Data Visualisation with Missing Data: A Non-Linear Approach

Martin Schroeder[*]and Dan Cornford

NCRG Research Group
Aston University, Aston Triangle, Birmingham, B4 7ET, UK

October 12, 2007

Abstract

Exploratory analysis of data in all sciences seeks to find common patterns to gain insights into the structure and distribution of the data. Typically, visualisation methods like principal components analysis are used but these methods are not easily able to deal with missing data nor can they capture non-linear structure in the data. One approach to discovering complex, non-linear structure in the data is through the use of multivariate statistics,linked plots, or brushing, while deleting or ignoring the missing data. In this technical report we discuss a complementary approach based on a non-linear probabilistic model. The generative topographic mapping enables the visualisation of the effects of very many variables on a single plot, which is able to incorporate far more structure than a two dimensional principal components plot could, and deal at the same time with missing data. We show that using the generative topographic mapping provides us with an optimal method to explore the data while being able to replace missing values in a dataset, particularly where a large proportion of the data is missing.

Tech Report NCRG/2007/04

[*] Corresponding author. Phone: +44 (0) 7847180216 // Email: shroderm@aston.ac.uk

Contents

1 Introduction

In all sciences the amount of available data is steadily growing. Increasing capabilities of the analysis methods used and decreasing costs for the capture, processing and storage of data is likely to further enhance this trend in the future. For example, modern technologies allow samples in chemistry and geology to be analysed for chemical composition or biomarkers in great detail. In molecular biology microarray analyses allow access to large quantities of data either stored in the DNA of living organisms or in terms of the reactivity of new found biologically active compounds.

To make use of these data one needs to explore their structure and try to find common patterns to generalise trends or infer further information from them. To find these patterns, visual inspection using a single plot is only an option in three dimensions (3D). Arguably this can be extended to 6D by adding colour, marker size and marker type to a 3D plot, however this can be very difficult to interpret. Linked plots and sub-setting methods can be further employed, however discovering non-trivial relationships in high dimensional complex data remains an open problem. To address this issue one can use visualisation of high dimensional data and map, or project, the high dimensional data onto a low dimensional representation while preserving as much information about the structure, in the high dimensional space, as possible. This low dimensional representation is usually 2D to be representable on screen or paper. Employing this 2D representation allows the human analyst to explore the data and discern structure more easily and naturally and a good example for this is Principal Components Analysis (PCA) [6].

The task of exploring and analysing data gets even more complicated if one has to deal with missing values. These represents a challenge in many scientific fields. Many existing analysis and visualisation methods like PCA cannot cope with missing values. The standard options that are typically employed include the deletion of samples with missing values or replacing missing values with the averages of columns and rows. Using these methods one looses a lot of information in the best case or changes the data distributions affecting the analysis in the worst. Thus finding a good method for the *imputation* of missing values is important in obtaining better and more reliable results across a range of analyses. In addition this should make the visualisation and interpretation easier and more reproducible.

In this technical report,we compare PCA, which is a widely employed method and provides a benchmark, with the Generative Topographic Mapping (GTM) [1] which is a more complex, flexible model. GTM offers a significant advantage over many other visualisation methods as its probabilistic formulation allows the natural incorporation of methods to deal with missing and uncertain data. We start with an overview of PCA and GTM and a discussion of their functionalities and limitations, illustrated on a toy data set. To illustrate the potential of the GTM model to provide a natural framework for replacing missing values we compare it against other imputation methods on another toy data set. We conclude with a summary of the benefits and limitations of GTM, and provides some suggestions for areas requiring further research.

2 Data Visualisation

To address visualisation of high dimensional data is to map, or project, the high dimensional data onto a low dimensional representation while preserving as much information about the structure in the high dimensional space as is possible. This low dimensional representation is usually 2D to be representable on screen or paper and will be referred to as the *latent space*. Employing a 2D latent space allows the human analyst to explore the data and discern structure more easily and naturally. There are a many possible ways to obtain such a low dimensional representation. Context will often guide the approach, together with the manner in which the latent space representation will be employed.

Some methods such as PCA and factor analysis [3] linearly transform the data space and project the data onto the lower dimension space while retaining the maximum information [1]. Other methods like Kohonen, or Self Organising, Maps [8] and the Generative Topographic Mapping (GTM) [1] try to capture the topology[2] of the data. Geometry preserving methods like multidimensional scaling and Neuroscale [10] try to find a representation in latent space which preserves the geometric distances between the data points.

2.1 PCA

The most used and traditional method for dimension reduction and thus visualising is the direct orthogonal projection of a point in $\Re^D$ onto a hyperplane in $\Re^L \subset \Re^D$ with $L \leq D$. Principal component analysis [6] takes a data set $\mathbf{T} = \mathbf{t}_1, ..., \mathbf{t}_N$ and finds a new orthonormal basis $\mathbf{u}_1, ..., \mathbf{u}_D$ with its axes ordered in such a manner that the first axis explains the largest variance in $\mathbf{T}$. The second axis is orthogonal to the first and accounts for a maximum of the remaining variance in the data and the subsequent axes follow this by iteration.

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

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

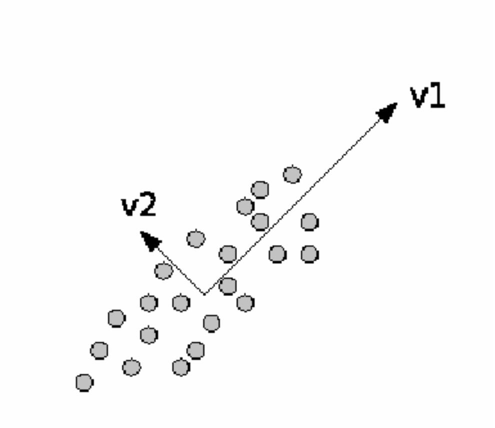


Figure 1: **PCA:** The plot shows a random data set together with the 2 principal components (v1,v2).

Given that the set of observations are centered, $\sum_n^N \mathbf{t}_n = 0$, PCA will find the principal components by diagonalising the covariance matrix,

$$\mathbf{C} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{t}_n \mathbf{t}_n^T \tag{1}$$

and then finding its eigen-structure

$$\mathbf{CU} = \mathbf{U}\boldsymbol{\Lambda} \tag{2}$$

$\mathbf{U}$ is a $D \times D$ matrix which has the unit length eigenvectors,$\mathbf{u}_1, ..., \mathbf{u}_D$, as its columns and $\boldsymbol{\Lambda}$ is a diagonal matrix with the corresponding eigenvalues,$\lambda_1, ..., \lambda_D$, along the diagonal. The eigenvectors are the principal components and the eigenvalues are the corresponding variances.

This technique may be used to project higher dimensional data onto a two-dimensional hyperplane to visualise it on a screen or to cut down the dimensionality to three or more dimensions. Commonly the first two principal components are used for the projection since they explain most of the original information as they preserve the most variance from the full data set. The property that the PCA defines a linear transformation gives it fast and stable computational characteristics, but it is also its main draw back since any non-linear structures in the data cannot be captured.

2.2 Generative Topographic Mapping

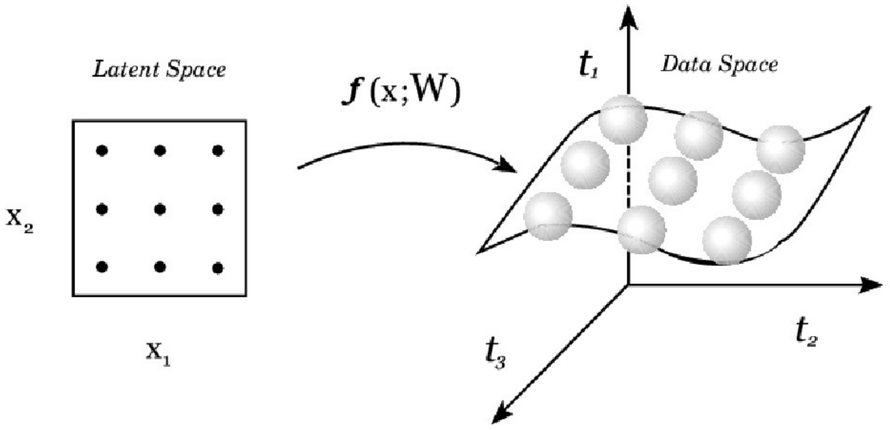


Figure 2: The non-linear function $\Theta(\mathbf{x}, \mathbf{W})$ defines a manifold S embedded in the data space given by the image of the latent variable space under the mapping $\mathbf{x} \rightarrow \mathbf{y}$. *(C.M. Bishop,1997)*

The essence of GTM is to try and fit a *density model*, which is constrained to lie on a 1 or 2 dimensional manifold, to the data in order to capture the structure in the high dimensional data space. This can be visualised as a flexible rubber sheet, typically 2 dimensional, which is being bent and stretched in the high dimensional space to best fit the data points. This rubber sheet consist of a grid of points in the latent space which are connected via a non-linear mapping function to a contorted grid in the data space. To learn the intrinsic structure in the data, the rubber sheet is distorted by learning the non-linear mapping function using an Expectation Maximisation (EM) [4] so that the model best explains the data.

In contrast to many other latent variable models the GTM algorithm is not defined in terms of a mapping from the data space into the latent space, but rather it defines a mapping from latent to data space and applies Bayes theorem to induce a posterior distribution in the latent space given some new data.

First one considers a function $\mathbf{y} = \Theta(\mathbf{x}, \mathbf{W})$ which maps points $\mathbf{x}$ in the L-Dimensional latent space into an L-dimensional non-Euclidean manifold S

embedded within the D-Dimensional data space onto the points $\mathbf{y}$, shown for $L = 2$ and $D = 3$ in Figure 3.

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

$$p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \beta) = (\frac{\beta}{2\pi})^{D/2} \exp\{-\frac{\beta}{2} \parallel \Theta(\mathbf{x}, \mathbf{W}) - \mathbf{y} \parallel^2\} \tag{3}$$

where one should note that it is possible to use other models for $p(\mathbf{y}|\mathbf{x})$ like Bernoulli for binary variables or a combination of different models if necessary. For a given value of $\mathbf{W}$, the distribution is obtained by integration over the $\mathbf{x}$-distribution

$$p(\mathbf{y}|\mathbf{W}, \beta) = \int p(\mathbf{y}|\mathbf{x}, \mathbf{W}, \beta) p(\mathbf{x}) dx \ . \tag{4}$$

For a given data set $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_N)$ of N data points, the parameter matrix $\mathbf{W}$ and the inverse variance β is obtained through the maximum likelihood method. This can be done via maximising the log likelihood, given by

$$L(\mathbf{W}, \beta) = \ln \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{W}, \beta) \ . \tag{5}$$

After determining the prior distribution $p(\mathbf{x})$ and the functional form of the mapping $\Theta(\mathbf{x}, \mathbf{W})$ it is in principle possible to determine β and $\mathbf{W}$ by maximising $L(\mathbf{W}, \beta)$. But the integral over $\mathbf{x}$ in (4) will, in general, be analytically intractable. Also one might wish to use a non-linear function $\Theta(\mathbf{x}, \mathbf{W})$. Therefore a specific form of $p(\mathbf{x})$ is considered, where $p(\mathbf{x})$ is given by a sum of delta functions centered on the nodes of a regular grid in latent space

$$p(\mathbf{x}) = \frac{1}{K} \sum_{i=1}^{K} \delta(\mathbf{x} - \mathbf{x}_i) \tag{6}$$

in which case the integral in (3) can be evaluated analytically. Now every point $\mathbf{x}_i$ is mapped to a corresponding point $\Theta(\mathbf{x}_i, \mathbf{W})$ in the data space, where it builds the center of a Gaussian density function. Combining (4) and (6) the distribution function in the data space takes the form

$$p(\mathbf{y}|\mathbf{W}, \beta) = \frac{1}{K} \sum_{i=1}^{K} p(\mathbf{y}|\mathbf{x}_i, \mathbf{W}, \beta) \tag{7}$$

and the corresponding log likelihood becomes

$$L(\mathbf{W}, \beta) = \sum_{n=1}^{N} \ln\{\frac{1}{K} \sum_{i=1}^{K} p(\mathbf{y}_n | \mathbf{x}_i, \mathbf{W}, \beta)\} \ . \tag{8}$$

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

$$\Theta(\mathbf{x}, \mathbf{W}) = \mathbf{W}\Phi(\mathbf{x}) \tag{9}$$

with the elements of $\Phi(\mathbf{x})$ consisting of M fixed radial basis functions [2] $\Phi_j(\mathbf{x})$ and $\mathbf{W}$ being a $D \times M$ matrix.

In the case under consideration it is assumed that the hidden variable i from (6) tells which component generated each data point $\mathbf{y}_n$. Therefore the EM algorithm can be formulated as follows. Assuming that $\mathbf{W}_{old}$ and β_{old} are given one can use the E-step to evaluate the posterior probabilities of each Gaussian component i for every data point $\mathbf{y}_n$ using Bayes theorem

$$\begin{aligned} R_{in}(\mathbf{W}_{old}, \beta_{old}) &= p(\mathbf{x}_i | \mathbf{y}_n, \mathbf{W}_{old}, \beta_{old}) && (10) \\ &= \tfrac{p(\mathbf{x}_i | \mathbf{y}_n, \mathbf{W}_{old}, \beta_{old})}{\sum_{j=1}^{K} p(\mathbf{x}_j | \mathbf{y}_n, \mathbf{W}_{old}, \beta_{old})} \ . && (11) \end{aligned}$$

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

$$\langle L_{comp}(\mathbf{W}, \beta) \rangle = \sum_{n=1}^{K} \sum_{i=1}^{N} R_{in}(\mathbf{W}_{old}, \beta_{old}) \ln\{p(\mathbf{y}_n | \mathbf{x}_i, \mathbf{W}, \beta)\} \ . \tag{12}$$

Maximising (12) with respect to W and using (3) and (9) one gets

$$\sum_{n=1}^{K} \sum_{i=1}^{N} R_{in}(\mathbf{W}_{old}, \beta_{old}) \{\mathbf{W}_{new}\Phi(\mathbf{x}_i) - \mathbf{y}_n\} \Phi^T(\mathbf{x}_i) \ . \tag{13}$$

This can be written in matrix notation

$$\boldsymbol{\Phi}^T \mathbf{G}_{old} \boldsymbol{\Phi} \mathbf{W}_{new}^T = \boldsymbol{\Phi}^T \mathbf{R} \mathbf{Y} \tag{14}$$

with $\boldsymbol{\Phi}$ being a $K \times M$ matrix with elements $\Phi_{ij} = \Phi_j(x_i)$, $\mathbf{Y}$ being a $N \times D$ matrix with elements y_{nk}, $\mathbf{R}$ being a $K \times N$ matrix with elements R_{in} and$\mathbf{G}$ being a $K \times K$ diagonal matrix with elements

$$G_{ii} = \sum_{n=1}^{N} R_{in}(\mathbf{W}_{old}, \beta_{old}) \ . \tag{15}$$

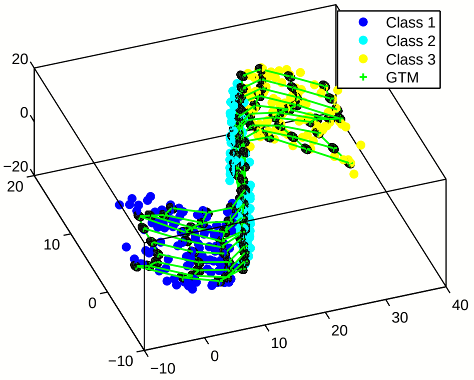


Figure 3: Fitting a GTM with a 20x20 grid in 2D latent space to 3D S-shaped data. Plotted is the result after 50 iterations with the EM algorithm.

Equation (14) can be solved for $\mathbf{W}_{new}$ using standard matrix inversion techniques. In similar steps to maximise (12) with respect β one obtains the following formula

$$\frac{1}{\beta_{new}} = \frac{1}{ND} \sum_{n=1}^{N} \sum_{i=1}^{N} R_{in}(\mathbf{W}_{old}, \beta) ||\mathbf{W}_{new}\Phi(\mathbf{x}_i) - \mathbf{y}_n||^2 . \tag{16}$$

The EM algorithm alternates between the E-step,given by evaluating 10, and the M-Step,evaluating $\mathbf{W}_{new}$ and β_{new}, until it converges to a (local) maximum and can be written like that:

E-Step:

- Set $\mathbf{W}_{old} = \mathbf{W}_{new}$ and $\beta_{old} = \beta_{new}$
- Calculate $R_{in}(\mathbf{W}_{old}, \beta_{old})$

M-Step:

- Calculate $\mathbf{W}_{new}$ with R_{in}
- Calculate β_{new} with R_{in}

2.2.1 Data Visualisation using GTM

The data visualisation can be achieved using Bayes theorem to invert the transformation from latent space to data space. Following the choice of the prior distribution given by (6) one gains again a posterior distribution as a sum of

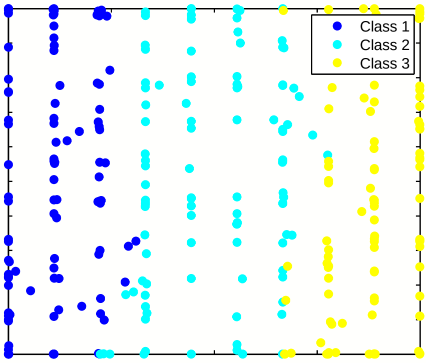


Figure 4: The projection of the 3D data onto the GTM manifold.

delta functions with coefficients given by the responsibilities R_{in}. These can be used to create a (posterior) responsibility map for single data points in the two-dimensional latent space. Since looking at the distribution of each data point is impossible and unreasonable for large data sets it is often convenient to summarise the posterior distribution by the mean given by

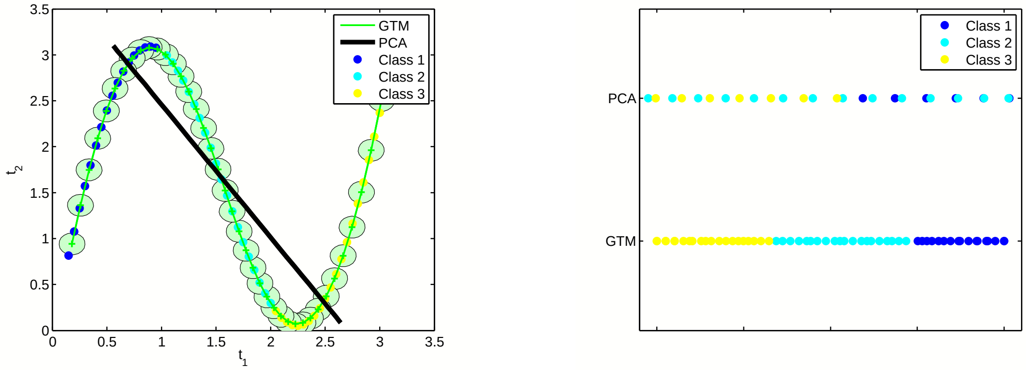
$$\begin{aligned} \langle \mathbf{x}|\mathbf{y}_n, \mathbf{W}^*, \beta^* \rangle &= \int p(\mathbf{x}|\mathbf{y}_n, W^*, \beta^*)\mathbf{x}dx \qquad (17) \\ &= \sum_{i=1}^{K} R_{in}\mathbf{x}_i \ . \qquad (18) \end{aligned}$$

However this can be misleading if one deals with a posterior distribution which is multi-modal. Therefore one should also asses the mode of the distribution

$$i^{max} = \arg\max_i R_{in} \ . \qquad (19)$$

2.3 Comparing PCA and GTM on 2D Toy Data

To compare PCA and GTM we created a 2D toy data set which is given by the function $t_2 = t_1 + 2.25\sin(2t_1)$ in the interval $[0,3]$. Then we cut the data into 3 classes according to the intervals from $[0,1]$,$[1,2]$ and $[2,3]$. In Figure 5(a) one can see how first principal components in PCA tries to align itself to the direction of the biggest variance, though since it is not able to deal with the non-linearity it aligns itself orthogonal to the direction in which the actual classes are distributed. GTM, however, is able to adapt to the non-linear structure in the simple synthetic data set, and thus capture the class boundaries between the three classes. PCA, due to the linear nature of the mapping, is not able to separate the classes in this case as we can see on the projection in Figure



(a) The function $t_2 = t_1 + 2.25\sin(2t_1)$ is divided into three classes, depending on the value of t_1.

(b) The projection of t_1, t_2 using PCA and GTM, with class labels

Figure 5: A simple example to contrast PCA and GTM when applied to non-linearly structured data.

5(b), even though the problem is a very simple one. If PCA based visualisation was the only method employed in this case an incorrect judgement about the separability of the classes might have been made.

3 Missing Data

Missing Data represent a general problem in many scientific fields [5] but are critical in environments where one has only small data sets with highly valuable samples. Usually the missing data can not be ignored since most analysis tools can not cope with them and a good imputation of these will give better and more reliable results.

In general we assume that the data set $\mathbf{T} = \mathbf{t}_1, ..., \mathbf{t}_N$ can be divided into an observed component $\mathbf{X}^o$ and a missing component $\mathbf{T}^m$. Also every point $\mathbf{t}_n = [\mathbf{t}_n^o, \mathbf{t}_n^m]$ can be split into an observed and an missing component. Assuming a missing indicator Matrix $\mathbf{M} = (M_{ij})$ the missing-data mechanism can be characterised by the conditional distribution of $\mathbf{M}$ given $\mathbf{T}$ with $p(\mathbf{M}|\mathbf{T}, \theta)$, with θ being an unknown parameter. Given this, three types of missing data can be distinguished:

- Missing completely at random (MCAR)

$$p(\mathbf{M}|\mathbf{T}, \theta) = p(\mathbf{M}|\theta) \tag{20}$$

If the missing data depend only on the unknown parameter θ.

- Missing at random (MAR)

$$p(\mathbf{M}|\mathbf{T}, \theta) = p(\mathbf{M}|\mathbf{T}^o, \theta) \tag{21}$$

  If the missing data only depend on other observed data.

- Not missing at random (NMAR)

$$p(\mathbf{M}|\mathbf{T}, \theta) = p(\mathbf{M}|\mathbf{T}, \theta) \tag{22}$$

  If the missing data depend on observed and other missing data.

Generally most existing imputation methods based on moments and estimation equations give unbiased results only on MCAR data, while likelihood methods can also deal with MAR data. There is no unbiased approach to deal with NMAR data, unless the missing data mechanism is completely known.

There are different standard methods to deal with missing data:

- The simplest approach is called complete-case analysis and confines attention to only those cases where all D variables are available. The advantage is that one can use all the standard statistical analyses without modification. The disadvantage is that one wastes a lot of information and in addition, if the MCAR assumption does not hold, a bias will be introduced to the imputation.

- Another approach is the Available-case [9] analysis where every variable is treated differently and one uses all the information for each variable to estimate the parameters of the data distribution for example the mean and variance. The advantage here is that one uses information from the incomplete cases but the disadvantage is that there are now different sample sizes for each variable. This makes analysis with more sophisticated methods quite complex and in addition it also has problems with bias and comparability across variables if the MCAR assumption does not hold.

In the following discussion a range of different methods for the imputation of missing values will presented where the advantages and draw backs of each will be discussed. In the subsequent chapter we present a benchmark to compare the different methods with each other.

3.1 Mean Imputation

In this very simple approach the missing values are replaced by the mean of the known values.

$$\mathbf{t}_n(m) = \frac{1}{N^o} \sum_{j=1}^{N^o} \mathbf{t}_j^o(m) \tag{23}$$

with N^o being the number of observed components for the variable ($=$ *dimension*) in question and m being the index for the missing dimension in the point t_n.
This method suffers from a lot of drawbacks which can be illustrated with the following MCAR example:
Lets assume we have 5 patients where we measure the height x in cm and the weight y in kg if the patient is under 190 cm. This results in an incomplete data set $[\mathbf{x}, \mathbf{y}]$
$\mathbf{x} = [55, 60, 63, m]$, where $m = 90kg$
$\mathbf{y} = [170, 173, 172, 193]$
Using mean imputation we would impute 59.3 and

- Introduce a serious bias to the produced results since MCAR assumption does not hold true: $E(\mathbf{x}) = 59.3$ *while the true value is 67*.

- Estimation of covariance matrix is biased *since we are reducing the correlation between the variables*.

- Estimation of variance is to small *since we are reducing the values with deviate from the mean*.

- Also the approach is clearly inappropriate for categorical variables.

3.2 Weighted Mean Imputation

Inspired through the KNN-based imputation in Bioinformatics [11] we developed another imputation method where we select points which are similar, based on a Euclidean norm, to the point of interest with the missing dimension(s) and use these to impute the missing value. The algorithm itself is therefore relatively simple:

1. Do a mean imputation to create the data set **C**

2. Compute the Euclidean distance between all the data points in **C**

3. Impute the missing components of $\mathbf{t}_n$ by calculating the average over all the data points in T which observed these components with the inverse distance to $\mathbf{t}_n$ as weight.

The algorithm exploits the local structure of the data space and does well in densely sampled areas of the data space while only doing as well as normal mean imputation in sparse populated areas of the data space, as might be expected.

3.3 Sequential Multiple Regression Imputation

Multiple Regression in general is used to approximate the linear relation between multiple variables in a data set $\mathbf{T}$. Hence assume than the values of one variable can be obtained through a linear combination of the others:

$$\mathbf{t}_i \approx a_0^i + a_1^i \mathbf{t}_1 + .. + a_{i-1}^i \mathbf{t}_{i-1} + a_i^i \mathbf{t}_{i+1} + .. + a_{d-1}^i \mathbf{t}_d \ . \tag{24}$$

Sequential Multiple Regression Imputation (SRI) [7] was introduced for the handling missing data in surveys. We tested a simplified version using only a linear multiple regression since the data we focus on are usually continuous rather than discrete. The algorithm is:

Part 1:

1. Order the variables $\mathbf{t}_{1:d}$ using the number of missing values $\hat{\mathbf{t}}_{1:d}$
2. Impute the missing values in $\hat{\mathbf{t}_1}$ with mean imputation, if there are any
3. Iteratively estimate the regression factors between the complete variables and the next incomplete variable with the least amount of missing values and use them to fill the missing values until
4. Iterate to Part2 until all values filled

Part 2:

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

This algorithm exploits the linear structures in the data but in general is vulnerable to outliers or data sets with very small variations. Furthermore the initialisation in Part 1 relatively important since it presumes a linear relationship in all the variables.

3.4 Multiple Regression Imputation with Mean initialisation and Correlation Cut

The SRI algorithm was designed for data sets where only a minority of the columns have missing data. This assumption may be true for surveys, where people do not wish to answer certain questions, but in general sciences we experience missing data in almost all the columns. This leads to problems on some of the *real data* sets we have used, though we will not explore this in this technical report.

However, to have a more stable linear imputation using the SRI we created a tuned multiple regression imputation. This method differers from the SRI because it is initialised with a mean imputation to permit the use of the complete data matrix and make it more robust against outliers. We also use the correlation coefficient between the dimensions as a quality cut to further improve stability in the regression models. The correlation coefficient tells us about the linear relation ship between two dimensions and if this is not sufficiently high it makes no sense to use this dimension in the calculation for the regression.

In our case we want to use this to estimate missing values but since we assume that all variables have missing values and that we may have data sets with a lot of missing values we will combine it with the mean imputation algorithm to obtain enough data to learn the coefficients $\mathbf{a}^j = [a^j_0, ..., a^j_{d-1}]$ directly at the beginning. First we construct a complete data set $\mathbf{C}$ using the mean imputation on the incomplete data set $\mathbf{T}$. Then we learn the regression factors $\mathbf{a}^j$ on this data set and after this we will use these regression factors to re-estimate the missing values in T while using them on the complete data matrix $\mathbf{C}$. Therefore the algorithm is:

0. Perform a mean imputation to create a complete data set, keeping track of the missing value locations.

1. Compute the correlation coefficient between the variables on the complete data set.

2. Estimate the regression factors $\mathbf{a}^j = [a^j_0, ..., a^j_{d-1}]$ on the dimensions where the correlation coefficient is sufficiently high for stability.

3. Use the regression model to recompute the missing values and create a new complete data set.

4. Check that none of the imputed values is outside the range of the known values and thus unreasonable *(this sanity check is required because even with only including well correlated variables, linear multiple regression can become unstable when large numbers of values are missing).*

5. Assess whether the algorithm has converged; if not go to step 1.

This algorithm exploits the linear structures in the data while still being able to cope with a large amount of missing data though we still expect singularities if the amount of missing data gets too high.

3.5 Extension of GTM for Missing Data

The EM algorithm with missing data can be extended to the GTM model [12] to deal with missing data. The error function given by the negative log-likelihood from (5) can be written as

$$E_{comp} = -\sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn} \ln p(\mathbf{t}_n|\theta_k)\ . \tag{25}$$

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

$$\begin{aligned} E_{comp} = &\quad -\textstyle\sum_{n=1}^{N}\sum_{k=1}^{K} z_{kn}[\frac{1}{2}\ln|\mathbf{\Sigma}_k| + \frac{D}{2}\ln 2\pi & (26)\\ + &\qquad \tfrac{1}{2}(\mathbf{t}_n^o - \boldsymbol{\mu}_k^o)^T \Sigma_k^{-1,oo}(\mathbf{t}_n^o - \boldsymbol{\mu}_k^o) & (27)\\ + &\quad \tfrac{1}{2}(\mathbf{t}_n^m - \boldsymbol{\mu}_k^m)^T \Sigma_k^{-1,mm}(\mathbf{t}_n^m - \boldsymbol{\mu}_k^m)] & (28)\end{aligned}$$

where $\boldsymbol{\mu}_k$ and $\mathbf{\Sigma}_k$ denote the means and covariance of the kth Gaussian respectively. The superscripts, for example, $(-1, oo)$ denotes inverse followed by submatrix operations where $\mathbf{\Sigma}_k$ is divided into $\begin{pmatrix}\mathbf{\Sigma}_k^{oo} & 0\\ 0 & \mathbf{\Sigma}_k^{mm}\end{pmatrix}$ corresponding to $\boldsymbol{t} = \begin{pmatrix}\boldsymbol{t}^o\\ \boldsymbol{t}^m\end{pmatrix}$. The expected value is now taken with respect to both sets of missing variables. After taking the expectation one ends up with 2 unknown terms $z_{kn}\boldsymbol{t}_n^m$ and $z_{kn}\boldsymbol{t}_n^m\boldsymbol{t}_n^{mT}$, so one must calculate the expectation for these terms. To compute these expectations, variables $\hat{\mathbf{t}}_{kn}^m$ are introduced,

$$\hat{\mathbf{t}}_{kn}^m \equiv \langle \boldsymbol{t}_n^m | z_{kn} = 1, \boldsymbol{t}_n^o, \theta_k \rangle = (\boldsymbol{y}_k^m)^{old} \tag{29}$$

which are the linear least-squares regression between $\boldsymbol{t}_n^m$ and $\boldsymbol{t}_n^o$ predicted by the kth Gaussian, where the superscript 'old' denotes the result from the last M-step: $(\boldsymbol{y}_k^m)^{old} = (\boldsymbol{W}_{old}\Phi(\boldsymbol{x}_k))^m$.

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

$$R_{kn} = \frac{\frac{\beta}{2\pi}^{D/2}\exp\{-\frac{\beta}{2}||\boldsymbol{y}(x_k; W) - \boldsymbol{t}_n||^2\}}{\sum_j^k \frac{\beta}{2\pi}^{D/2}\exp\{-\frac{\beta}{2}||\boldsymbol{y}(x_j; W) - \boldsymbol{t}_n||^2\}} \tag{30}$$

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

- M-step: The weights are updated to $\boldsymbol{W}_{new}$ as in equation (14) for complete training data:

$$\Phi^T \boldsymbol{G}_{old}\mathbf{\Phi}\boldsymbol{W}_{old}^T = \mathbf{\Phi}^T\boldsymbol{R}_{old}\boldsymbol{T} \tag{31}$$

where the missing data are filled in with the posterior means

$$\langle \boldsymbol{t}_n^m | \boldsymbol{t}_m^o, \theta_{old} \rangle = \sum_{k=1}^{K} R_{kn} \hat{\boldsymbol{t}}_{kn}^m \; . \tag{32}$$

Then the inverse variance is updated as follows

$$\frac{1}{beta} = \frac{1}{ND} \sum_{n=1}^{N} \sum_{i=1}^{N} R_{in} (||\boldsymbol{t}_n^o - \boldsymbol{y}_k^o||^2 + \langle z_{kn} || \boldsymbol{t}_n^m - \boldsymbol{y}_n^m ||^2 \rangle) \tag{33}$$

where

$$\langle z_{kn} || \boldsymbol{t}_n^m - \boldsymbol{y}_n^m ||^2 \rangle = n_m (\beta^{-1})^{old} + (\hat{\boldsymbol{t}}_{kn}^m)^T \hat{\boldsymbol{t}}_{kn}^m - 2 (\hat{\boldsymbol{t}}_{kn}^m)^T \boldsymbol{y}_k^m + (\boldsymbol{y}_k^m)^T \boldsymbol{y}_k^m \tag{34}$$

and n_m is the number of missing values in data point $\boldsymbol{t}_n$. A more detailed derivation can be found in [12].

3.6 Performance Indicators

To compare the different imputation methods a measure of performance is needed. The following error measures account for the difference in the original $\mathbf{t}_i$ values and the imputed $\mathbf{y}_i$ values and give an idea about how well the imputation has performed.

Error (bias)

$$ER = \sum_{i=1}^{N} \mathbf{t}_i - \mathbf{y}_i$$

This measure is generally referred to as the bias of the method and should be zero.

Root Mean Square Error

$$RMSE = (\frac{1}{N} \sum_{i=1}^{N} [\mathbf{t}_i - \mathbf{y}_i]^2)^{\frac{1}{2}}$$

The RMSE is an approximation to the standard deviation of the residuals (errors) from the predictions. It can be sensitive to outliers since it is a second moment statistic.

4 Benchmarking the Imputation Methods

To compare the different imputation methods we created a high dimensional data set which is generated from a 2 dimensional data set. The 2 dimensional

Parameters	Value
Base domain of data set	[-4,4]
Number of Samples in the data set	100
Dimension of base data set	2
Projection Function	RBF
Number of RBF functions	9
Set of the weights	Random
Dimension of the projected data	15

Table 1: Summary of the toy data

data set is composed of 4 distinct classes. We looked at the performance of the imputations methods on $p_i = [0.1, .., 0.6]$ proportion of missing values where we generated 100 random missing data patterns for each p_i to average the results of the performance indicators and get a representative value with respect to the missing data pattern. To get an idea about the overall usability of all the algorithms and to compare PCA against GTM we looked at samples of the visualisation on these imputed data sets to determine whether they still allowed a visual distinction between the 4 classes.

4.1 Toy Data

To simulate a high dimensional data set which has non-linear relations between the variables we created a toy data set. We used a two-dimensional base data set and transformed it into a 15 dimensional data set via a *Gaussian* RBF mapping function. The summery of the procedure can be seen in Table 1. First we created 100 random data points in the domain [-4,4] in 2 dimensions consisting of 4 partially different classes with 25 points in each class. Then we used a RBF network with 9 hidden layers to project these data onto a 15 dimensional data space. The RBF network was initialized with random weights thus no prior information or restrictions were enforced on the mapping of the data.

4.2 Imputation Results

The results on the performance indicators show a similar trend when averaged over all the dimensions. The most widely employed missing data imputation method of using MI is always the worst alternative and should not be used. The WMI always performs better then the MI and is more stable at very high rates of missing data (> 0.6). WMI has low computational cost and the simple implementation may make it interesting for some applications. Also WMI would be expected to perform better on larger data sets. The MRI algorithm is slightly better then WMI algorithm but never gets close to the performance of the SRI or GTMI on the test data but nevertheless its stability might prove useful for future benchmarks.

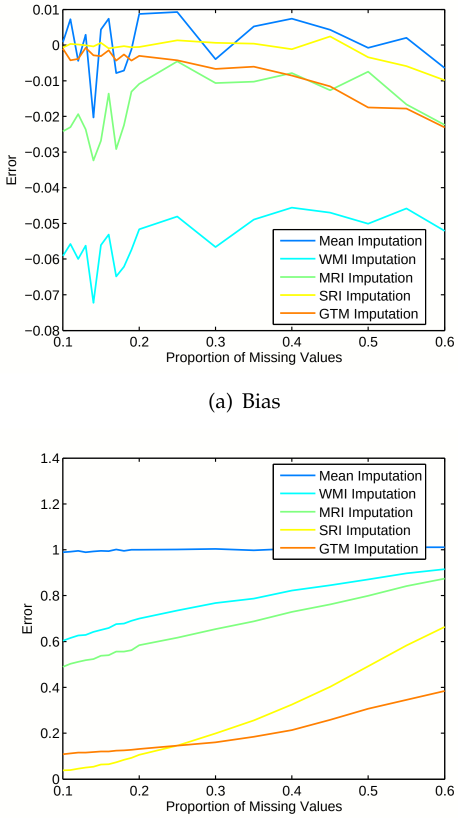
GTMI performs slightly worse then SRI on missing data patterns with a low proportion of missing values (< 0.2) but proves to be more stable once there are higher proportions of values missing as shown in Figure 4.2.

To further analyse the results of the behavior of the imputation methods we looked at the scatter plot of true versus estimated value for the missing test points and the RMSE on the first dimension, the results being shown in Figure 4.2. As expected the RMSE shows the same patterns as the average over all the dimensions and the scatter plot reveals how the performance of the different imputation methods varies. The SRI and GTM fit closely to the line of the original values, while the MRI is a little more scattered around it. The WMI is superior to the MI algorithm in the densely populated areas but is only able to infer structure data locally and can not grasp general trends.

4.3 Projection Results

Measuring imputation algorithms with performance indicators will inform one about the proportional difference in the rating of the algorithms but it does not provide a lot of evidence about their use in data exploration. For the applied scientist it is important to know when an imputation algorithm might still help to reveal the hidden structure of the data and when it is unlikely to make much of a difference.

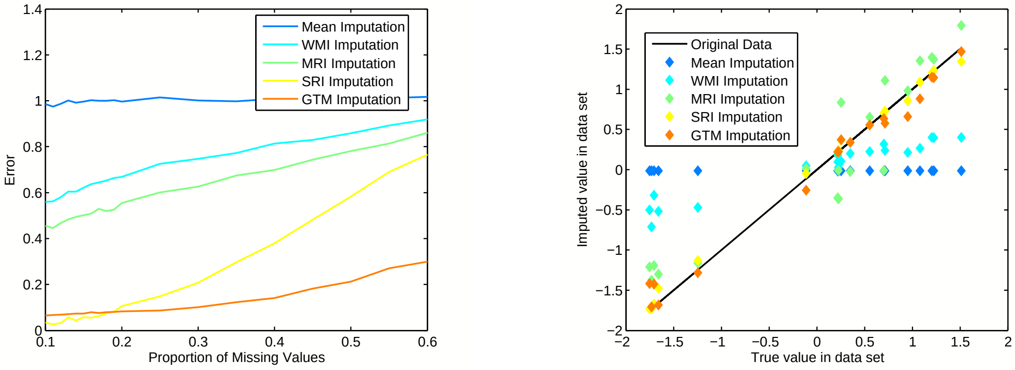
To find an indicator about the usability and to compare the projection of PCA and GTM we plotted the PCA projection on missing data patterns with a p_i of $[0.15, 0.3, 0.45, 0.6]$ of missing values, while imputing missing data with MI. The results in Figure 4.3 indicate that the it is already hard to distinguish different structures with only a proportion of 0.15 of missing values and it becomes impossible if this rises to 0.45. Then on the same missing data patterns we performed a GTMI before using the PCA and there it is possible to distinguish all the 4 different classes to up to a proportion of 0.6 missing values when using PCA, 4.3. For an applied scientist this are good news since they



(a) Bias

(b) Root Mean Square Error

Figure 6: The RMSE on the toy data reveals that GTMI is far more stable at high rates of missing data, while being only slightly worse a low rates of missing data. Further all the imputation methods produce an essentially unbiased result on the simple test data.



(a) Root Mean Square Error

(b) Scatter Plot

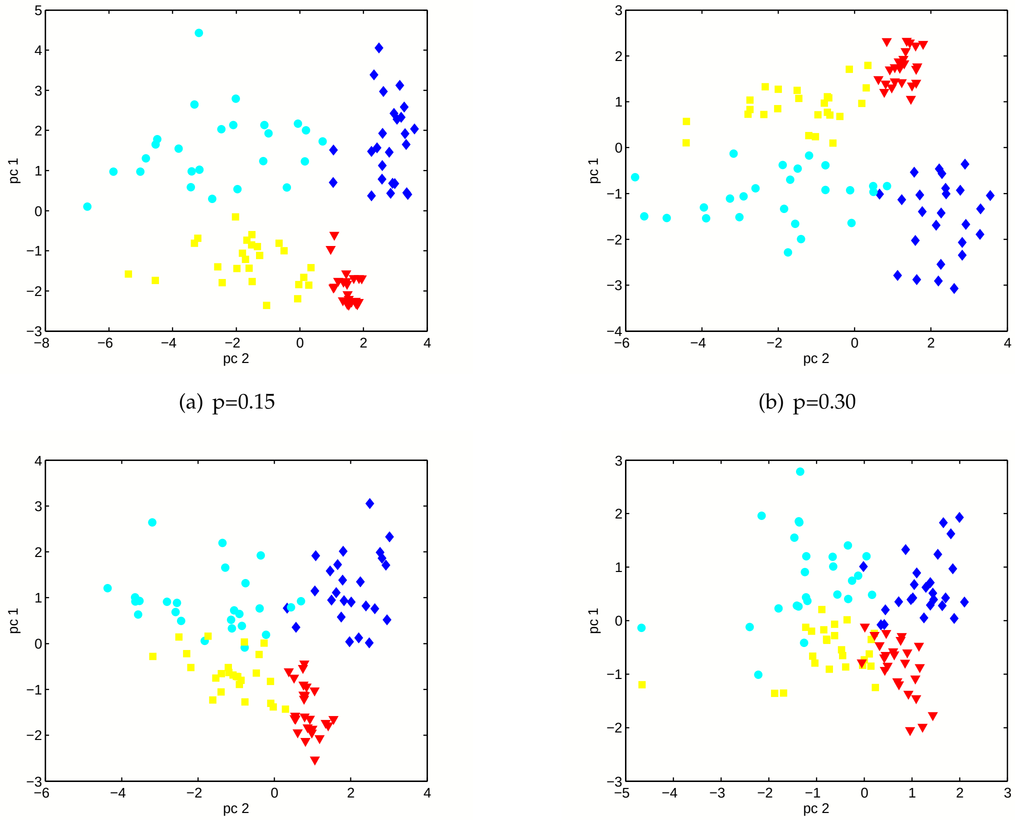
Figure 7: The error and the scatter plot of the first dimension on a missing data pattern with $p_i = 0.2$ reveal how the different imputation methods fit the missing data.

can use very sparse data sets as long as there is a strong structure in the data set. These results become even more evident when looking at the projections in Figure 4.3 where we show the results of the GTM projection learnt in the GTMI, which makes it possible to easily distinguish between all the classes, except some small outliers at $p_i = 0.6$.

5 Summary

The GTM model maps complex, high dimensional data onto a low dimension, typically 2D, representation and enables exploration of both the linear and non-linear relations between different samples and variables in a data set, on a single plot. This helps us to understand the data and draw conclusions from it, in conjunction with other visual and analytic methods.

GTM successfully copes with missing values in the data, due to its probabilistic formulation and can therefore make use of *all the information* that is available from the given data set. We have shown that GTM remains relatively robust even when a high proportion of data is missing. This suggests GTM might also have a very useful role in the replacement of missing values in a data set, particularly where the proportion of missing values is large.



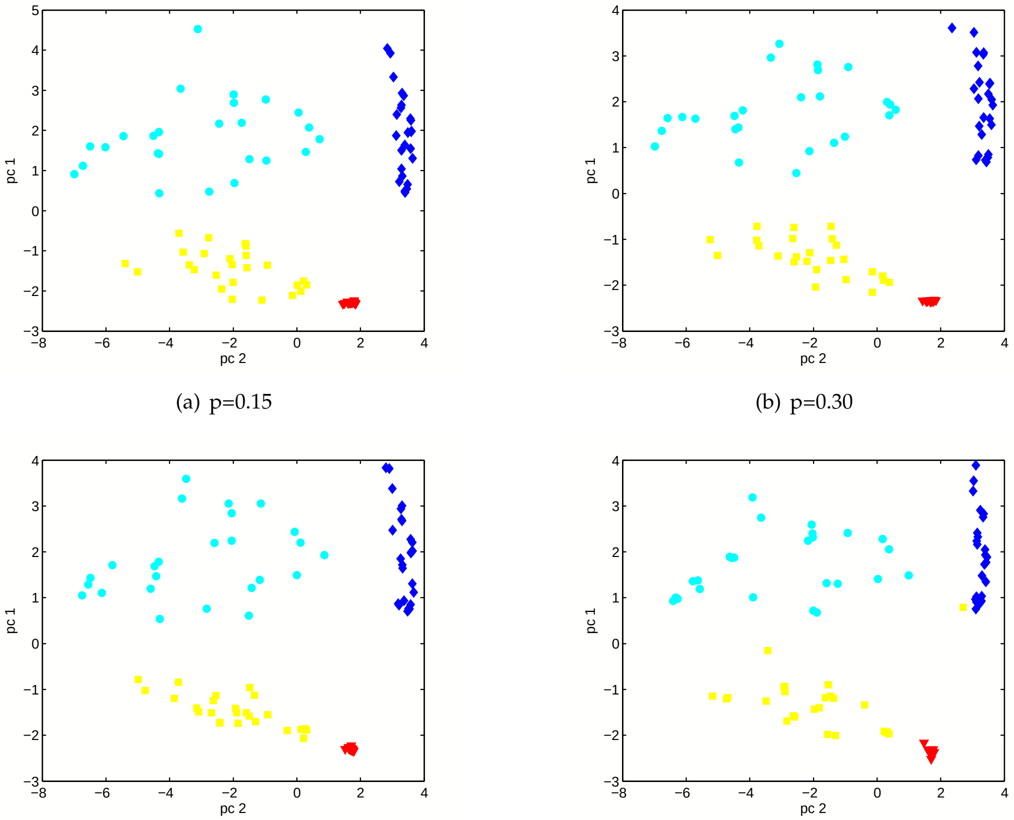
(a) p=0.15

(b) p=0.30

(c) p=0.45

(d) p=0.60

Figure 8: PCA on toy data with Mean Imputation, where the p value denotes the proportion of missing data. The class boundaries become indistinct as soon as the proportion of missing data raises.



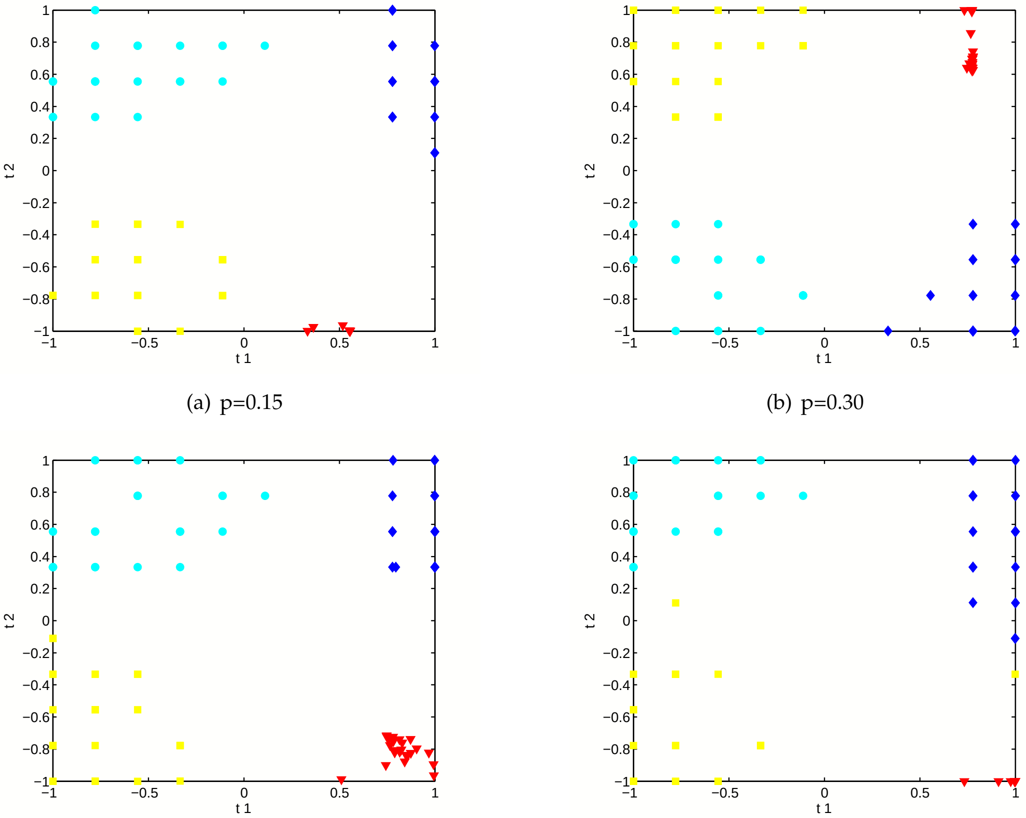
(a) p=0.15

(b) p=0.30

(c) p=0.45

(d) p=0.60

Figure 9: PCA on toy data with GTMI, where the p value denotes the proportion of missing data. The class boundaries are easy to distinguish even with a high amount of missing data.



(a) p=0.15

(b) p=0.30

(c) p=0.45

(d) p=0.60

Figure 10: The direct GTM mapping on the toy data. The class boundaries are easy to distinguish even with a high amount of missing data.

6 Future Work

Future work in this area will aim to integrate the knowledge of experts within GTM via the structure of the covariance matrix which is very simple in the present work. It is also interesting to assess whether the definition of metrics or measures to assign a value to the quality of the visualisation can be implemented successfully in a commercial setting where automation and simplicity are crucial to successful deployment. It is planned to further extend GTM to deal with missing data using a 'multiple imputation' approach to be able to assign a value for the uncertainty (variance) of the missing data estimates.

References

[1] C. M. Bishop, M. Svensen, and C. K. I. Williams. Developments of the generative topographic mapping. *Neuro Computation*, 10:215–234, 1997.

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

[3] C. Chatfield and A.J. Collins. *Introduction to Multivariate Analysis*. Chapman and Hall, 1980.

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

[5] G.Latini and G. Passerini. *Handling Missing Data*. WIT Press, 2004.

[6] I.T. Jollieffe. *Principal Component Analysis*. Springer Verlag, 1986.

[7] John Van Hoewyk JTrivellore E. Raghunathan, James M. Lepkowski and Peter Solenberger. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology*, 27:85–95, 2001.

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

[9] Roderick J A Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

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

[11] Gavin Sherlock Pat Brown Trevor Hastie Robert Tibshirani David Botstein Olga Troyanskaya, Michael Cantor and Russ B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, Vol. 17:Pp. 520–525, 2001.

[12] Yi Sun. *Non-linear Hierarchical Visualisation*. PhD thesis, Aston University, 2002.