE, in a deep MLP with greater than two layers. Figure 7.1a shows a four layer deep MLP in the pre-training phase. Once each AE is trained to map from layer to layer, the final layer is trained as a standard classifier based on minimising the MSE. These

pre-trained weights are used as the starting point for a deep MLP classifier:

$$X^0 \underset{W^0}{\longrightarrow} X^1 \underset{W^1}{\longrightarrow} X^2 \underset{W^2}{\longrightarrow} X^3 \underset{W^3}{\longrightarrow} Y,$$

Now all weights are trained to minimise the MSE on the classification task using backpropogation of the error derivatives; shown in figure 7.1b. Note that the AE does not always need to reduce dimensionality. Successful results such as in [97] where a five layer network has feature spaces with dimensions: 784-500-500-2000-30. The final layer here is a nearest neighbour classifier, but the AE training and backpropagation remain the same. Using a deep MLP reduced the misclassification error for the MNist database from 2.45% [98] to 0.83% [99].

7.2.2 Convnets

A popular network used for image classification tasks is the Convolutional Neural Network (Convnet) [100]. The main difference between a Convnet and a deep MLP is that Convnets use weight convolutions and multiplications:

$$X^1 = W^2(X^0 * W^1),$$

$$X^2 = W^4(X^1 * W^3),$$

$$Y = W^6(X^2 * W^5),$$

where $*$ represents the convolution operator in the above 2-layer Convnet. Since their creation Convnets have been used in a deep architecture to mimic the biological model for the human visual system. The best MNist results based on Convnet classification is a 0.23% misclassification error [101]. This used 35 Convnets in a committee with a two layer deep MLP used for data fusion. This network is an extremely complex parameterised model (almost 10^9 free parameters) requiring a sophisticated training procedure and observation alterations which improves the robustness and accuracy. The observations undergo a noising process at each layer in order to improve robustness and

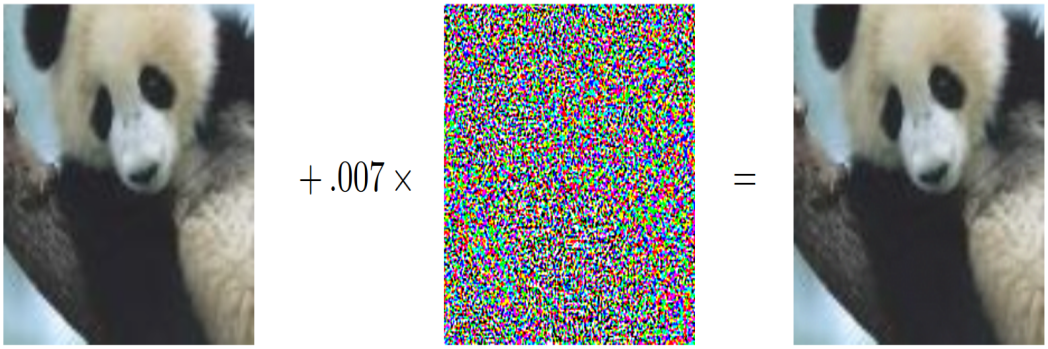


Figure 7.2: Adversarial example showing an image classified correctly as a panda (57.7% confidence) with random noise, classified as a nematode (8.2% confidence) which when added still appear to be a panda but is incorrectly classified as a gibbon (99.3% confidence)

artifically extend the size of the dataset to ensure there are enough different observations to train the large networks upon. Interestingly, the reasons for the success and focus of Convnets are not reliably known, as shown by the results of [102].

7.2.3 Issues

The results achieved using deep MLPs and Convnets are impressive. However, some issues with the learned mappings have been discovered. Two significant problems are outlined in [103] and [104].

Firstly, the mappings define unstable functions in the observation space. Networks trained on the MNist and CIFAR [105] image recognition databases were used to map and classify two sets of images. The first set of images was selected from the training data which was correctly classified with high confidence. The other set of images was created by perturbing the first set of images by an amount so small that the difference was visually imperceptible. The second set of images was incorrectly classified with high confidence, i.e. very small perturbations of input data result in large output deviations. Two of the images from [104] used to demonstrate this point are shown in figure 7.2. This example is not unique. The so called ‘adversarial example’ is constructed

specifically to generate this effect. Algorithms exist to replicate this effect for deep MLPs and Convnets for any training image [104]. The cause of this problem is the fractured input space created by the deep learning techniques. The observations are not smoothly interpolated, causing regions in the neighbourhood of an observation to be cracked, leading to incorrect interpolation in the data space and therefore incorrect classification. This is seen in the creation of adversarial examples, which search for a classification boundary in the neighbourhood of an observation by moving in the opposite direction of the decreasing cost function. In the case of smooth interpolation this boundary should be between two observations of different classes, not in the neighbourhood of observations of the same class, as found in [105]. In the following section it will be shown that this issue can be avoided by using smooth interpolating mappings, for instance, an RBF network.

The second issue with current deep learning machines is that of unreliable mappings. In [103] it was shown that not only is the input space fractured, creating instabilities, but that MLPs and Convnets can be easily 'fooled'. Trained networks were made to classify random noise and other artificially generated images that bore no resemblance to the class given at the output (for example the random image in figure 7.2). In the following sections, mapping uncertainty, motivated by Fisher Information as outlined in chapter 4, will be integrated with the Cascading RBF network. The purpose of this is to help prevent both of these failings in Deep Learning Machines.

7.3 The Cascading RBF

7.3.1 The Process

The main functional difference between the deep MLP and the Cascading RBF [106] is that the RBF structure uses topographic mappings optimised against STRESS measures as intermediate layers in the pre-training phase. RBF networks are in themselves a linear combination of a set of random basis functions over observed dissimilarities. In a shallow context RBF networks have been shown to be resistent to adversarial examples

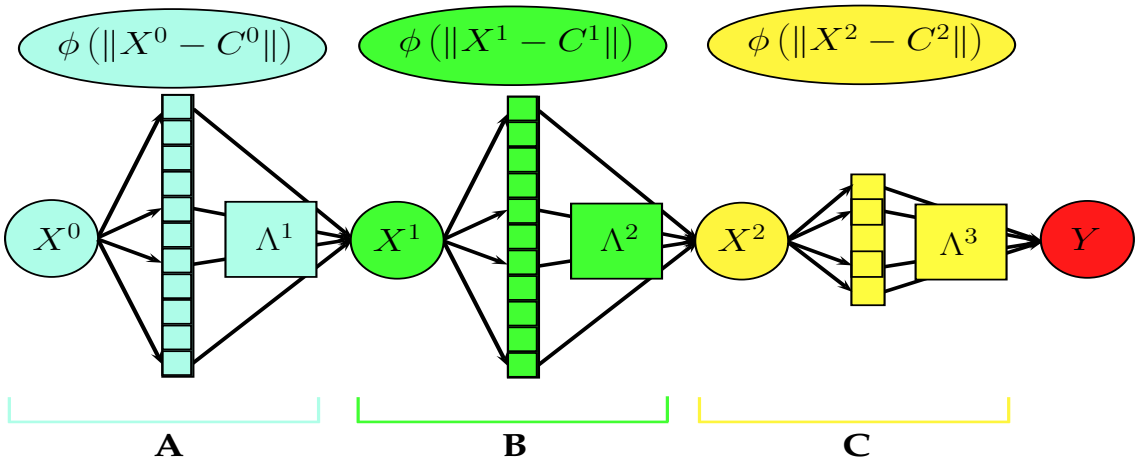


Figure 7.3: Schematic for a three layer cascading RBF structure mapping inputs, X^0, to predicted class labels, X^3. The mapping consists of two NS mappings with large feature spaces (A and B) and a classifier with a small feature space (C).

[107]. It should be noted that weights in the context of the Cascading RBF will be denoted Λ as opposed to W to signify the difference in approach from the use of weight matrices in classical Deep Learning Machines. For illustration of the basic principles, the example of a three layer cascaded model is used. The extension to further layers will be obvious. This model, illustrated in figure 7.3, can be expressed as:

- C: $Y = \Lambda^3 \Phi\left(d(X^2, C^2)\right)$,
- B: $X^2 = \Lambda^2 \Phi\left(d(X^1, C^1)\right)$,
- A: $X^1 = \Lambda^1 \Phi\left(d(X^0, C^0)\right)$.

With the weights, Λ^l, Φ, the matrix of nonlinearities over the dissimilarities between feature space (or observation), X^l, and centres C^l. The A,B and C notations relate the above equations to figure 7.3. As with standard RBFs described in Appendix A, the network centres can be fixed using distributions over the feature spaces, X^l, or randomly drawn from the space. The pre-training phase for the three layer network consists of:

1. Select network centres C^0 and use NeuroScale to map from X^0 to X^1. Any typically nonlinear basis function Φ used in ANN regression can be used. However, in light of the effectiveness of non-metric dissimilarity measures as introduced in [64], particularly measures which can have negative dissimilarities, splines or terms containing logarithms should be avoided since these require inputs

to be positive. The dissimilarity measure, $d(X^l, C^l)$ should be differentiable with respect to the inputs, preferably efficiently. Since this layer is trained using NeuroScale, the weights cannot overtrain and the mapping benefits from a large feature space, Φ. The cardinality of C^0 should be as close to that of X^0 as is feasible. X^1 is then fixed in the pre-training process.

2. Select network centres, C^1 and use NS to map from X^1 to X^2 as above. X^2 is then fixed in the pre-training process. The dimensionality of X^1 and X^2 should be large since a level of information is invariably lost in the NeuroScale mapping of dimension reduction. Since NeuroScale does not overtrain, there is no reason for the dimensionality to be small. In addition to this it is well known that models with a high number of parameters are better classifiers (typically such models overfit, but this is not of concern here).

3. Train a classifier to map from X^2 to Y to minimise the MSE:

$$Y = \Lambda^3 \Phi\left(d(X^2, C^2)\right) \Leftarrow \Lambda^3 = \Phi\left(d(X^2, C^2)\right)^{\dagger} T,$$

where T is a matrix of the true class labels. As a classifier, this feature space and therefore the weight matrix Λ^3, should be small to avoid overtraining. The network has now been pre-trained and two options exist for training the entire Cascading RBF as a classifier.

   (a) As with deep MLPs and Convnets, the weights Λ^1, Λ^2 and Λ^3 can be re-trained using backpropagation of gradients with respect to the error. Weight regularisation, such as an L_2 penalty, is normally used here to discourage overtraining.

   (b) As discussed in section 2.3.2, the training of the RBF weights in the NS framework is more efficient when the Shadow Targets algorithm is used than with standard gradient based methods. In contrast to the network outlined in section 2.3.2, an adapted Shadow Targets algorithm can still be implemented in the Cascading RBF. Firstly, the current error is given by:

$$E_{CRBF} = \sum_{i=1}^{N} \|Y^i - T^i\|^2,$$

then the feature space X^1 is updated as:

$$X^1 \leftarrow X^1 - \eta \frac{\partial E_{CRBF}}{\partial X^1},$$

where η is a learning rate. The weight matrix is updated as:

$$\Lambda^1 = \Phi\left(d(X^0, C^0)\right)^{\dagger} X^1.$$

The inputs X^0 are then mapped to Y to test whether the error has been reduced. The learning rate, η, is adapted to allow for a steepest descents approach such that η grows if the E_{CRBF} reduces and vice versa. After some iterations Λ^1 is fixed. The training then moves onto Λ^2:

$$X^2 \leftarrow X^2 - \eta \frac{\partial E_{CRBF}}{\partial X^2} \Lambda^2 = \Phi\left(d(X^1, C^1)\right)^{\dagger} X^2,$$

again mapping the inputs to Y using the learning rate in a gradient descent optimisation procedure to minimise the error. Following some stopping criteria, such as maximum iterations or error convergence rates, Λ^2 is then fixed. Finally, Λ^3 is then re-learned in a one step update:

$$\Lambda^3 = \Phi\left(d(X^2, C^2)\right)^{\dagger} T.$$

For the MNist dataset discussed in the next section the retraining of Λ^1 required only 20 iterations and Λ^2 required 30 iterations to converge to minima.

The layer-wise gradients are given in Appendix F. We now discuss a practical implementation of Cascading RBF's and compare with current state-of-the-art deep learning techniques.

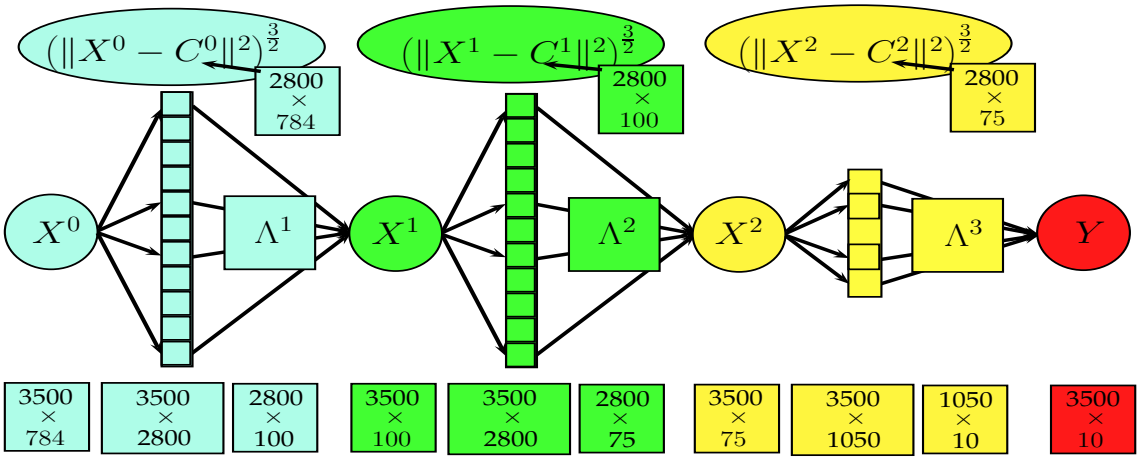


Figure 7.4: Schematic for a three layer Cascading RBF structure used on the MNist dataset, mapping inputs, X^0, to predicted class labels, X^3. The dimensions of each part of the mapping are shown at the bottom of the figure. The first two layers are pre-trained using NS and as such have a larger feature space than the final layer, pre-trained as a classifier.

7.3.2 The Test: MNist

The MNist database, as described in section 5.1, was used to construct a Cascading RBF. Since the database contains 60,000 training images there are computational constraints, particularly the storing, multiplying and pseudo-inversion of a 60,000 $\times M$ matrix, where M is the number of centres in C^0. With this in mind, a Cascading RBF was trained on sets of 3,500 samples of the training set. In order to make a fully representative classifier separate Cascading RBFs were trained on disjoint training sets and then subsequently combined with a fusion of outputs. The dissimilarity measures used were the squared Euclidean distances, which were more computationally convenient to compute than the standard Euclidean distance. The images $X_i^0 \in \mathbb{R}^{28\times 28}$ were treated as 784 dimensional vector observations, as is typical in the literature. The Cascading RBF architecture used on this dataset is shown in figure 7.4.

The initial NS mapping takes X^0 as 3,500 vectors using 2,800 of these as network centres. The 784 dimensional X^0 is mapped to $X^1 \in \mathbb{R}^{100}$. The desired nonlinear effect was a cubic over Euclidean distances so $\phi = d^{\frac{3}{2}}$.

The second NS mapping from $X^1 \in \mathbb{R}^{100}$ uses another randomly chosen 2,800 centres in X^1 to map to $X^2 \in \mathbb{R}^{75}$. The nonlinearity used is again $\phi = d^{\frac{3}{2}}$. The final layer is the classifier mapping from $X^2 \in \mathbb{R}^{75}$ to $Y \in \mathbb{R}^{10}$. 1,050 of the points from X^2 are used as

network centres with the same nonlinearity as the previous layers. The weights in all layers are then recomputed using the cascading Shadow Targets algorithm outlined above.

The Cascading RBF structure was repeated six times on disjoint sets of 3,500 training samples. The average training error was 0.11% and the average test error, based on the 10,000 test samples, was 4.63% . An equivalent deep MLP using Denoising AEs was trained using the Deep Learning Toolbox [108] which achieved training and test errors of 21.79% and 20.4% respectively. This performance shows the network is incapable of appropriately mapping between observations and class labels, matching the accuracy of the Cascading RBF. MLPs are traditionally considered better classifiers than RBFs as they perform nonlinear functional learning, focussing on the class boundaries. On the other hand RBF classification is linear in its weights, focussing on class centres. This is often a more difficult classification problem for complex observations.

As with the MCDNN described in [101] a fusion model was used to combine the six Cascading RBFs using an MLP. This nonlinear fusion model then draws on the strengths of each of the networks, trained on a total of 21,000 data observations. The fusion MLP took the concatenated Cascading RBF outputs as its input, $\bar{Y}^2 \in \mathbb{R}^{60}$ with a large hidden layer, $H \in \mathbb{R}^{50}$ and output layer $Y^2 \in \mathbb{R}^{10}$. The fusion model was trained on all training examples (60,000) and used a 'tanh' nonlinearity. This fusion model achieved a training error of 0.14%, based on the 60,000 training images, and test error of 0.09%, based on the 10,000 unseen images. This is a new world record over previously reported results. These results, along with other published record misclassification rates, are shown in table 7.1. The topographic mapping of intermediary layers and smoothness in input space helps the Cascading RBF to generalise well to out of sample data leading to the impressive results.

7.3.3 Unstable Functions

A simple way to test the instability of the Cascading RBF in a similar way to that of [105] is to inject small magnitude random noise into a known observation. This

Approach	Test Error
RBF [98] (1998)	3.6%
MLP [98] (1998)	1.6%
MLP [109] (2003)	0.7%
CNN [98] (1998)	0.7%
Virtual SVM [110] (2002)	0.56%
CNN [96] (2006)	0.39%
Deep MLP [111] (2010)	0.35%
CNN Committee [112] (2011)	0.27%
MCDNN [101] (2012)	0.23%
Cascading RBF [106] (2014)	**0.09%**

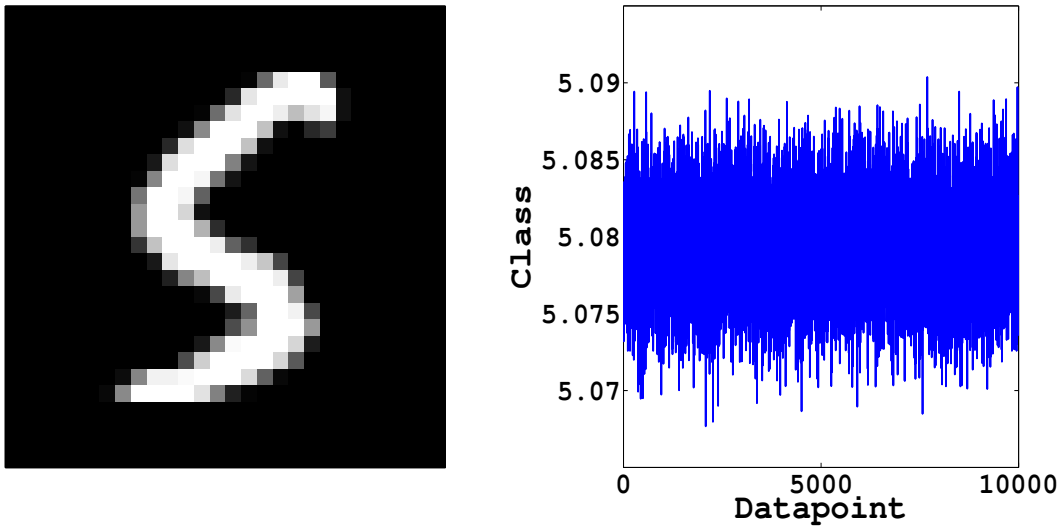
Table 7.1: Misclassification rates for several leading MNist classification methods with years in brackets. The Cascading RBF is a significant improvement over the previous world record.

observation should be correctly classified and, ideally, the perturbed images should also be correctly classified. The level of perturbation used is chosen to be at maximum the distance, d_{ij}, to the nearest neighbour j of an observation i (for the MNist dataset this is 18.5481). Perturbations larger than this could easily move the image beyond a class boundary and appear to be mapped incorrectly to a different class when the mapping is in fact correct. The example used on a Cascading RBF is a '5' shown in figure (7.5a). The image undergoes 10,000 separate random perturbations with Gaussian random noise ($\sigma = 9.2740$) with the resulting class predictions shown in figure (7.5b). It can be seen that the perturbations have not altered the output class as they can do with deep MLPs and Convnets.

A more thorough approach to proving the stability of the Cascading RBF networks is to prove that the functions are smooth, as the standard RBF is.

Theorem The Cascading RBF is of class C^{∞} and therefore smooth.

Proof We can show that the Cascading RBF is smooth by illustrating that the derivatives of all orders, with respect to the inputs, exist and are continuous. The derivatives calculated layer-by-layer using the chain rule are contained in Appendix F. It is clear that the first order derivatives of each individual element exist and are continuous. Furthermore, it can be shown that derivatives of higher orders exist due to



(a) Original Observation (b) Predicted Classes

Figure 7.5: Adversarial examples with the original image (left) and classification based on 10000 individual perturbations of the image. The predicted class labels do not change, ensuring the perturbed images are correctly classified.

the fact that differentiating ϕ with respect to D in each layer yields a continuous function.

Lemma The above proof relies on the infinite differentiability of ϕ. Typical functions used in RBF regression are polyharmonic splines $\left(\sum_j z^j \log(z)\right)$, polynomials or Gaussians. In order to preserve the smoothness of the Cascading RBF, preferred functions are Gaussians or polyharmonics and polynomials should be avoided.

Proposition The deep MLP is a subset of the Cascading RBF with $d(X,C) = XC$ and $\phi = \tanh(z)$. This dissimilarity measure is observed in the case of the Euclidean distance where the norm of X and C are unity. The cause of the fractures in the input space in deep MLP's is the discontinuities caused in the derivatives of the tanh or sigmoid functions typically used in MLPs. The discontinuities are observed as the functions approach step functions in the large derivative limit, with obvious discontinuities.

It can be concluded that the function defined by the Cascading RBF is contained within C^∞ and therefore smooth, preventing unstable mappings.

7.3.4 Unreliable Mappings

Despite being more complex than the standard RBF network used for N-NS, the mapping uncertainty described in chapter 4 can still be applied. The mapping uncertainty was given by $tr(I(\theta)^{-1})$ where $I(\theta)$ is the Fisher Information Matrix. For a Normal Distribution with constant scalar variance:

$$(I(\theta))_{ij} = \frac{\partial \mathbf{y}^T}{\partial \theta_i}\left(\frac{1}{\sigma^2}\right)\frac{\partial \mathbf{y}}{\partial \theta_j}. \tag{7.3}$$

The Cascading RBF used here is a pointwise mapping, using the probabilistic alternative mapping based on N-NS is possible and potentially more accurate, but more computationally demanding. As such, each output has no individual uncertainty or corresponding covariance matrix. In order to apply the mapping uncertainty to the deterministic Cascading RBF, a global variance can be found through the training process as the MSE:

$$\sigma^2 = E_{CRBF} = \frac{1}{2N}\sum_{i=1}^{N}\|Y_i - T_i\|^2, \tag{7.4}$$

describing the uncertainty of a class prediction from all trained observations. This global variance will be used in the calculation of $I(\theta)$ from equation (7.3). The gradients with respect to the weight parameters Λ^1, Λ^2 and Λ^3 are given in Appendix F. To test the reliability of a Cascading RBF, 100 training images, 100 test images and 100 images of random noise at differing intensities were propagated to generate outputs. The resulting mapping uncertainties, given by $FI_i = tr(I(\theta)^{-1})$, for each of these datapoints are shown in figure 7.6. The mapping uncertainty for the random noise is less than for the standard images. This is because the dissimilarities between each of the network centres are approximately equal, and much larger than those of the real images, ensuring $\phi(d(X_i, C_j))$ in each layer is large. This forces the trace of the inverse matrix as given by equation (4.12) to be lower than for standard images. Typically it is expected that a high level of mapping uncertainty indicates an anomaly, however this is because such an observation is usually coupled with a high level of uncertainty. This is not the case here due to the shared global variance, σ^2. It should be noted that the mapping uncertainty

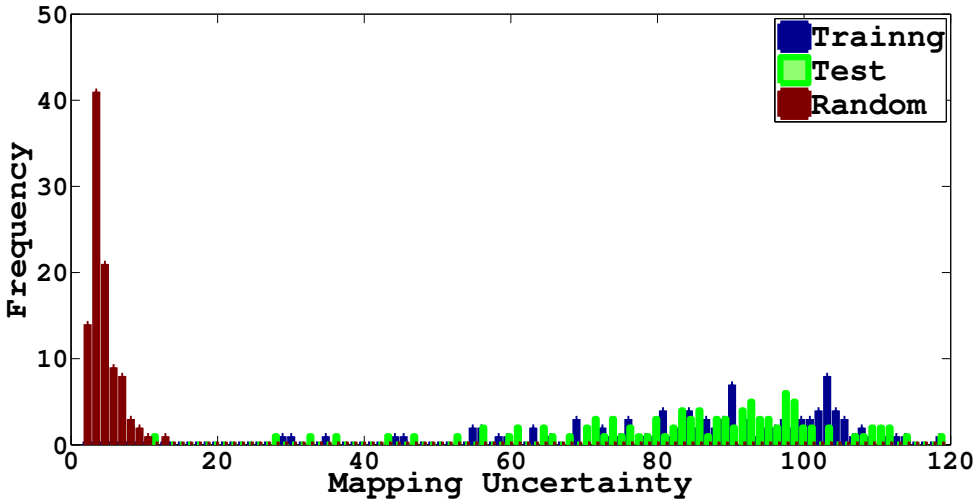


Figure 7.6: Histogram of mapping uncertaintes for the trained, test and random images for the MNist dataset Cascading RBF. The random noise images are clearly separated from the training and test images in terms of their mapping uncertainty. Due to the global variance used in the caluclation of FI_i the anomalies are indicated by a low level of mapping uncertainty.

can also be implemented at the first layer of the Cascading RBF network acting upon Φ^0, since the anomalous mappings are identified such that the elements of D^0 are outside of the expected range for which the network is trained. This is computationally less demanding than the calculation of the total mapping uncertainty, but can be a less sensitive method of anomaly detection than assessing the compounded mapping uncertainties.

It is clear that the random noise is outside of the expected region of uncertainty. A Gaussian distribution can be fit to the trained mapping uncertainties providing a confidence probability. The choice of distribution will be data dependent but a GMM could be adapted to any dataset as required. Thresholding these probabilities can provide a simple way of making a 'null' classification, for instance, that the observation is unclassifiable. This is particularly useful since it not only prevents the failings of deep MLPs and Convnets but provides a simple solution to the difficult problem of automatically deciding that an observation has no known class. Training classifiers to try and perform this with random noise has not worked in the literature [104].

7.4 Overview

Deep learning provides unparalleled results on classification tasks and is the current topic of research for many in the field of Machine Learning. The failings of the two main deep architectures currently in use have prompted their reliability to be questioned. The use of RBF networks as building blocks for creating a Cascading RBF structure, using dissimilarity-based pre-training motivated from topographic visualisation, avoids these shortcomings. The Cascading RBF has also been shown to achieve world record performance when tested on the MNist handwritten digits database. Further work will incorporate the probabilistic N-NS framework into the network in combination with the mapping uncertainty measures to build a robust and informative network.

8 Conclusions

> 'Mathematics is written for mathematicians.'
>
> - Nicholaus Copernicus

This chapter will provide a review of the thesis, re-iterate the contributions made and suggest avenues for ways in which the work presented could be extended.

8.1 Review Of Thesis

An introductory background to some of the popular methods in the field of data visualisation was given in chapter 2. This review is in no way exhaustive, but provides the required detail for knowledge of popular methods, particularly those pertinent to the later work in this thesis. Three criteria for quantitatively analysing visualisation performance were also outlined. A quantitative approach to assessing quality in visualisations is particularly important as low-dimensional representations can easily be

poorly interpreted. It is easy for mappings to create visualisations which appear to have structure when in fact there is none [49]. In addition to this, different mappings generate different representations and researchers are not in agreement as to which mapping creates the 'best' visualisations. This highlights the requirement for these performance criteria.

Chapter 3 extended the deterministic mappings outlined in chapter 2 to allow for observation uncertainty. This avoids the deficiencies in certain mappings in the presence of noisy data. The extensions also allow for distributions, instead of points, in the visualisation space, ensuring they are as representative of the observations as possible.

Chapter 4 proposed a method for representing both the uncertainties generated by observations and the visualisation mapping itself. These allow for an informative data respresentation such that outliers can be easily identified. The proposed mappings from chapter 3 were incorporated into the framework of an RBF network, allowing for feed-forward projection of new data.

In chapter 5 the new methods of chapters 3 and 4 were implemented on three vectorial datasets, accounting for data uncertainty. The representations were intuitive, informative and reliable, as judged by the quality criterion used in this thesis. In the case of the MNist dataset the probabilistic mappings generated a latent representation with a greater degree of separation between classes, a desirable property in pattern recognition.

A process for visualising anomalies in time series data, Residual Modelling, was introduced in chapter 6, following which it was demonstrated on three different datasets. The novel approach focussed on deviations from an expected signal, and as such fitted perfectly into the probabilistic visualisation framework.

In contrast to the other areas of the thesis, chapter 7 combined topographic mapping with a deep learning machine in a classification setting. It was shown that the new Cascading RBF, when considered as a committee, outperformed other state-of-the-art classifiers, in addition to possessing other desirable properties such as smoothness and reliability.

8.2 Contributions

The contributions of this thesis include:

- Probabilistic extensions to NeuroScale, Locally Linear Embedding, Isomap and Laplacian Eigenmaps were introduced, accounting for observation uncertainty.
- A framework for interpreting observation and the imposed mapping uncertanty in visualisation spaces was outlined.
- A novel method for detecting anomalies in time series data using topographic visualisation, Residual Modelling, was described.
- A new form of Deep Learning Machine, the Cascading RBF, consisting of topographically pre-trained RBF networks was implemented in a classification setting

8.3 Future Work

1. It was found in chapters 5 and 6 that visualised data observations appear in clusters; and as such the Uncertainty Surface can have a level of redundancy. This is caused by the crowding of several of the latent distributions in a relatively small area of visualisation space. A summary of the mixture distributions which generate the Uncertainty Surface could be used, also reducing the number of RBF network centres. The impact of this will be the subject of future research.
2. The representations generated using T-NS appeared to be of worse quality than those of N-NS. This is possibly due to an incorrectly chosen ‘ν’ parameter describing the degrees of freedom, however further experimentation on other datasets is required to prove this.
3. The other probabilistic mappings of chapter 3, PLLE, PIso and PWNM, can also be extended to allow for a T-distributed latent space instead of the Gaussian latents used in the current forms. The different error functions of PLLE and PWNM in

particular may allow for better representations in this instance than those of T-NS. Other, parameterless, multivariate distributions such as the Multivariate Laplace [113] will also be considered.

4. The extension of metric MDS using Bregman divergences in [24] showed that the local mapping quality was improved by altering the standard error functions. Other divergences were applied to SNE in [114] with significant changes to the visualisation space. It is expected that these changes and benefits will transfer to the newly proposed N-NS, T-NS and PIso.

5. It is typical in the literature that dissimilarities in the visualisation space are the Euclidean distance. In this thesis all probabilistic dissimilarities in the visualisation space are the Kullback-Leibler divergence. The impact of different dissimilarity measures is untested and as such will be the subject of future research.

Bibliography

[1] A. R. Webb, *Front Matter*, pp. 1–496. John Wiley & Sons, Ltd, 2003.

[2] C. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[3] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2nd ed., 1998.

[4] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[5] J. B. Tenenbaum, V. Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[6] K. Pearson, "On lines and planes of closest fit to systems of points in space.," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.

[7] C. M. Bishop, M. Svensen, and C. K. I. Williams, "GTM: The generative topographic mapping," *Neural Computation*, vol. 10, no. 1, pp. 215–234, 1998.

[8] N. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Advances in Neural Information Processing Systems 16*, pp. 329–336, December 2003.

[9] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Trans. Comput.*, vol. 18, pp. 401–409, May 1969.

[10] J. Lee, A. Lendasse, and M. Verleysen, "Curvilinear distances analysis versus Isomap," in *Proceedings of ESANN 2002, 10th European Symposium on Artificial Neural Networks* (M. Verleysen, ed.), (Bruges, Belgium), pp. 185 – 192, April 2002. d-side.

[11] P. Demartines and J. Herault, "Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets," *Neural Networks, IEEE Transactions on*, vol. 8, pp. 148–154, Jan 1997.

[12] J. Lee, A. Lendasse, N. Donckers, and M. Verleysen, "A robust nonlinear projection method," in *Proceedings of ESANN 2000, 8th European Symposium on Artificial Neural Networks* (M. Verleysen, ed.), pp. 13–20, D-Facto public., Bruges, Belgium, April 2000.

[13] A. C. Damianou and N. D. Lawrence, "Deep gaussian processes," in *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April 29 - May 1, 2013*, pp. 207–215, 2013.

[14] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-sne," *JLMR*, vol. 9, pp. 2579–2605, Nov 2008.

[15] T. Lin, H. Zha, and S. Lee, "Riemannian manifold learning for nonlinear dimensionality reduction," in *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, (Berlin, Heidelberg), pp. 44–55, Springer-Verlag, 2006.

[16] J. A. Lee, C. Archambeau, and M. Verleysen, "Locally Linear Embedding versus Isotop," in *ESANN*, pp. 527–534, 2003.

[17] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*. Springer Publishing Company, Incorporated, 1st ed., 2007.

[18] W. Torgerson, "Multidimensional scaling i: Theory and method," *Psychometrika*, vol. 17, pp. 401–419, 1952.

[19] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, pp. 1373–1396, 2002.

[20] L. van der Maaten, "drtoolbox - matlab toolbox for dimensionality reduction," 2007. http://lvdmaaten.github.io/drtoolbox/.

[21] I. Nabney, "Netlab neural networking toolbox," 2001. http://www.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/.

[22] N. Lawrence, "Gpmat toolbox," 2009. http://staffwww.dcs.shef.ac.uk/people/N.Lawrence/software.html.

[23] J. Sun, "Open box dataset," 2012. http://cis.uws.ac.uk/research/JigangSun/index.html.

[24] J. Sun, *Extending Metric Multidimensional Scaling with Bregman Divergences*. PhD thesis, University of the West of Scotland, 2011.

[25] H. Hotelling, "Analysis of a complex statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417–441, 1933.

[26] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," nov 2012. Version 20121115.

[27] I. Rice, R. Benton, L. Hart, and D. Lowe, "Analysis of multibeam SONAR data using dissimilarity representations," in *3rd IMA Mathematics In Defence 2013,*, October 2013.

[28] I. Rice and D. Lowe, "Topographic visual analytics of multibeam dynamic sonar data," in *Sensor Signal Processing for Defence (SSPD), 2014*, pp. 1–5, Sept 2014.

[29] D. Lowe and M. E. Tipping, "Neuroscale: Novel topographic feature extraction using rbf networks," in *Advances in Neural Information Processing Systems 9* (M. Mozer, M. Jordan, and T. Petsche, eds.), pp. 543–549, MIT Press, 1997.

[30] J. Mao and A. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Transactions on Neural Networks*, vol. 6, pp. 296–317, March 1995.

[31] M. E. Tipping and D. Lowe, "Shadow targets: A novel algorithm for topographic projections by radial basis functions," *NeuroComputing*, vol. 19, pp. 211–222, 1997.

[32] M. Tipping, *Topographic mappings and feed-forward neural networks*. PhD thesis, Aston University, Aston Street, Birmingham B4 7ET, UK., February 1996.

[33] M. Sivaraksa and D. Lowe, "Probabilistic neuroscale for uncertainty visualisation," in *13th International Conference on Information Visualisation, IV 2009, 15-17 July 2009, Barcelona, Spain*, pp. 74–79, 2009.

[34] E. Dijkstra, "A note on two problems in connection with graphs," *Numerical Mathematics*, vol. 1, pp. 269–271, 1959.

[35] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, p. 345, June 1962.

[36] S. Amari, *Differential-geometrical methods in statistics*. Lecture Notes in Statistics, Springer-Verlag, 1985.

[37] J. A. Lee, A. Lendasse, and M. Verleysen, "Curvilinear distance analysis versus isomap.," in *Proceedings of ESANN 2002, 10th European Symposium on Artificial Neural Networks* (M. Verleysen, ed.), pp. 185–192, 2002.

[38] J. A. Lee and M. Verleysen, "Nonlinear dimensionality reduction of data manifolds with essential loops," *Neurocomput.*, vol. 67, pp. 29–53, Aug. 2005.

[39] F. R. K. Chung, *Spectral Graph Theory*. Providence, RI: Amer. Math. Soc., 1997.

[40] J. Li, J.-M. Guo, and W. Shiu, "Bounds on normalized Laplacian eigenvalues of graphs," *Journal of Inequalities and Applications*, vol. 2014, no. 1, p. 316, 2014.

[41] T. Kohonen, "Self-organization of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.

[42] I. Olier and A. Vellido, "On the benefits for model regularization of a variational formulation of GTM," in *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence)*, pp. 1568–1575, June 2008.

[43] C. M. Bishop, M. Svensén, and C. K. I. Williams, "Magnification factors for the GTM algorithm," in *Proceedings IEE Fifth International Conference on Artificial Neural Networks, Cambridge, U.K.*, pp. 64–69, January 1997.

[44] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[45] A. G. Wilson, D. A. Knowles, and Z. Ghahramani, "Gaussian process regression networks," in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, pp. 1–8, June 2012.

[46] B. Rakitsch, C. Lippert, K. Borgwardt, and O. Stegle, "It is all in the noise: Efficient multi-task gaussian process inference with structured residuals," in *Advances in Neural Information Processing Systems 26* (C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, eds.), pp. 1466–1474, 2013.

[47] M. K. Titsias and N. D. Lawrence, "Bayesian gaussian process latent variable model," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, pp. 844–851, 2010.

[48] N. D. Lawrence and J. Q. Candela, "Local distance preservation in the GP-LVM through back constraints," in *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pp. 513–520, 2006.

[49] N. P. Hughes and D. Lowe, "Artefactual structure from least-squares

multidimensional scaling.," in *NIPS* (S. Becker, S. Thrun, and K. Obermayer, eds.), pp. 913–920, MIT Press, 2002.

[50] J. A. Lee and M. Verleysen, "Quality assessment of dimensionality reduction: Rank-based criteria," *Neurocomput.*, vol. 72, pp. 1431–1443, Mar. 2009.

[51] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[52] L. Chen and A. Buja, "Local multidimensional scaling for nonlinear dimension reduction, graph drawing and proximity analysis.," tech. rep., University of Pennsylvania, 2006.

[53] Nist Digital Library of Mathematical Functions, "Equation 5.19.4," May 2006. http://dlmf.nist.gov/5.19.

[54] T. P. S. University, "Geometry of the multivariate normal distribution," 2015. https://onlinecourses.science.psu.edu/stat505/node/36.

[55] A. G. Akritas, E. K. Akritas, and G. I. Malaschonok, "Various proofs of Sylvester's (determinant) identity," *Mathematics and Computers in Simulation*, vol. 42, no. 4-6, p. 585, 1996.

[56] A. Shah, A. Wilson, and Z. Ghahramani, "Student-t Processes as Alternatives to Gaussian Processes," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, pp. 877–885, 2014.

[57] S. Kotz and S. Nadarajah, *Multivariate t Distributions and Their Applications*. Cambridge University Press, 2004.

[58] R. B. Arellano-Valle, J. E. Contreras-Reyes, and M. G. Genton, "Shannon entropy and mutual information for multivariate skew-elliptical distributions," *Scandinavian Journal of Statistics*, vol. 40, no. 1, pp. 42–62, 2013.

[59] J. E. Contreras-Reyes, "Asymptotic form of the Kullback-Leibler divergence for multivariate asymmetric heavy-tailed distributions," *Physica A*, vol. 395, pp. 200–208, 2014.

[60] J. Hershey and P. Olsen, "Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, pp. 317–320, April 2007.

[61] J.-L. Durrieu, J.-P. Thiran, and F. Kelly, "Lower and upper bounds for approximation of the Kullback-Leibler divergence between Gaussian Mixture Models," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4833–4836, March 2012.

[62] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering," in *Advances in Neural Information Processing Systems 16* (S. Thrun, L. Saul, and B. Schölkopf, eds.), pp. 177–184, MIT Press, 2004.

[63] L. van der Maaten, "Learning a parametric embedding by preserving local structure.," in *AISTATS* (D. A. V. Dyk and M. Welling, eds.), vol. 5 of *JMLR Proceedings*, pp. 384–391, JMLR, 2009.

[64] E. Pękalska and R. P. W. Duin, *The Dissimilarity Representation for Pattern Recognition: Foundations And Applications (Machine Perception and Artificial Intelligence)*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2005.

[65] E. Pękalska, R. P. W. Duin, and P. Paclík, "Prototype selection for dissimilarity-based classifiers," *Pattern Recogn.*, vol. 39, pp. 189–208, Feb. 2006.

[66] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal, The*, vol. 27, pp. 379–423, July 1948.

[67] B. R. Frieden, *Science from Fisher Information*. Cambridge University Press, 2nd ed., 2004.

[68] C. R. Rao, "Information and the accuracy attainable in the estimation of statistical parameters," *Bulletin of the Calcutta Mathematical Society*, vol. 37, pp. 81–89, 1945.

[69] A. C. Atkinson and A. N. Donev, *Optimum Experimental Designs*. Oxford University Press, 1992.

[70] I. Rice and D. Lowe, "A robust, realistic noise model for visualisation and anomaly detection of multi-beam sonar targets," in *10th IMA Mathematics in Signal Processing, 2014*, pp. 1–4, 2014.

[71] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database." http://yann.lecun.com/exdb/mnist/.

[72] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting," in *Advances in Neural Information Processing Systems 15, Vancouver*, pp. 529–536, 2002.

[73] T. Wittman, "Mani fold learning matlab demo," 2005. http://www.math.ucla.edu/ wittman/mani/.

[74] S. Brahim-Belhouari and A. Bermak, "Gaussian Process for nonstationary time series prediction," *Computational Statistics & Data Analysis*, vol. 47, no. 4, pp. 705 –712, 2004.

[75] B. R. Matam, B. K. Fule, H. P. Duncan, and D. Lowe, "Predictability of unplanned extubations," in *Proceedings of IEEE-EMBS International Conference on Biomedical and Health Informatics, BHI 2014, Valencia, Spain, June 1-4, 2014*, pp. 488–491, 2014.

[76] J. D. Hamilton, *Time series analysis*. Princeton University Press, 1994.

[77] D. Nguyen-Tuong, B. Scholkopf, and J. Peters, "Sparse online model learning for robot control with support vector regression," in *Intelligent Robots and Systems,*

2009. IROS 2009. IEEE/RSJ International Conference on, pp. 3121–3126, Oct 2009.

[78] A. F. Eamonn Keogh, Jessica Lin, 2005. http://www.cs.ucr.edu/ eamonn/discords/.

[79] C. Bauckhage, "Computing the kullback-leibler divergence between two generalized gamma distributions," *arxiv*, pp. 1–7, 2014.

[80] E. Keogh, J. Lin, and A. Fu, "Hot sax: efficiently finding the most unusual time series subsequence," in *Data Mining, Fifth IEEE International Conference on*, pp. 1–8, Nov 2005.

[81] A. E. Society, "American epilepsy society seizure prediction challenge," 2014. https://www.kaggle.com/c/seizure-prediction.

[82] H. Liu, J. D. Lafferty, and L. A. Wasserman, "Sparse nonparametric density estimation in high dimensions using the rodeo," in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)* (M. Meila and X. Shen, eds.), vol. 2, pp. 283–290, 2007.

[83] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007.

[84] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 721–741, 1984.

[85] D. M. Blei and M. I. Jordan, "Variational methods for the Dirichlet process," in *In Proceedings of the 21th International Conference on Machine Learning (ICML)*, pp. 121–144, 2004.

[86] Y. Raykov, A. Boukouvalas, and M. Little, "Simple approximate MAP Inference for Dirichlet processes." arXiv, 2014.

[87] D. Abraham, J. Gelb, and A. Oldag, "K-Rayleigh mixture model for sparse active SONAR clutter," in *OCEANS 2010 IEEE - Sydney*, pp. 1–6, May 2010.

[88] S. Haykin and S. Puthusserypady, *Chaotic dynamics of sea clutter*. Adaptive and Learning Systems for Signal Processing, Communications and Control, John Wiley & Sons, 1999.

[89] R. Bareš, *Environmentally Adaptive Noise Estimation for Active Sonar*. PhD thesis, Cardiff University, January 2011.

[90] D. Abraham and A. Lyons, "Reliable methods for estimating the k -distribution shape parameter," *Oceanic Engineering, IEEE Journal of*, vol. 35, no. 2, pp. 288–302, 2010.

[91] S. Hochreiter and J. Schmidhüber, "Long short-term memory," tech. rep., Fakultät für Informatik, Technische Universität München, August 1995. FKI-207-95.

[92] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[93] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[94] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[95] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, (New York, NY, USA), pp. 1096–1103, ACM, 2008.

[96] M. Ranzato, C. S. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model.," in *NIPS 2006* (B. Schölkopf, J. C. Platt, and T. Hoffman, eds.), pp. 1137–1144, MIT Press, 2006.

[97] R. Salakhutdinov and G. E. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," in *Proceedings of AISTATS*, vol. 2, pp. 412–419, 2007.

[98] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, Nov 1998.

[99] L. Deng and D. Yu, "Deep convex network: A scalable architecture for speech pattern classification," in *Interspeech*, pp. 1–4, International Speech Communication Association, August 2011.

[100] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in Neural Information Processing Systems*, vol. 2, pp. 396–404, 1990. Denver 1989, Morgan Kaufmann, San Mateo.

[101] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *arxiv*, pp. 1–8, 2012.

[102] D. Eigen, J. T. Rolfe, R. Fergus, and Y. LeCun, "Understanding deep architectures using a recursive convolutional network," in *In Proc. International Conference on Learning Representations (ICLR) 2014*, pp. 1–9, 2014.

[103] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," pp. 1–9, 2014.

[104] A. Nguyen, J. Yosinski, and J. Clune, "Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–10, June 2015.

[105] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar dataset," 2009. http://www.cs.toronto.edu/ kriz/cifar.html.

[106] I. Rice and D. Lowe, "Deep layer radial basis function networks for classification," in *10th IMA Mathematics in Signal Processing, 2014*, pp. 1–4, 2014.

[107] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, no. 3, pp. 29–41, 2015.

[108] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," Master's thesis, Technical University of Denmark, DTU Informatics, Asmussens Alle, Building 305, DK-2800 Kgs. Lyngby, Denmark, 2012.

[109] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *7th International Conference on Document Analysis and Recognition (ICDAR 2003), 2-Volume Set, 3-6 August 2003, Edinburgh, Scotland, UK*, pp. 958–962, 2003.

[110] D. Decoste and B. Schölkopf, "Training invariant support vector machines," *Mach. Learn.*, vol. 46, pp. 161–190, Mar. 2002.

[111] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep big simple neural nets excel on handwritten digit recognition," *arxiv*, 2010.

[112] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*, pp. 1135–1139, 2011.

[113] T. Eltoft, T. Kim, and T.-W. Lee, "On the multivariate laplace distribution," *Signal Processing Letters, IEEE*, vol. 13, pp. 300–303, May 2006.

[114] O. Dikmen, Z. Yang, and E. Oja, "Learning the information divergence," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, pp. 1442–1454, July 2015.

[115] D. Broomhead, D. S.; Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," Tech. Rep. 4148, RSRE, 1988.

[116] D. Lowe, "Radial basis function networks - revisited," *IMA Mathematics Today*, vol. 51, pp. 124 – 128, June 2015.

[117] G. Hinton and S. Roweis, "Stochastic neighbor embedding," *Advances in neural information processing systems*, vol. 15, pp. 833–840, 2003.

[118] L. Van-der Maaten, "Tsne examples." http://lvdmaaten.github.io/tsne/implementations, 2015.

[119] D. K. Duvenaud, O. Rippel, R. P. Adams, and Z. Ghahramani, "Avoiding pathologies in very deep networks," in *AISTATS 14*, pp. 202–210, 2014.

[120] C. Bishop, M. Svensen, and K. Williams, "GTM: A principled alternative to the self-organizing map," *Neural Computation*, vol. 10, no. 1, pp. 215–234, 1998.

A Radial Basis Function networks

A Radial Basis Function (RBF) network [115] is a form of Artificial Neural Network performing nonlinear interpolation using a combination of basis functions and linear weights. A historical overview of RBF networks is given in [116]. The network inputs, X_i, are compared to a series of network centres, C_j, such that:

$$\mathbf{y}_i^l = \sum_j \mathbf{W}_{lj} \phi \left[d(X_i, C_j) \right], \tag{A.1}$$

where l is the output dimension and d some dissimilarity measure. In the case that X_i is vectorial, $X_i = \mathbf{x}_i$ and $C_j = \mathbf{c}_j$. Typically d is taken to be the Euclidean distance, but other measures can be used, particularly if the observations are non-vectorial. The network centres are treated as prototypes, sampled from the data for example by Gaussian Mixture Models or randomly chosen from the data. The matrix formed by $d(X_i, C_j)$ over multiple observations, i, can be considered as a dissimilarity space as described in [64]. Methods for prototype selection in these spaces are outlined in [65] but it is particularly

worthy of note that random selection outperforms many measures for low numbers of prototypes (e.g. $\frac{N}{2}$) for many datasets. The RBF outputs from equation (A.1) can also be written in matrix form:

$$Y = \Phi W. \tag{A.2}$$

The network weights are optimised to minimise the Mean-Square Error (MSE):

$$E_{RBF} = \frac{1}{2}\sum_i \|\mathbf{t}_i - \mathbf{y}_i\|^2 = \frac{1}{2}\|T - \Phi W\|^2, \tag{A.3}$$

where T are the true observed targets. This error function is minimised by setting $W = \Phi^{\dagger}T$. Minimisation of E_{RBF} is relatively insensitive to the choise of ϕ, but Splines (thin plate or Polyharmonic) and Gaussian activation functions are a popular choice. The nature of the flexibility of the feed-forward RBF and its ability to use any dissimilarity measure, d, make it suitable for implementation in topographic mappings. It should be noted that the network is radial because of the radial symmetry in the Euclidean distance. Despite d sometimes being non-radial in this thesis it will still be referred to as an RBF as the learning procedure and nature of the centres remains the same.

B Non-Topographic Visualisation Mappings

B.1 Introduction

This appendix will introduce three popular visualisation algorithms, T-distributed Stochastic Neighbour Embedding (T-SNE), the AutoEncoder (AE) and the Deep Gaussian Process (Deep GP) and justify why they are not used comparatively in this thesis. The focus of this thesis is on topographic mappings, largely because if a visualisation is not topographic it can easily lead to unrealistic and untrue interpretations of the observations. Firstly T-SNE will be outlined and then a comparative example will show why it cannot map observations topographically. The reasons for which the AE and Deep GP are not topographic is fairly obvious and therefore do not require the same level of detail.

B.2 T-SNE

T-SNE [14] is an extension to the concept of LLE. A distribution is specified over neighbourhoods such that the probability that an observation j is a neighbour of another observation i is:

$$p_{j|i} = \frac{\exp\{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2\}}{\sum_{k\neq i}\exp\{-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2\}}, \tag{B.1}$$

where σ_i is induced by the distribution over neighbours, called the perplexity:

$$\sigma_i = \text{Perp}(p_i) = 2^{H(p_i)},$$

$$H(p_i) = -\sum_j p_{j|i}\log_2(p_{j|i}).$$

The conditional distribution $p_{j|i}$ can be symmetrised to give p_{ij}:

$$p_{ij} = \frac{\exp\{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2\}}{\sum_l\sum_{k\neq l}\exp\{-\|\mathbf{x}_k - \mathbf{x}_l\|^2/2\sigma^2\}}. \tag{B.2}$$

In the original approach to Stochastic Neighbour Embedding (SNE) [117] a Gaussian distribution is used to model the latent dissimilarities, however this resulted in the 'crowding problem'. This isue is that moderately distant points in the observation space will not be given a sufficiently large enough space in a two-dimensional (or low-dimensional) mapping, compared with the mapping space allocated to nearby observations. As such T-SNE uses a t-distribution to create a mismatch between the neighbourhood distributions' tails of observations (p_{ij}) and visualised points (q_{ij}):

$$q_{ij} = \frac{\left(1+\|\mathbf{y}_i - \mathbf{y}_j\|^2\right)^{-1}}{\sum_{k\neq l}\left(1+\|\mathbf{y}_k - \mathbf{y}_l\|^2\right)^{-1}}. \tag{B.3}$$

The mapping error for T-SNE attempts to match q_{ij} and p_{ij} using the Kullback-Leibled divergence:

$$E_{TNSE} = KL(P\|Q) = \sum_i\sum_j p_{ij}\log\left(\frac{p_{ij}}{q_{ij}}\right). \tag{B.4}$$

The visualised points, $\mathbf{y}_i$, are learned through gradient descent using:

$$\frac{\partial E_{T-SNE}}{\partial \mathbf{y}_i} = 4\sum_j (p_{ij} - q_{ij})(\mathbf{y}_i - \mathbf{y}_j)(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}. \tag{B.5}$$

The error function is nonconvex, often with many poor local minima. As such, the gradient learning phase requires the use of a momentum term to try and avoid these local minima. There are three issues with T-SNE which prevent it from mapping topographically. Firstly the assumption of dissimilarities in observation space being approximately Gaussian distributed is not suitable for many real world problems. All datasets considered in this thesis in sections 5 and 6 do not follow this assumption, they are more closely related to a Gamma distribution. This will cause a poor dissimilarity representation in observation space, ensuring that the matching T-distribution in latent space will not topographically map the data.

Secondly the minimisation of the KL between a Gaussian and a T-distribution behaves strangely when the parameter is the dissimilarity. Ignoring the normalisation constants the integral is approximately:

$$KL(P\|Q) \approx \int \exp\{-\frac{1}{2}d_{ij}\} \log\left(\frac{\exp\{-\frac{1}{2}d_{ij}\}}{(1+d^*_{ij})^{-1}}\right), \tag{B.6}$$

where d_{ij} and d^*_{ij} represent the dissimilarities in observation and visualisation spaces respectively. This integral is analytically intractable however, numerical integration shows the error is reduced when $d^*_{ij} \ll d_{ij}$, tending towards a constant value (-0.6267 for the un-normalised integral (B.2)). This issue is equivalent to the possibility of the error in LLE from equation (2.8) being reduced by artifically rescaling the axis, except that in LLE there is a constrainty, $YY^T = I$, used to prevent this. This is why the recommendation in [117] to set the initial neighbourhood probabilities to $P \leftarrow 4P$, artificially increasing the observed dissimilarities, helps in the optimisation process. This will force q_{ij} to be larger at the initial stages of the training process, attempting to prevent this problem. Thirdly the T-SNE mapping uses global dissimilarities, similar to the Sammon Map and as such should preserve global relationships. This is not the case

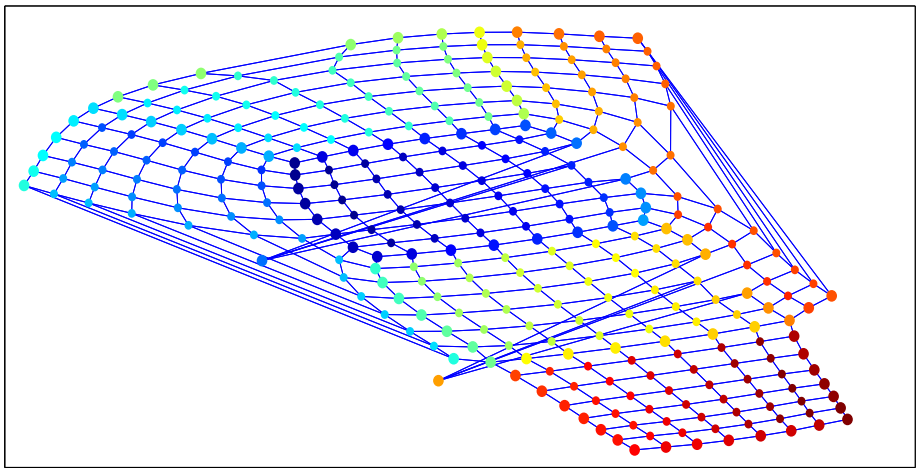


Figure B.1: Open box embedded by T-SNE. The mapping has unfolded the box from the open top. The sides of the structure have been split from the top down with most of the points joined to the front face. The lid is approximately mapped to a square as in the observation space. There are however tears in the mapping, with some observations not evenly spaced along the latent shape of the front face as would be expected. Two points from the side of the bottom and connected side in visualisation space are incorrectly separated from their neighbours.

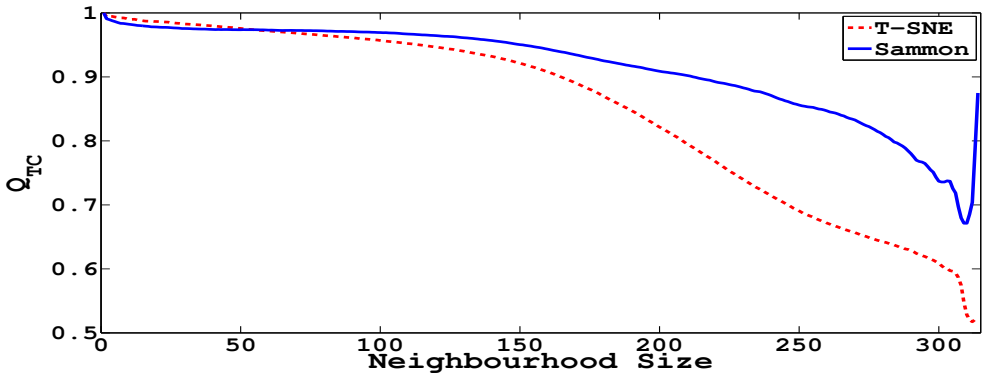
in practice, as will be shown with the Open Box and a randomly generated dataset. The T-SNE embedding of the Open Box dataset is shown in figure B.1.

It is clear that each face of the box has been preserved at the cost of the global continuity by tearing open the structure. Figure B.2 shows a comparison of the quality criterion of the T-SNE box versus the Sammon mapped box.

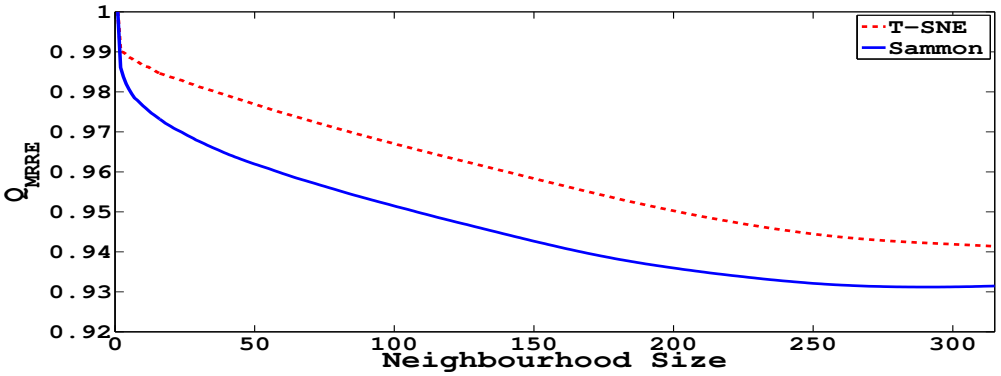
The local rank errors are better for T-SNE, illustrated by the higher Q_{MRRE} and LCMC however, the poor global trustworthiness of the T-SNE compared to the Sammon box is concerning. This is caused by the tearing and the imposed curvature of the box faces.

A further illustrative example is that of a series of randomly organised 2-dimensional points sat on a flat plane in 3-dimensional space (the values of the third dimension are all 0). Both the T-SNE and Sammon Map visualisations should be able to retain the relative local and global neighbourhoods well in a 2-dimensional visualisation space.

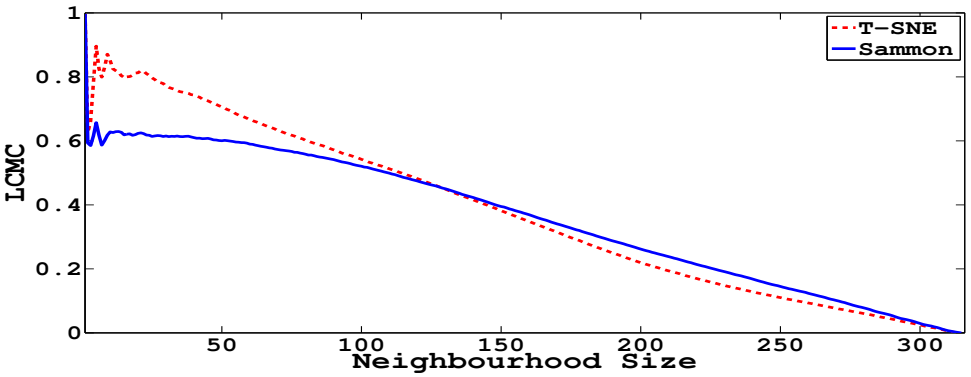
Figure B.3a shows the 3-dimensional observation space in a 2-dimensional view (since the third dimension is empty). The T-SNE visualisation is shown in figure B.3b and the Sammon visualisation in (B.3c).



(a) Q_{TC}

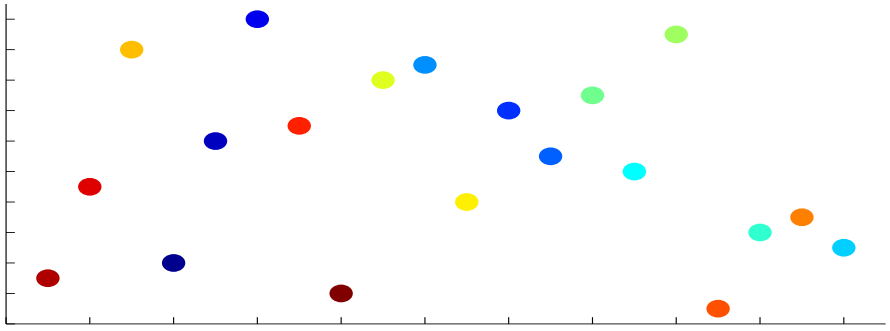


(b) Q_{MRRE}

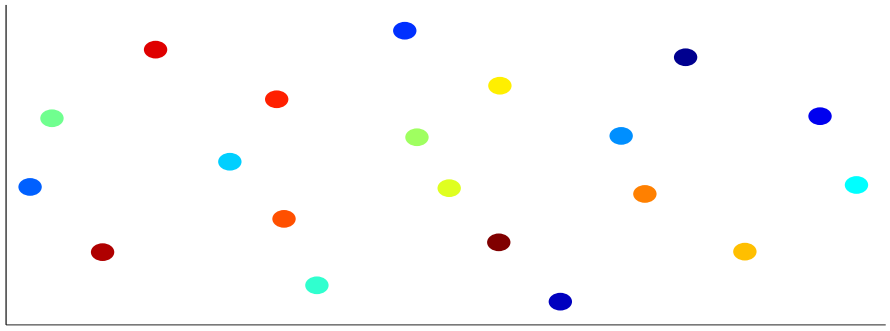


(c) *LCMC*

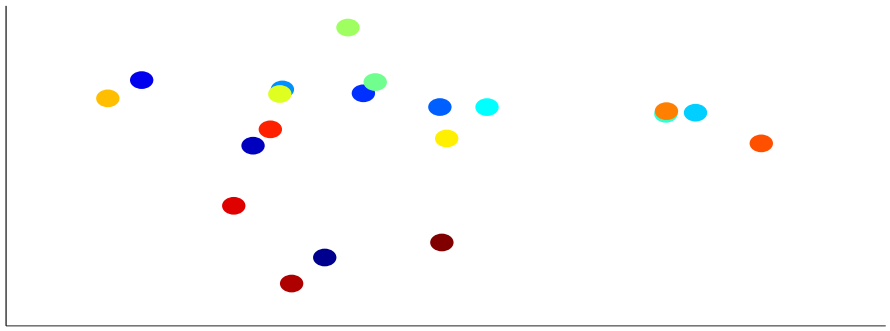
Figure B.2: Quality criterion: (a) Q_{TC}, (b) Q_{MRRE} and (c) LCMC for T-SNE and Sammon (figure 2.4) mappings of the Open Box dataset. The Q_{TC} results for the Sammon visualisation is far better than the T-SNE mapping for neighbourhoods greater than 60 represting the inter-face neighbourhoods (each box face consists of 60 points on a 2-dimensional plane). This worse result for the trustworthiness in T-SNE is caused by the tearing of the sides and unfolding motion of the mapping compared to the more rigid structure of the Sammon visualisation. The rank-based performance in terms of Q_{MRRE} and LCMC for T-SNE is slightly better than that of the Sammon map since T-SNE focuses more on the reconstruction of close neighbourhoods. These results for T-SNE for neighbourhoods up to 150 are better than all other mappings considered in chapter 2.



(a) Observation Space



(b) T-SNE



(c) Sammon Map

Figure B.3: Visualisations of a randomly generated 2-dimensional dataset embedded in 3-dimensional space. (a) Original points in 3-dimensional space viewed from above, (b) T-SNE visualisation in 2-dimensional space and (c) Sammon visualisation. The Sammon reconstruction has preserved both the topological ordering of points and neighbourhood structures since points in the 2-dimensional space, e.g. the two brown points and the dark blue point in the bottom left of the 3-dimensional and Sammon mapping are still close neighbours. The Sammon visualisation has shrunk the relative size of many neighbourhood regions due to the local focus of the STRESS measure. The T-SNE generated 2-dimensional space appears to have randomly re-ordered the points with no logical reason for this. The algorithm was run multiple times achieving this mapping as optimum.

The Sammon mapping has preserved the topological ordering and local neighbourhoods of observations. Some points are placed closer or atop one-another which is caused by the local focus of the Sammon map. A more accurate representation could be found with the un-normalised metric MDS error, $E = \sum_{ij}(d_x(i,j) - d_y(i,j))^2$. On the other hand T-SNE appears to order points in a filled circle, apparently randomly without preserving nearest neighbours or global neighbourhood structures.

For these three reasons it is concluded that T-SNE is not topographic and therefore not considered as a comparative method in this thesis. In its' defence T-SNE does perform very well in clustering tasks in 2-dimensional spaces, for example the MNist and CIFAR visualisations found in [118].

B.3 AutoEncoder

The AutoEncoder, as introduced in chapter 7, is a nonlinear extension to PCA using MLP networks. As such there is no constraint that the mapping is smooth and that neighbours are preserved, prevent the AE from mapping topographically. The nonlinear mapping also prevents it from having the dual relationship with MDS as PCA does. As such, it is also not included in this thesis as a visualisation algorithm.

B.4 Deep Gaussian Process

With the surge in research in deep learning since 2006 using stacked Autoencoders the parallel using stacked GPs was made in [13]. The training becomes very complex compared to that of the GPLVM. Model parameters are trained by optimising over the model evidence in a Bayesian framework:

$$\log[p(Y)] = log\left[\int p(Y|X)p(X|Z)p(Z)dXdZ\right],$$

which is analytically intractable. The integral above characterises a 3 layer deep GP with mappings from $Z \Rightarrow X \Rightarrow Y$. The optimisation procedure requires the use of a variational

lower bound to separate the integral and to introduce approximations to the distribution $p(Z)$. The asymptotic behaviour of the deep GP is analysed in [119] where it is shown that arbitrarily deep structures such as this show deficiencies. The dependence of each layer rests almost entirely on the layer above and the final layer Z often bears no resemblence to the original observations. The solution proposed to solve this deficiency allows the deep GP to map complex nonlinear manifolds (still dependent on diagonal covariance matrices and independent observations and latents). Although it is a complicated mathematical solution to the problem of deep learning it is incapable of topographically mapping observations the same way as the GPLVM with back constraints is and therefore is not discussed further in this thesis.

C Optimisation of GTM

This appendix details the Expectation-Maximisation (EM) algorithm optimisation of GTM. The distribution of observations, $p(\mathbf{y}_i|\mathbf{x},W)$ are spherical Gaussian kernels, $\mathcal{N}(\mathbf{m}(\mathbf{x},W),\beta^{-1}I)$. The precision of each Gaussian is β and the mean given by a parameterised mean function with weights W, $m(\mathbf{x},W)$. The distibution is therefore:

$$p(\mathbf{y}_i|\mathbf{x},W,\beta) = \left(\frac{\beta}{2\pi}\right)^{\frac{l}{2}} \exp\left[-\frac{\beta}{2}\|\mathbf{y}_i - \mathbf{m}(\mathbf{x},W)\|^2\right], \tag{C.1}$$

where l is the dimensionality of the observations. The prior distribution over the latent grid $p(\mathbf{x})$ is given by:

$$p(\mathbf{x}) = \frac{1}{c}\sum_{r=1}^{c}\delta(\mathbf{x}-\mathbf{g}(r)) = \begin{cases} 0 & \text{if } \mathbf{x} \neq \mathbf{g}(r), \\ \frac{1}{c} & \text{if } \mathbf{x} = \mathbf{g}(r), \end{cases} \tag{C.2}$$

where the c points $\mathbf{g}(r)$ are on a (rectangular) grid. Visualisation of the grid requires knowledge of $p(\mathbf{x}|\mathbf{y}, W, \beta)$ which by Bayes' rule is:

$$p(\mathbf{x}|\mathbf{y}_i, W, \beta) = \frac{p(\mathbf{y}_i|\mathbf{x}, W, \beta)p(\mathbf{x})}{p(\mathbf{y}_i|W, \beta)}.$$

In order to compute this posterior, the marginal likelihood must be calculated:

$$p(\mathbf{y}_i|W, \beta) = \int p(\mathbf{y}_i|\mathbf{x}, W, \beta)p(\mathbf{x})d\mathbf{x}.$$

This integral is typically analytically intractable for many prior choices but since the prior is a grid of delta points, the marginal likelihood becomes:

$$p(\mathbf{y}_i|W, \beta) = \frac{1}{c}\sum_{r=1}^{c} p(\mathbf{y}_i|\mathbf{g}(r), W, \beta).$$

The data log-likelihood is given by:

$$\mathcal{L}(W, \beta) = \sum_{i=1}^{N} \log\left(p(\mathbf{y}_i|W, \beta)\right).$$

The mean function in equation (C.1) is typically taken to be an RBF network as described in appendix A. Other extensions using Gaussian Processes (GPs) and mean field approximations for the marginal likelihood have also been proposed [42]. Using an RBF network in this framework allows for an Expectation-Maximisation (EM) optimisation procedure:

E-step

Calculate the responsibilities P of each latent point wrt. each observation i:

$$\begin{aligned} P_{i,r}(W, \beta) &= p(\mathbf{g}(r)|\mathbf{y}_i, W, \beta), \\ &= \frac{p(\mathbf{g}(r)|\mathbf{y}_i, W, \beta)p(\mathbf{g}(r))}{\sum_{s=1}^{c} p(\mathbf{g}(s)|\mathbf{y}_i, W, \beta)p(\mathbf{g}(s))}, \\ &= \frac{p(\mathbf{g}(r)|\mathbf{y}_i, W, \beta)}{\sum_{s=1}^{c} p(\mathbf{g}(s)|\mathbf{y}_i, W, \beta)}, \\ &= \frac{\mathcal{N}\left(\mathbf{y}_i|\mathbf{m}(\mathbf{g}(r), W), \beta^{-1}I\right)}{\sum_{s=1}^{c} \mathcal{N}\left(\mathbf{y}_i|\mathbf{m}(\mathbf{g}(s), W), \beta^{-1}I\right)}. \end{aligned}$$

Finally the diagonal matrix G is given by:

$$G_{r,r}(W,\beta) = \sum_{i=1}^{N} P_{i,r}(W,\beta),$$

which defines the total responsibility of each latent point.

M-step

The parameters W,β are updated as follows:

- $\hat{W} = YP\Phi^T(\Phi G\Phi^T)^{-1}$ where Φ is the matrix of basis functions from the RBF network.
- $(\hat{\beta}^{-1}) = \frac{1}{Nc}\sum_{r=1}^{c}\sum_{i=1}^{N} P_{ir}(\hat{W},\beta)\|\mathbf{y}_i - \mathbf{m}(\mathbf{g}(r),\hat{W})\|^2$.

Convergence can typically be achieved following a few tens of iterations [120].

D Derivation of Fisher Information

This appendix derives Fisher Infromation and the Cramer-Rao bound from first principles. The derivation is taken from [67, p. 29-30].

Consider the class of estimators $\hat{\theta}(\mathbf{z})$ that are unbiased, obeying:

$$\langle \hat{\theta}(\mathbf{z}) - \theta \rangle \equiv \int d\mathbf{z}[\hat{\theta}(\mathbf{z}) - \theta] p(\mathbf{z}|\theta) = 0. \tag{D.1}$$

The PDF $p(\mathbf{z}|\theta)$ describes the fluctuations in data values $\mathbf{z}$ in the presence of the parameter value θ. PDF $p(\mathbf{z}|\theta)$ is called the "likelihood". Differentiation of equation (D.1) with respect to θ gives:

$$\int d\mathbf{z}(\hat{\theta} - \theta)\frac{\partial p}{\partial \theta} - \int d\mathbf{z} p = 0. \tag{D.2}$$

Using the identity:

$$\frac{\partial p}{\partial \theta} = p\frac{\partial \log p}{\partial \theta},$$

and the fact that p obeys normalisation such that the right term in equation (D.2) is equal to one, equation (D.2) becomes:

$$\int d\mathbf{z}(\hat{\theta}-\theta)\frac{\partial \log p}{\partial \theta}p = 1. \tag{D.3}$$

By splitting the integrand into two seperate factors we have that:

$$\int d\mathbf{z}\left[\frac{\partial \log p}{\partial \theta}\sqrt{p}\right]\left[(\hat{\theta}-\theta)\sqrt{p}\right] = 1. \tag{D.4}$$

Squaring equation (D.4) and using the Schwarz inequality gives:

$$\left[\int d\mathbf{z}\left(\frac{\partial \log p}{\partial \theta}\right)^2 p\right]\left[\int d\mathbf{z}(\hat{\theta}-\theta)^2 p\right] \geq 1. \tag{D.5}$$

The left factor is defined to be the Fisher Information, $I(\theta)$:

$$I(\theta) \equiv \int d\mathbf{z}\left(\frac{\partial \log p}{\partial \theta}\right)^2 p \equiv \left\langle\left(\frac{\partial \log p}{\partial \theta}\right)^2\right\rangle, p \equiv p(\mathbf{z}|\theta), \tag{D.6}$$

where the notation $\langle\rangle$ is an average operator. The right factor in equation (D.5) defines the mean-square error:

$$e^2 \equiv \int d\mathbf{z}[\hat{\theta}(\mathbf{z})-\theta]^2 p \equiv \langle[\hat{\theta}(\mathbf{z})-\theta]^2\rangle. \tag{D.7}$$

Substituting equations (D.6), (D.7) into equation (D.5) gives the important result:

$$e^2 I(\theta) \geq 1,$$

which is more commonly stated as the Cramer-Rao bound:

$$e^2 \geq \frac{1}{I(\theta)}. \tag{D.8}$$

Practically this means that the variance of an unbiased estimator of θ is bounded by the inverse of the Fisher Information of $p(\mathbf{z}|\theta)$. This inequality holds for the multiparameter

case, where the covariance of an unbiased estimator of $\boldsymbol{\theta}$ based on an observation system $T(X)$:

$$cov_{\boldsymbol{\theta}}(T(X)) \geq I(\theta)^{-1}. \tag{D.9}$$

In the A-optimality scheme used in section 4.3 the trace of the inverse of the information matrix, $I(\boldsymbol{\theta})_{ij}$, is taken as a suitable lower bound to the error:

$$\sum_{ij} cov_{\theta}(T(X))_{ij} \geq tr\left(I(\boldsymbol{\theta})^{-1}\right). \tag{D.10}$$

This reduces the computational complexity and neglects poor estimators of cross-covariances, i.e. $I(\theta)_{ij}$ for $j \neq i$. The mapping uncertainty definition in section 4.3 is reliant on the Fisher Information, given by equation (D.6), and the A-optimality approximation to the Cramer-Rao bound in equation (D.10).

E Gradients for SONAR Noise Model

This chapter describes the gradients used for optimistation of the SONAR compound mixture model outlined in section 6.5.

The known physical noise sources from the SONAR environment, and their respective probability distributions, are combined to represent residuals from a background additive noise perspective. A five-order compound mixture model is used for this task, consisting of:

1. Extraneous signals - Laplace distribution. Any prominent signals not well fit by the NAR model should appear in a small area of residual space and as such the sharply peaked Laplace is an appropriate choice.

2. Rain - Gamma distribution. Typically in the literature the poisson distribution is used to describe rain [88] however, the other distributions in this mixture are continuous and therefore the Gamma, which behaves similarly to the Poisson, is chosen.

3. Clutter - K and Rayleigh distributions. It is well etablished in the SONAR literature that K and Rayleigh distributions are suitable for describing background clutter such as biological and environmental effects in the underwater environment [87].

4. Remainder - Normal. Any leftover residual elements can be fit with a Normal distribution.

This model can be implemented in a Maximum Likelihood (ML) or, provided appropriate priors are specified, a more robust Maximum-a-Posteriori (MAP) scheme. This appendix outlines the MAP mixture gradient descent process.

The mixtures are fit using a hybrid version of gradient descent. This is required since optimising the parameters of the K-distribution using gradient descent often leads to unrealistic parameters [89]. In order to remedy this, the parameters are fit to the residuals in a Bayesian scheme introduced in [90] before fitting the remaining parameters, mixture weights and hyperparameters, in the MAP mixture case, using gradient descent over the negative log-likelihood.

The priors specified for each distribution were initially set as the conjugate priors, however some of these were found to be a poor fit for the optimised parameters on real datasets. As such, the distributions were chosen as to be representative of the real world data described in section 6.5. It should be noted that there are no priors specified over the mixture weights as this has only been found to slow convergence of the gradient learning procedure.

Firstly the individual distributions and their relevant priors will be defined, following which the gradient learning procedure will be described.

E.1 The Model

Laplace distribution

The PDF of the Laplace distribution is:

$$P(x|\mu,b) = \frac{1}{2b}\exp\left[-\frac{|x-\mu|}{b}\right].$$

The priors over μ and b are Laplace and Gamma distributions respectively, i.e:

- $P(x|\mu,b) = Laplace(x|\mu,b)$,
- $P(\mu|\mu_0,b) = Laplace(\mu|\mu_0,b)$,
- $P(b|\alpha_b,\beta_b) = Gamma(b|\alpha_b,\beta_b)$.

Rayliegh distribution

The PDF of the Rayleigh distribution is:

$$P(x|\sigma) = \frac{x}{\sigma^2}\exp\left[-\frac{x^2}{2\sigma^2}\right].$$

The prior over σ is a Gamma distribution:

- $P(x|\sigma) = Rayleigh(x|\sigma)$,
- $P(\sigma|a_\sigma,b_\sigma) = Gamma(\sigma|a_\sigma,b_\sigma)$.

K-distribution

The K-distribution is optimised using the procedure from [90].

Gamma distribution

The PDF of the Gamma distribution used in the SONAR noise model is:

$$P(x|\alpha,\beta) = \frac{1}{\Gamma(\alpha)}\beta^{-\alpha}x^{(\alpha-1)}\exp(-\frac{x}{\beta}).$$

The prior over β is again a Gamma distribution:

- $P(x|\alpha,\beta) = Gamma(x|\alpha,\beta)$,
- $P(\beta|a_\beta,b_\beta) = Gamma(\beta|a_\beta,b_\beta)$.

There is no closed form conjugate prior for α and the distribution over α appears to change on each fit. As such, there is no prior over α.

Normal distribution

The PDF of the Normal distribution is given by:

$$P(x|m,s) = \frac{1}{\sqrt{2\pi s^2}} \exp\left[-\frac{(x-m)^2}{2s^2}\right].$$

The priors over m and s are Normal and Gamma distributions respectively:

- $P(x|m,s) = \mathcal{N}(x|m,s^2)$,
- $P(m|m_0,s) = \mathcal{N}(m|m_0,s^2)$,
- $P(s^2|a_s,b_s) = Gamma(s^2|a_s,b_s)$.

Likelihood

The probabilistic model can now be written as the maximisation of:

$$P(x|\theta_p)P(\theta_p|\theta_h)$$

where $P(x|\theta_p)$ is the data likelihood of x given the parameters θ_p and $P(\theta_p|\theta_h)$ is the prior distribution of the parameters, θ_p, given the hyperparameters, θ_h. The data-log likelihood is given as:

$$\begin{aligned} P(x|\theta_p) = \quad & \textstyle\sum_i [\log(\pi_1 Laplace(x_i|\mu,b) + \pi_2 Rayleigh(x_i|\sigma) + \pi_3 K + \pi_4 Gamma(x_i|\alpha,\beta) \\ & + \pi_5 \mathcal{N}(x_i|m,s^2) + N(\textstyle\sum_j \pi_j - 1))], \end{aligned}$$

where π_j are the mixture weights and $N(\sum_j \pi_j - 1)$ relates to the lagrange multiplier used to ensure that the $\pi'_j s$ sum to unity.

Optimisation in the Maximum-a-Posteriori (MAP) context involves maximising the posterior distribution, typically ignoring the effects of the unknown distribution $P(x)$, known as the model evidence. In reality it is often the case that the negative log-likelihood and negative log-prior distributions are analysed. Denoting the set of parameters and hyperparameters $\theta = \{\theta_p, \theta_h\}$ the optimisation involves evaluating the gradients for each θ^i:

$$\frac{-\partial \log P(x|\theta_p)}{\partial \theta^i} + \frac{-\partial \log P(\theta_p|\theta_h)}{\partial \theta^i},$$

in order to find a minimum. This is often performed using a nonlinear gradient optimiser, such as SCG. It should be noted that many of the above distributions require the parameters to be greater than zero and as such the computational implementation of the derivative was taken with respect to the square root of the parameter, following which it was then squared. This notation is ommitted here to keep the expressions as simple as possible.

E.2 The Gradients

Laplace parameters and hyperparameters

The parameters and hyperparameters relevant to the Laplace distribution are $\{\mu, b, \mu_0, \alpha_b, \beta_b\}$ with the following derivatives:

- $\frac{\partial}{\partial \mu} = \begin{cases} -\frac{1}{P(x|\theta_p)} \left(\frac{1}{2b^2} \exp\left[\frac{|x-\mu|}{b} \right] \right) \pi_1 - (-1)^h \frac{1}{b}, & \text{if } x \geq \mu, \\ \frac{1}{P(x|\theta_p)} \left(\frac{1}{2b^2} \exp\left[\frac{|x-\mu|}{b} \right] \right) \pi_1 - (-1)^h \frac{1}{b}, & \text{if } x < \mu, \end{cases}$
- $\frac{\partial}{\partial b} = \frac{1}{P(x|\theta_p)} \left(-\frac{1}{2b^2} + \frac{|x-\mu|}{2b^3} \right) \exp(\frac{-|x-\mu|}{b}) \pi_1 - \left[\frac{1}{b} - \frac{|\mu-\mu_0|}{b^2} + \frac{1}{2\beta_b^2}(2b - 2\alpha_b) \right],$
- $\frac{\partial}{\partial \mu_0} = -\frac{1}{b},$
- $\frac{\partial}{\partial \alpha_b} = \frac{1}{2\beta_b^2}(-2b + 2\alpha_b).$

where h is the Heaviside function, $h = 1$ if $\mu < \mu_0$ and $h = 0$ if $\mu \geq \mu_0$.

Rayleigh parameters and hyperparameters

The parameters and hyperparameters relevant to the Rayleigh distribution are $\{\sigma, a_\sigma, b_\sigma\}$ with the following derivatives:

- $\frac{\partial}{\partial \sigma} = \frac{1}{P(x|\theta_p)} \left(\frac{x^3}{\sigma^5} - \frac{2x}{\sigma^3} \right) \exp\left[\frac{-x^2}{2\sigma^2} \right] \pi_2 - \left[\frac{b_\sigma}{\sigma^2} + \frac{a_\sigma}{\sigma} + \frac{1}{\sigma} \right]$,
- $\frac{\partial}{\partial a_\sigma} = -\log(\frac{b_\sigma}{\sigma}) + \Psi(a_\sigma)$,
- $\frac{\partial}{\partial b_\sigma} = \frac{1}{\sigma} - \frac{a_\sigma}{b_\sigma}$.

Gamma parameters and hyperparameters

The parameters and hyperparameters relevant to the Gamma distribution are $\{\alpha, \beta, a_\beta, b_\beta\}$ with the following derivatives:

- $\frac{\partial}{\partial \alpha} = \frac{1}{P(x|\theta_p)} \left(\log(x) - \log(\beta) - \Psi(\alpha) \right) \left(\frac{1}{\Gamma(\alpha)} x^{(\alpha-1)} \beta^{-\alpha} \exp\left[\frac{-x}{\beta} \right] \right) \pi_4$,
- $\frac{\partial}{\partial \beta} = \frac{1}{P(x|\theta_p)} \left(\frac{1}{\Gamma(\alpha)} \exp\left[\frac{-x}{\beta} \right] \left(x^\alpha \beta^{(-2-\alpha)} - x^{-1+\alpha} \alpha \beta^{-1-\alpha} \right) \right) \pi_4 + \left[-\frac{(a_\beta - 1)}{\beta} + b_\beta \right]$,
- $\frac{\partial}{\partial a_\beta} = -\log(\beta) - \log(b_\beta) + \Psi(a_\beta)$,
- $\frac{\partial}{\partial b_\beta} = \beta - \frac{a_\beta}{b_\beta}$.

Normal parameters and hyperparameters

The parameters and hyperparameters relevant to the Normal distribution are $\{m, s, m_0, a_s, b_s\}$ with the following derivatives:

- $\frac{\partial}{\partial m} = \frac{1}{P(x|\theta_p)} \left(\frac{(x-m)}{\sqrt{2\pi} s^3} \exp\left[\frac{-(x-m)^2}{2s^2} \right] \right) \pi_5 - \frac{(m-m_0)}{s^2}$,
- $\frac{\partial}{\partial s^2} = \frac{1}{P(x|\theta_p)} \left(\frac{-1}{\sqrt{2\pi} s^2} + \frac{(x-m)^2}{s^4} \right) \exp\left[\frac{-(x-m)^2}{2s^2} \right] \pi_5 + \left[\frac{1}{2s^2} - \frac{(m-m_0)^2}{2s^4} - \frac{b_s}{s^4} + \frac{a_s}{s^2} + \frac{1}{s^2} \right]$,
- $\frac{\partial}{\partial m_0} = \frac{(m_0 - m)}{s^2}$,
- $\frac{\partial}{\partial a_s} = -\log(\frac{b_s}{s^2}) + \Psi(a_s)$,
- $\frac{\partial}{\partial b_s} = \frac{1}{s^2} - \frac{a_s}{b_s}$.

Mixture Coefficients

The derivatives with respect to the mixture coefficients, $\{\pi_j, j = 1:5\}$ are given by:

- $\frac{\partial}{\partial \pi_1} = \frac{\text{Laplace}}{P(x|\theta_p)} - N,$
- $\frac{\partial}{\partial \pi_1} = \frac{\text{Rayleigh}}{P(x|\theta_p)} - N,$
- $\frac{\partial}{\partial \pi_1} = \frac{\text{K}}{P(x|\theta_p)} - N,$
- $\frac{\partial}{\partial \pi_1} = \frac{\text{Gamma}}{P(x|\theta_p)} - N,$
- $\frac{\partial}{\partial \pi_1} = \frac{\mathcal{N}}{P(x|\theta_p)} - N,$

where the left term in each derivative is the relative responsibility of the mixture weight and the N (number of datapoints) term is given by the Lagrange multiplier constraint. These terms for the parameters and hyperparameters can be used by any standard nonlinear optimisation algorithm to fit the noise distribution outlined in section 6.5, the implementation in chapter 6 uses SCG.

F Gradients for the Cascading RBF

This appendix derives the gradients for the 3-layer Cascading RBF outlined in chapter 7. It also uses these gradients to show that the Cascading RBF is a smooth mapping.

Starting with the initial error:

$$E(Y) = \frac{1}{2}\sum_{i=1}^{N} \|Y_i - T_i\|^2, \tag{F.1}$$

the gradients for Λ^3, Λ^2 and Λ^1 are given as follows.

Λ^3 Gradients

The gradients for Λ^3 which map from X^2 to Y are given by:

$$\frac{\partial E(Y)}{\partial Y(\Phi^2)} = (Y - T),$$

$$\frac{\partial Y}{\partial \Lambda^3} = \Phi^2.$$

In the cascaded Shadow Targets optimisation procedure outlined in chapter 7 the gradient with respect to Λ^3 is not explicitly required, since the MSE optimum parameters are given by pseudo-inverse of Φ^2 as in standard RBF learning.

Λ^2 Gradients

The gradients for Λ^2 which map from X^1 to X^2 are given by:

$$\frac{\partial E}{\partial Y(\Phi^2)} = (Y - T),$$

$$\frac{\partial Y}{\partial \Phi^2(D)} = \Lambda^3,$$

$$\frac{\partial \Phi^2}{\partial D(X^2, C^2)} = \frac{3}{2}(D)^{\frac{1}{2}},$$

$$\frac{\partial D}{\partial X^2(\Phi^1)} = \sum_i X_i^2 - C_j^2,$$

$$\frac{\partial X^2}{\partial \Lambda^2} = \Phi^1.$$

This is assuming the form of Φ and D are those specified in chapter 7, namely that $\phi = z^{\frac{3}{2}}$ and D is given by the square Euclidean distance.

Λ^1 Gradients

The gradients for Λ^1 which map from X^0 to X^1 are given by:

$$\frac{\partial E}{\partial X^2(\Phi^1)} = (Y - T)\left(\Lambda^3\right)\left(\frac{3}{2}(D)^{\frac{1}{2}}\right)\left(\sum_i X_i^2 - C_j^2\right).$$

The three right hand terms are the recursive elements which can be used successively when constructing arbitrarily deep layers.

$$\frac{\partial X^2}{\partial \Phi^1(D)} = \Lambda^2,$$

$$\frac{\partial \Phi^1}{\partial D(X^1, C^1)} = \frac{3}{2}(D)^{\frac{1}{2}},$$

$$\frac{\partial D}{\partial X^1(\Phi^0)} = \sum_i X_i^1 - C_j^1,$$

$$\frac{\partial X^1}{\partial \Lambda^1} = \Phi^0.$$

This is again assuming the form of Φ and D are those specified in chapter 7, namely that $\phi = z^{\frac{3}{2}}$ and D is given by the square Euclidean distance for both layers.

CRBF Smoothness

In order to prove that the CRBF is smooth it must be shown to be infinitely differentiable with respect to the inputs, X^0. The layer-wise gradients are given by:

$$\frac{\partial E}{\partial X^1(\Phi^0)} = (Y - T)\left(\Lambda^3\right)\left(\frac{3}{2}(D)^{\frac{1}{2}}\right)\left(\sum_i X_i^2 - C_j^2\right)\left(\Lambda^2\right)\left(\frac{3}{2}(D)^{\frac{1}{2}}\right)\left(\sum_i X_i^1 - C_j^1\right),$$

$$\frac{\partial X^1}{\partial \Phi^0} = \Lambda^1,$$

$$\frac{\partial \Phi^0}{\partial D(X^0, C^0)} = \frac{3}{2}(D)^{\frac{1}{2}},$$

$$\frac{\partial D}{\partial X^0} = \sum_i X_i^0 - C_j^0.$$

Each of these parts are in-themselves continuous and as such the product will be continuous. The derivatives of higher orders could be calculated to prove the infinite differentiability, however this is not required. It is clear that, since the basis function used to construct Φ is infinitely differentiable, that the gradients found using the chain rule will not allow $\frac{\partial \Phi^i}{\partial D(X^i, C^i)}$ to vanish. This is seen in the Cascading RBF used in chapter 7 with $\phi(z) = z^{\frac{3}{2}}$. This will ensure that the gradient at each layer exists and as such it is infinitely differentiable with respect to X^0, ensuring smoothness.