Aston University
Birmingham B4 7ET
http://webct.aston.ac.uk/

Block GTM: Incorporating Prior Knowledge of Covariance Structure in Data Visualisation

Martin Schroeder[a], Ian T. Nabney and Dan Cornford

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

Technical Report: NCRG/2008/006 September 25, 2008

[a]Corresponding author. Phone: +44 (0) 7908123546 // Email: shroderm@aston.ac.uk

Abstract

Visualising data for exploratory analysis is a big challenge in scientific and engineering domains where there is a need to gain insight into the structure and distribution of the data. Typically, visualisation methods like principal component analysis and multi-dimensional scaling are used, but it is difficult to incorporate prior knowledge about structure of the data into the analysis.
In this technical report we discuss a complementary approach based on an extension of a well known non-linear probabilistic model, the Generative Topographic Mapping. We show that by including prior information of the covariance structure into the model, we are able to improve both the data visualisation and the model fit.

Contents

1 Introduction

Data visualisation is widely recognised as a key task in exploring and understanding data sets. Including prior knowledge from experts into probabilistic models for data exploration is important since it constrains models, which usually leads to more interpretable results. As measurement becomes cheaper, datasets are becoming steadily higher dimensional. This is not true for the number of samples which might stay quite low since it is quite costly for example to include more patients in clinical trials or do more drills in the oil exploration industry. These high dimensional data sets poses a great challenge when using probabilistic models since the training time and the accuracy of these models depends on the number of free parameters. A common fix for gaussian models is to reduce the number of parameters and to ensure sparsity in the model is to constrain the covariance matrix to be either diagonal, or spherical in the most restricted case. These constraints exclude valuable information about the data structure, especially in cases where there is some understanding of the structure of the covariance matrix.

A good example can be seen in chemical analysis like Gas Chromatography-Mass Spectrometry (GC-MS). When one examines the results of GC-MS run over different samples, one knows that certain compounds are highly correlated with each other. Incoperating this information into the model will result in a block matrix covariance structure. Theoretically this will help to reduce the number of free parameters without loosing too much valuable information. In this technical report we will look at a common probabilistic model for data exploration and imputation called the Generative Topographic Mapping (GTM) [1]. The standard GTM uses a spherical covariance matrix and we will modify this algorithm to work with a informative block covariance matrix.

The technical report has the following structure. In Section 2 we briefly review common data exploration algorithms and describe in more detail problems with assessing unsupervised learning tasks like data visualisation. In Section 3 we introduce the standard GTM model, extend it to the case of a block covariance matrix and describe how GTM can deal with missing data. In Section 4 we present some experiments on artificial data, where we compare the normal GTM against the block version and show where the advantages of both models lie. Finally in Sections 5 and 6 we conclude the report and point out further areas of research.

2 Data Exploration

2.1 Data Visualisation

A fundamental requirement for visualisation of high-dimensional data is to be able to map, or project, the high-dimensional data onto a low-dimensional representation while preserving as much information about original the structure in the high-dimensional space as possible. This low-dimensional representation is usually 2D so that the projected data can be shown 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 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 [4] linearly transform the data space and project the data onto the lower-dimensional space while retaining the maximum information [1]. Other methods like the Kohonen, or Self Organising, Maps [9] and the Generative Topographic Mapping (GTM) [1] try to capture the topology[2] of the data. Another recent topology-preserving method, the Gaussian Process Latent Variable Model (GP-LVM) [10] utilises a Gaussian Process prior over the mapping function and obtains the latent space projection by directly optimising the latent point. Geometry-preserving methods like multi-dimensional scaling [2] and Neuroscale [11] try to find a representation in latent space which preserves the geometric distances between the data points. The later approach can even be extended through a technique called Locally Linear Embedding [12, 7], which defines another metric to calculate the geometric distances, one uses to optimise the mapping function. In this technical report we will focus on the classical Generative Topographic Mapping (GTM) and an extension which we will call Block Generative Topographic Mapping (B-GTM).

2.2 Assessing Unsupervised Learning

The dimensionality reduction methods discussed in this report are all examples of unsupervised learning. Thus we cannot tell *a priori* what is the expected or desired target. This makes it very difficult to judge which method is the best in the sense of telling us the most about a certain dataset. In the simple case of artificial data one can use prior knowledge about the structure of the data in the original space to quantify the error on the projection. For the more complex case of real data there are various approaches to this problem ranging from different resampling methods [13] to a Bayesian approach using the GP-LVM [6].

In this technical report we are going to focus mainly on the following measures of the quality of a projection:

Nearest Neighbour Label Error: The nearest neighbour label error can only be computed on labelled data, where we know the class of each data point. The idea is to consider the projected data and calculate for each point how many of the k nearest points are in the same class. Then we average the fraction of k-nearest neighbours in the same class over all the points. Finally we average over all the classes as well.

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

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

Model Likelihood: The model likelihood is the likelihood that the actual data in the observation space were created given the actual model. This quantity can be calculated for every probabilistic or generative projection model as long as it induces a density in the data space.

GP-LVM model selection likelihood: The GP-LVM likelihood as described in [6] is calculated in two steps. We first take an GP-LVM model and fix the latent space to the projection we want to evaluate. We then optimise the hyperparameters for the GP-LVM and calculate the data likelihood afterwards. The idea is that the GP-LVM, since it is based on a Gaussian Process, should be flexible enough to model most possible mapping functions and thus give a comparable estimate of how well the actual latent space is representing the given data space.

3 Block GTM

3.1 Standard GTM

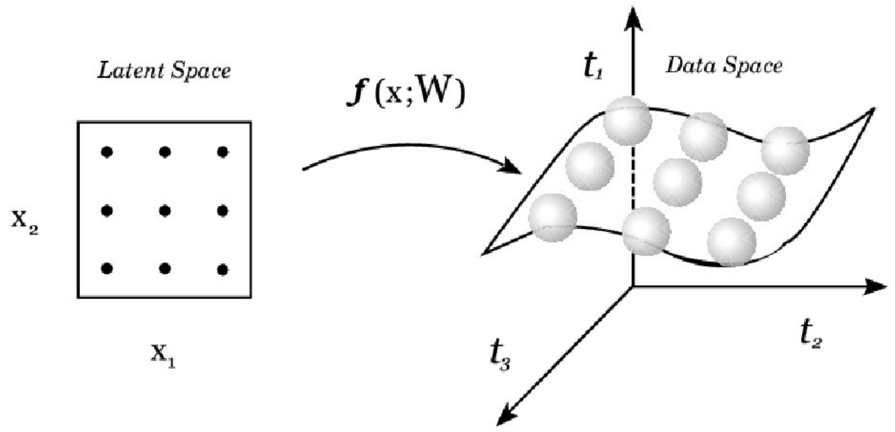


Figure 1: The non-linear function $\Theta(\mathbf{x}, \mathbf{W})$ defines a manifold S embedded in the data space given by the image of the latent variable space under the mapping $\mathbf{x} \to \mathbf{t}$.

The essence of GTM is to try and fit a *density model*, which is constrained to lie on 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, which is 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 of Gaussian centres in the data space. Thus the GTM may be described as a mixture of Gaussians constrained to lie on a manifold. 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) algorithm [5] to maximise the data likelihood.

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 $\mathbf{t} = \Theta(\mathbf{x}, \mathbf{W})$ 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 $\mathbf{t}$, shown for the case $L = 2$ and $D = 3$ in Figure 3.1.

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

$$p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\{-\frac{\beta}{2} \parallel \Theta(\mathbf{x}, \mathbf{W}) - \mathbf{t} \parallel^2\} , \tag{1}$$

where one should note that it is possible to use other distributions for $p(\mathbf{t}|\mathbf{x})$, such as a Bernoulli for binary variables . For a given value of $\mathbf{W}$, the distribution is obtained by integration over the distribution $p(\mathbf{x})$

$$p(\mathbf{t}|\mathbf{W}, \beta) = \int p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) p(\mathbf{t}) \, dx . \tag{2}$$

For a given data set $\mathbf{T} = (\mathbf{t}_1, ..., \mathbf{t}_N)$ of N data points, the parameter matrix $\mathbf{W}$ and the inverse variance β are obtained through by optimising the data set log likelihood

$$L(\mathbf{W}, \beta) = \ln \prod_{n=1}^{N} p(\mathbf{t}_n|\mathbf{W}, \beta) . \tag{3}$$

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 (2) 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 centred on the nodes of a regular grid in latent space

$$p(\mathbf{x}) = \frac{1}{K} \sum_{i=1}^{K} \delta(\mathbf{x} - \mathbf{x}_i) \tag{4}$$

in which case the integral in (1) can be evaluated analytically and reduces to a sum of distributions. 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 (2) and (4) the distribution function in the data space takes the form

$$p(\mathbf{t}|\mathbf{W}, \beta) = \frac{1}{K} \sum_{i=1}^{K} p(\mathbf{t}|\mathbf{x}_i, \mathbf{W}, \beta) \tag{5}$$

and the corresponding log likelihood becomes

$$L(\mathbf{W}, \beta) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{K} \sum_{i=1}^{K} p(\mathbf{t}_n | \mathbf{x}_i, \mathbf{W}, \beta) \right\} . \tag{6}$$

Since the model consists of a mixture of distributions it is possible to find the optimal solution via an 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

$$\Theta(\mathbf{x}, \mathbf{W}) = \mathbf{W}\Phi(\mathbf{x}) , \tag{7}$$

with the elements of $\Phi(\mathbf{x})$ consisting of M fixed radial basis functions [3] and $\mathbf{W}$ being a $D \times M$ matrix.

In the case under consideration it is assumed we have a hidden variable z_{in} which tells us which component from (4) generated data point $\mathbf{t}_n$. If the i-th component generated data point t_n then $z_{in} = 1$. Since we don't know z_{in} we are using taking the expectations which are the responsibilities of the Gaussian components in the algorithm. 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 responsibilities of each Gaussian component i for every data point $\mathbf{t}_n$ using Bayes' theorem

$$R_{in}(\mathbf{W}_{old}, \beta_{old}) = p(\mathbf{x}_i | \mathbf{t}_n, \mathbf{W}_{old}, \beta_{old}) \tag{8}$$

$$= \frac{p(\mathbf{t}_n | \mathbf{x}_i, \mathbf{W}_{old}, \beta_{old})}{\sum_{j=1}^{K} p(\mathbf{t}_n | \mathbf{x}_j, \mathbf{W}_{old}, \beta_{old})} . \tag{9}$$

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

$$\langle L_{comp}(\mathbf{W}, \beta) \rangle = \sum_{n=1}^{N} \sum_{i=1}^{K} R_{in}(\mathbf{W}_{old}, \beta_{old}) \ln\{p(\mathbf{t}_n | \mathbf{x}_i, \mathbf{W}, \beta)\} . \tag{10}$$

Maximising (10) with respect to W and using (1) and (7) one gets

$$\sum_{n=1}^{N} \sum_{i=1}^{K} R_{in}(\mathbf{W}_{old}, \beta_{old}) \{\mathbf{W}_{new}\Phi(\mathbf{x}_i) - \mathbf{t}_n\} \Phi^T(\mathbf{x}_i) . \tag{11}$$

This can be written in matrix notation

$$\mathbf{\Phi}^T \mathbf{G}_{old} \mathbf{\Phi} \mathbf{W}_{new}^T = \mathbf{\Phi}^T \mathbf{R} \mathbf{T} , \tag{12}$$

with $\mathbf{\Phi}$ being a $K \times M$ matrix with elements $\Phi_{ij} = \Phi_j(x_i)$, $\mathbf{T}$ being a $N \times D$ matrix with elements t_{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

$$G_{ii} = \sum_{n=1}^{N} R_{in}(\mathbf{W}_{old}, \beta_{old}) . \tag{13}$$

Equation (12) can be solved for $\mathbf{W}_{new}$ using standard linear algebra techniques. By a similar argument, to maximise (10) with respect to β one obtains the following formula

$$\frac{1}{\beta_{new}} = \frac{1}{ND} \sum_{n=1}^{N} \sum_{i=1}^{K} R_{in}(\mathbf{W}_{old}, \beta) || \mathbf{W}_{new}\Phi(\mathbf{x}_i) - \mathbf{t}_n ||^2 . \tag{14}$$

The EM algorithm alternates between the E-step,given by evaluating 8, and the M-Step,evaluating $\mathbf{W}_{new}$ and β_{new}, until it converges to a (local) maximum as follows:

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}.

3.2 Extension to Block GTM

A novel approach to include prior information about the correlations of variables into the model is to use a full covariance matrix in the GTM noise model and to enforce a block structure onto it. This still results in a relatively sparse covariance matrix but keeps the number of unknown parameters at an acceptable level while helping the model to fit the data including prior information about its structure. After ordering the variables by their known groups, the covariance matrix will have the following structure:

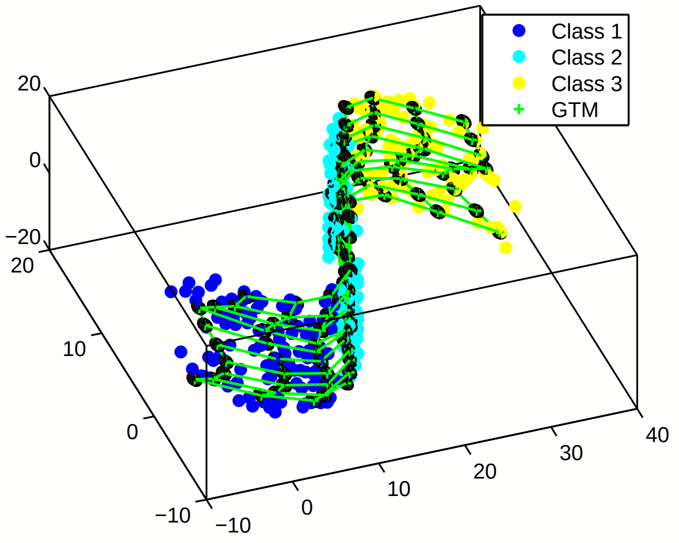
$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 & \dots & 0 \\ 0 & \Sigma_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \Sigma_p \end{bmatrix} \tag{15}$$

with Σ_1 to Σ_p being the submatrices of correlated group of variables. We further assume that there is no correlation between variables in distinct groups. The extension of the learning algorithm is straightforward since the only changes occur in the computation of R in the E-step and of Σ in the M-step. In the E-Step the computation of the posterior probabilities of each Gaussian component index by i changes since the Gaussian is no longer spherical but rather has the block covariance matrix Σ:

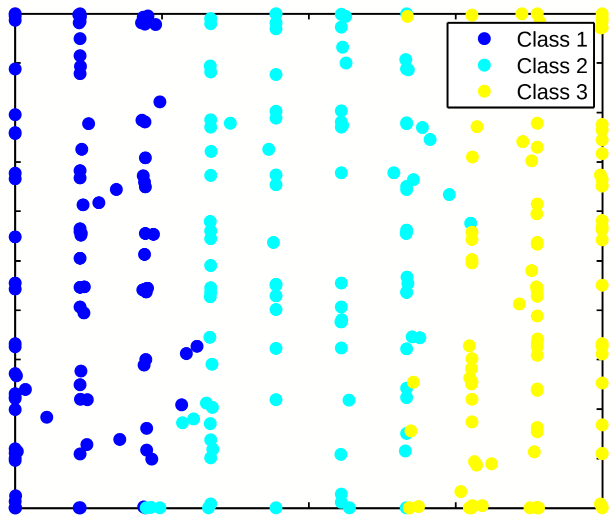
$$\begin{aligned} R_{in}(\mathbf{W}_{old}, \Sigma_{old}) &= p(\mathbf{x}_i | \mathbf{t}_n, \mathbf{W}_{old}, \Sigma_{old}) && (16) \\ &= \frac{p(\mathbf{t}_n|\mathbf{x}_i, \mathbf{W}_{old}, \Sigma_{old})}{\sum_{j=1}^{K} p(\mathbf{t}_n|\mathbf{x}_j, \mathbf{W}_{old}, \Sigma_{old})} \ . && (17) \end{aligned}$$

For the M-step we have to derive the update for the full covariance matrix Σ. Taking the derivative of the negative log likelihood with respect to Σ we get:

$$-\frac{\partial L(\mathbf{W}, \Sigma)}{\partial \Sigma} = -\sum_{n=1}^{N} \frac{D}{2} \Sigma^{-1} - \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} R_{in} \Sigma^{-1} \mathbf{a}_{kn} \mathbf{a}_{kn}^T \Sigma^{-1} \ , \tag{18}$$



(a) Fit of the GTM to the data



(b) Projection of the data

Figure 2: The fit and projection of a 15x15 GTM with 16 RBF centres to S-shaped test data after 50 iterations with the EM algorithm and initialisation with PCA.

where $\mathbf{a}_{kn} = (\Theta(\mathbf{x}_k, \mathbf{W}) - \mathbf{t}_n)$. Setting the derivative to zero we obtain

$$\Sigma = \frac{1}{ND} \sum_{n=1}^{N} \sum_{k=1}^{K} R_{in} \mathbf{a}_{kn} \mathbf{a}_{kn}^T \, , \tag{19}$$

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

4 Experiments On Artificial Data

To explore the behaviour of block GTM in comparison to spherical (i.e. standard) GTM, full GTM, as well as PCA, several experiments were carried out. The data were sampled from a GTM with a 8×8 grid in the latent space. The grid was projected into a higher dimensional space using a 2×2 RBF network. The weights were randomly sampled from a normal distribution with zero mean and standard deviation one. Since the RBF was chosen with random weights the restriction to a 2×2 RBF ensured a non-linear but smooth and realistic mapping. The GTM used to generate the data had a block diagonal covariance matrix and experiments were conducted with varying levels of variance and correlation. The overall variance of the data varied from 6.45 to 7.55, with covariances around the single Gaussians varying from 2 to 20. In each experiment 100 data points were sampled from this GTM and each experiment was conducted 20 times, with a different randomly generated GTM each time. Further the 8×8 grid was split into 4 classes with the 16 Gaussians building one corner of the grid being defined as one class. The data projection models were all fitted to the same data set and their performance was evaluated.

To check if the block GTM yields any advantage over the other methods through the knowledge of the block structure, a set of experiments were conducted on block sizes ranging from 2 to 5. To see how the block GTM performed when being misspecified a shuffle experiment on 24-dimensional data sets was conducted where a certain percentage of the correlations were “shuffled” into another block. The other block was randomly selected. Therefore one has to keep in mind that variables might end up in the same group; in the case of just two groups this is always the case. In all cases the fitted GTM (spherical, block and full) was chosen to have a 4×4 RBF and a 15×15 grid to provide sufficient flexibility to fit the data.

4.1 Results

As expected the results for the different experiments vary greatly depending on the variance one chooses for the generating GTM. This can be seen in Figure 3 where the average NNL Error for 20 experiments was plotted for different overall variances in the generated data. The data set in this case was always 20-dimensional. The variance correlates strongly with the enforcement of the block structure in the data. It also has to be said, that the average was only taken until the block GTM algorithm broke down. The block GTM algorithm breaks down, once a certain data dimensionality is reached. The reason for this has yet to be determined. The exact breakdown points can be seen in the later experiments. The experiment shows nicely that the performance of the block GTM stays relatively constant regardless of the noise level in the data, while the spherical GTM and the PCA perform worse with increasing variance. This leads to an advantage in the case where 2, 3 or 4 blocks are present, while in the case where we have 5 blocks or more it takes a certain level of variance before the block version performs better. In the following sections we first look at 3 cases, where we enforced the block structure to different levels (weakly, moderately and highly). Then we look at the performance when the block structure is misspecified.

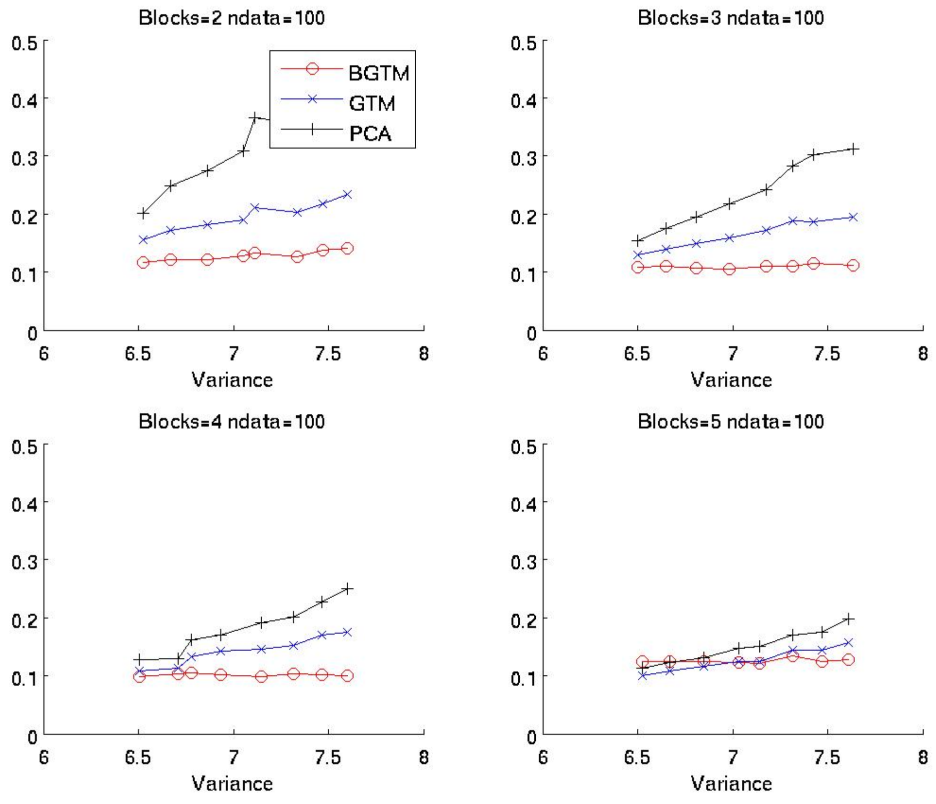


Figure 3: Average NNL Error for different levels of variance in the data.

4.2 Weakly structured data

As an example for low structured cases we have chosen the covariances to be 2 around the single Gaussians from which we sample in the grid, which results in an overall variance of 6.45 in the data set. Figures (4 to 7) show how the NNL Error varies with increasing dimensionality of the data set. The NNL Error calculated for $k = 1 : 4$ shows congruent behaviour for different choices of k. The results are similar for all block sizes running from 2 to 5.

Block GTM performs as well as or slightly better than the other methods until the data dimensionality reaches 20. Looking at the range from 20 to 38 dimensions, the performance is slightly worse and it completely breaks down when 40 dimensions are reached. The full GTM model has no advantage in this case and breaks down even earlier than the block GTM. PCA and spherical GTM are performing constantly at the same good level and are not affected by the increase in dimensionality.

Figure (8) shows the average GP-LVM likelihood of the GTM cases over increasing dimensionality in the data set. The results are harder to distinguish but congruent with the results of the NNL Error until the data dimensionality reaches 44. The trend for a breakdown of block GTM and full GTM is not as strong as in the NNL Error case but is still present. Strangely, after going beyond 44 dimensions the likelihood for the block and full GTM drops radically.

Figure (9) shows the model negative log likelihood of the different GTMs over increasing dimensionality in the data set. The spherical GTM exhibits a slow linear increase in likelihood with increasing dimensionality. The same is true for the block GTM, which also perform slightly better than spherical GTM until it starts to break down. The breakdown levels are different for the different block sizes but increase with an increasing number of blocks. The full GTM model always performs the worst and exhibits a near-exponential increase in its negative log likelihood.

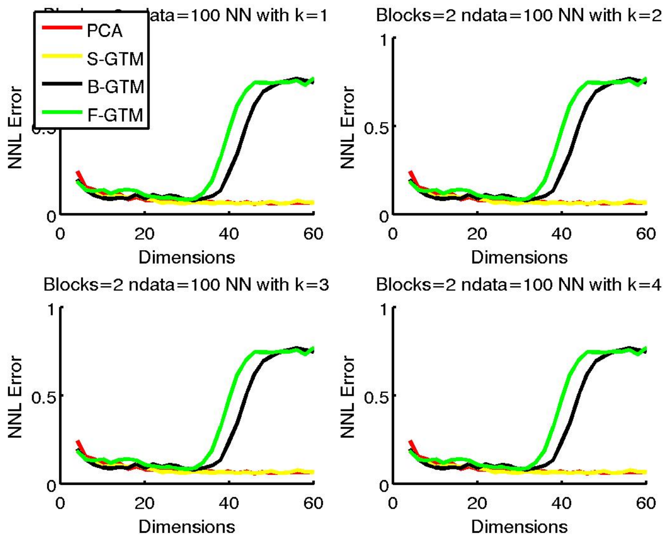


Figure 4: Comparing the different methods with 2 blocks in the covariance. The NNL Error is calculated for different values of *k*.

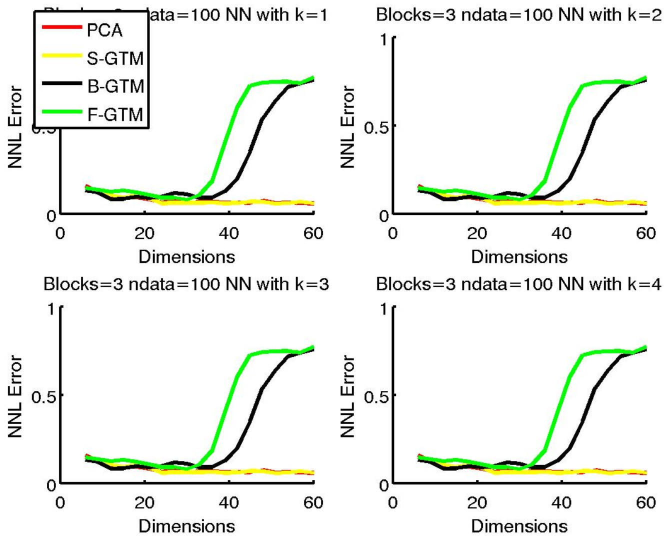


Figure 5: Comparing the different methods with 3 blocks in the covariance. The NNL Error is calculated for different values of *k*.

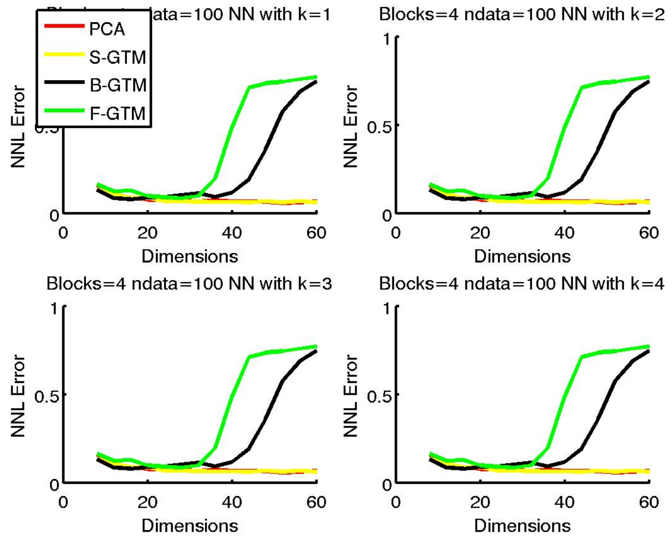


Figure 6: Comparing the different methods with 4 blocks in the covariance. The NNL Error is calculated for different values of k.

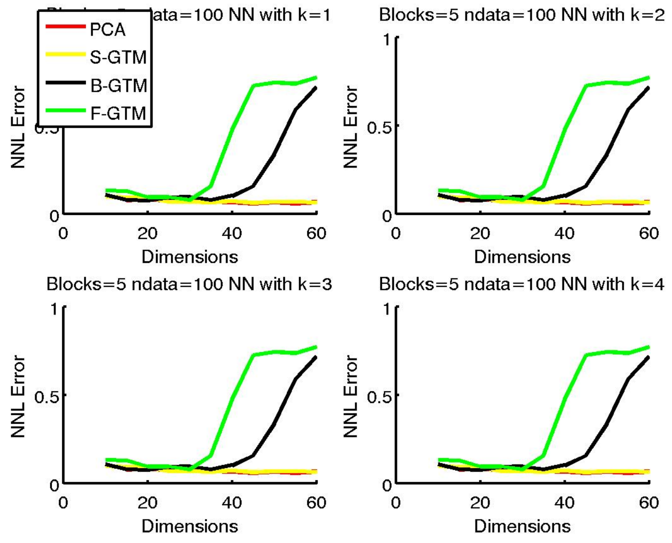


Figure 7: Comparing the different methods with 5 blocks in the covariance. The NNL Error is calculated for different values of k.

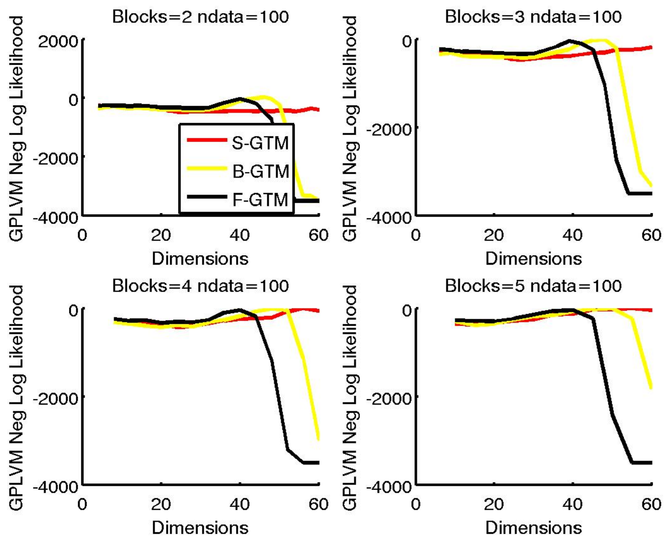


Figure 8: Comparing the different methods using the GP-LVM selection likelihood. The different block sizes in the covariance matrix range from 2 to 5.

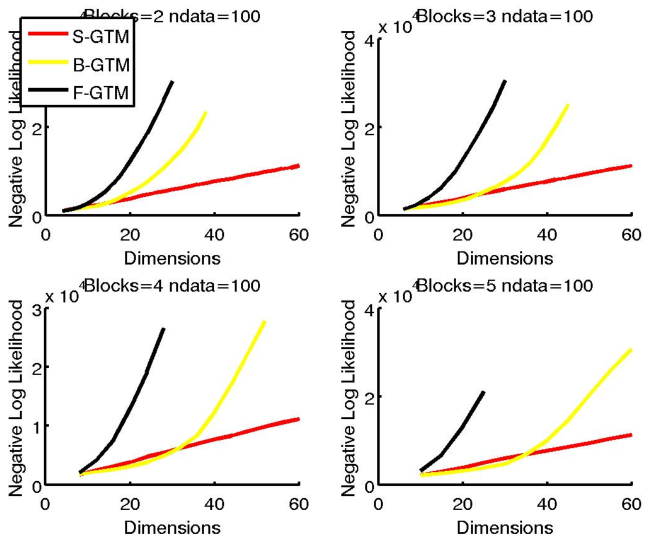


Figure 9: Comparing the different methods using the average model log likelihood. The different block sizes in the covariance matrix range from 2 to 5.

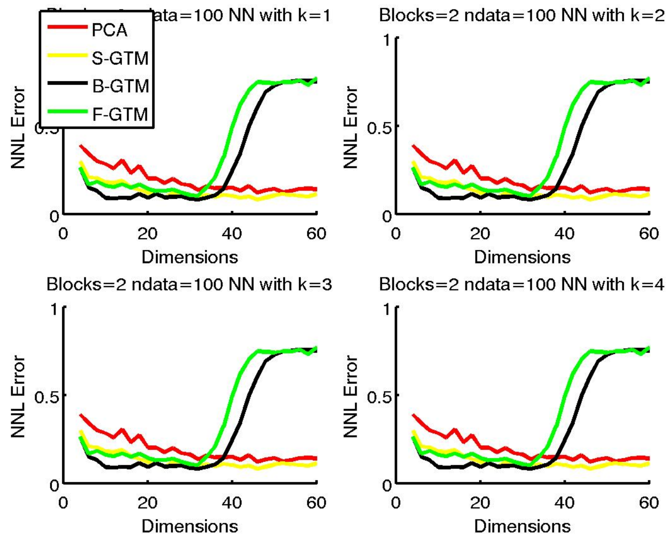


Figure 10: Comparing the different methods with 2 blocks in the covariance. The NNL Error is calculated for different values of *k*.

4.3 Moderately structured data

As an example for medium structured cases we have chosen the covariances to be 10 around the single Gaussian, which results in an overall standard deviation of 7 in the data set. Figures (10 to 13) show how the NNL Error changes with increasing dimensionality of the data set. Compared to the low structured the case the results differ a little depending on the block size. Block GTM always performs better than the other methods until the dimensionality reaches 36 and then quickly breaks down when 40 dimensions are reached. The full GTM model performs worse than block GTM but also outperforms the spherical GTM and PCA. However depending on the group size,the full GTM loses its advantage relatively quickly until it breaks down at a dimensionality of about 36. PCA constantly performs worse than spherical GTM but comes closer to the performance of spherical GTM with increasing dimensionality. It is also interesting to note that the performance of both PCA and spherical GTM increases up to a dimensionality of 20–26 and then stays constant. Figure (14) shows the average GP-LVM likelihood which shows the same strange behaviour as in the low structure case. Figure (15) shows the model log likelihood of the different GTMs over increasing dimensionality in the data set. The results correspond to the findings in the low structure case but they exhibit a bit more strongly the advantage of the block GTM especially when dealing with data between 16 and 30 dimensions.

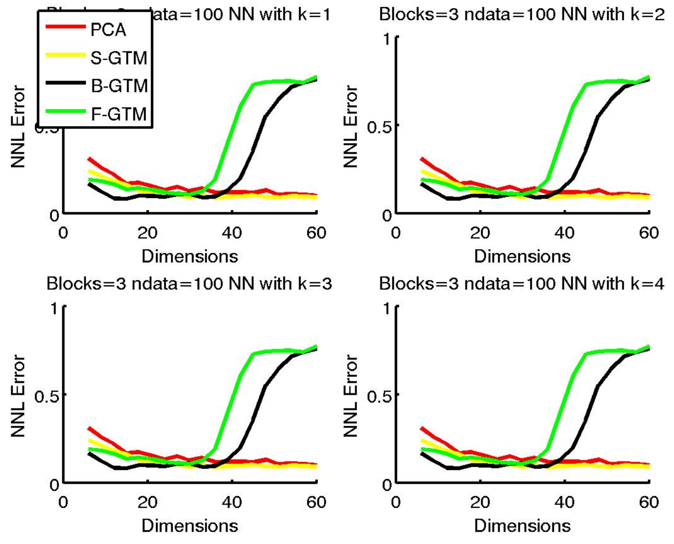


Figure 11: Comparing the different methods with 3 blocks in the covariance. The NNL Error is calculated for different values of k.

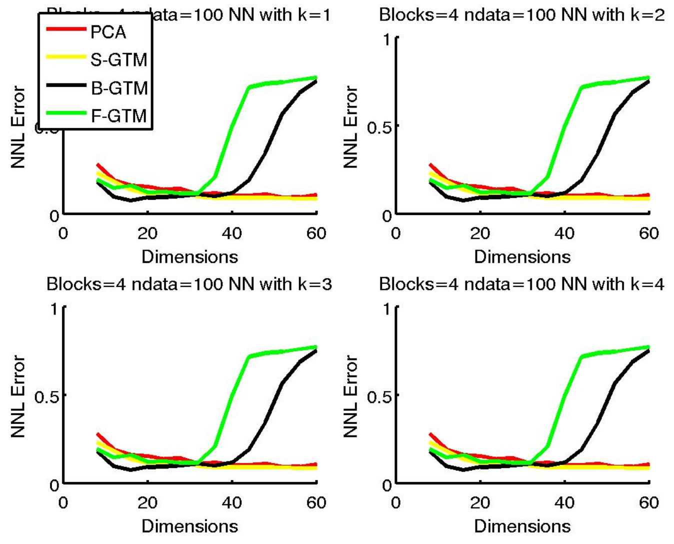


Figure 12: Comparing the different methods with 4 blocks in the covariance. The NNL Error is calculated for different values of k.

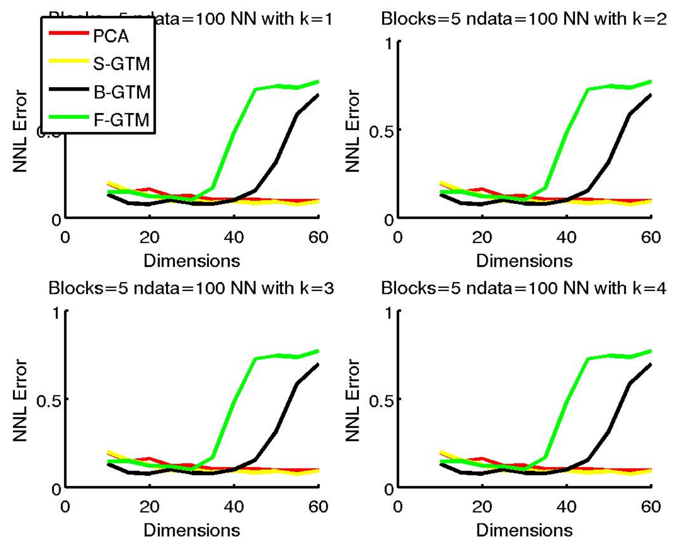


Figure 13: Comparing the different methods with 5 blocks in the covariance. The NNL Error is calculated for different values of k.

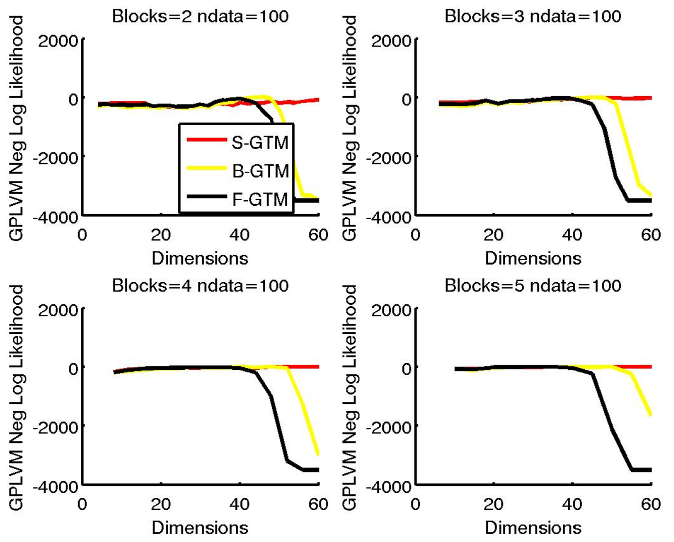


Figure 14: Comparing the different methods using the GP-LVM selection likelihood. The different block sizes in the covariance matrix range from 2 to 5.

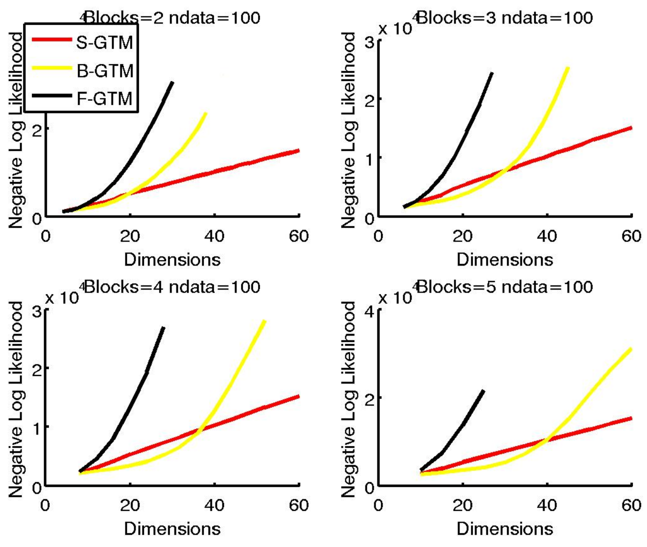


Figure 15: Comparing the different methods using the average model log likelihood. The different block sizes in the covariance matrix range from 2 to 5.

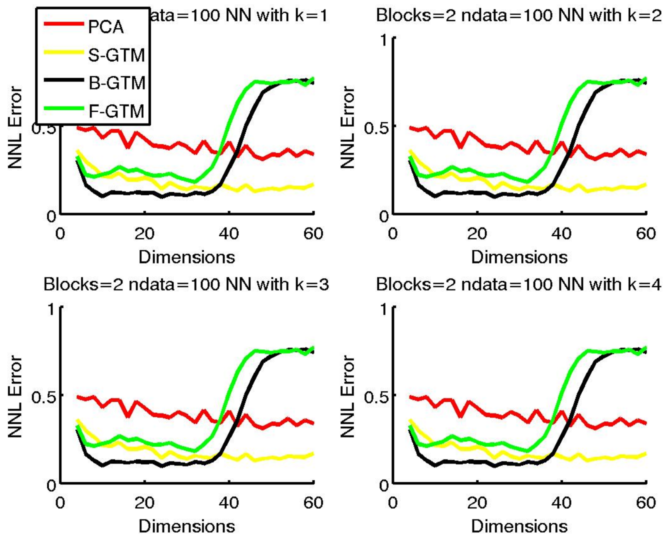


Figure 16: Comparing the different methods with 2 blocks in the covariance. The NNL Error is calculated for different values of k.

4.4 Highly structured data

As an example for high structured cases we have chosen the covariances to be 20 around the single Gaussian, which results in an overall standard deviation of 7.55 in the data set. Figures (16 to 19) show how the NNL Error changes with increasing dimensionality of the data set. The results in the high structure case separate the methods clearly. Block GTM performs better than the other methods until the breakdown at 40 dimensions. The full GTM model only performs better then the spherical GTM at very low dimensionality and with increasing number of blocks the effect diminishes. PCA always performs worse than all the other methods until these break down but again is approximating the performance of spherical GTM with increasing block size and dimensionality. Figures (20) and (21) show essentially the same behaviour as in the previous experiments.

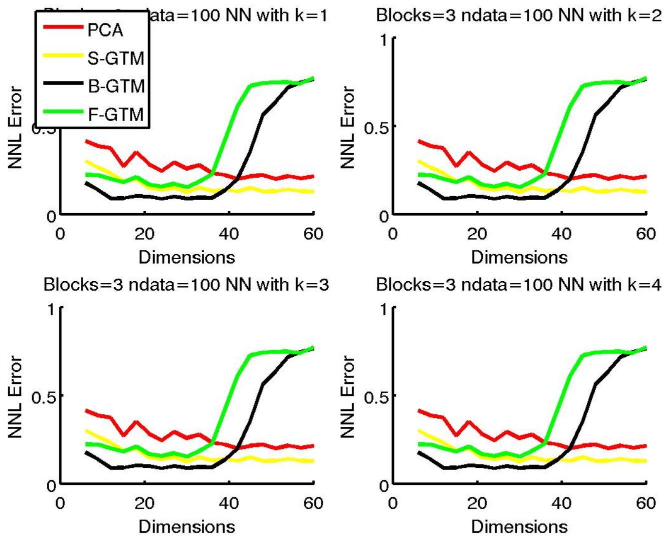


Figure 17: Comparing the different methods with 3 blocks in the covariance. The NNL Error is calculated for different values of k.

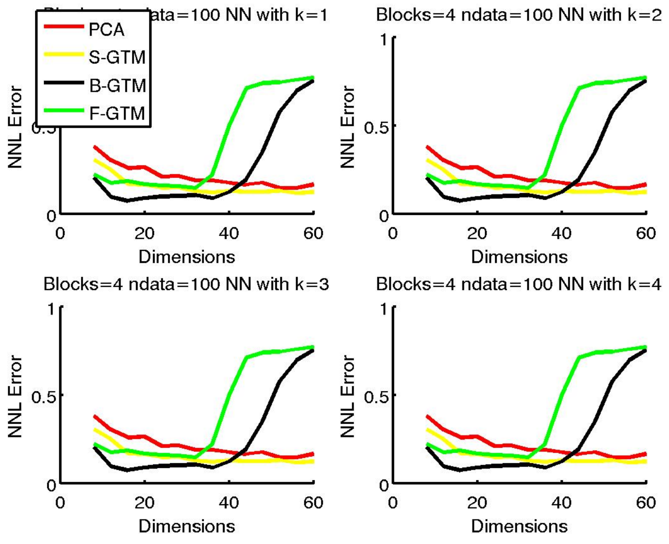


Figure 18: Comparing the different methods with 4 blocks in the covariance. The NNL Error is calculated for different values of k.

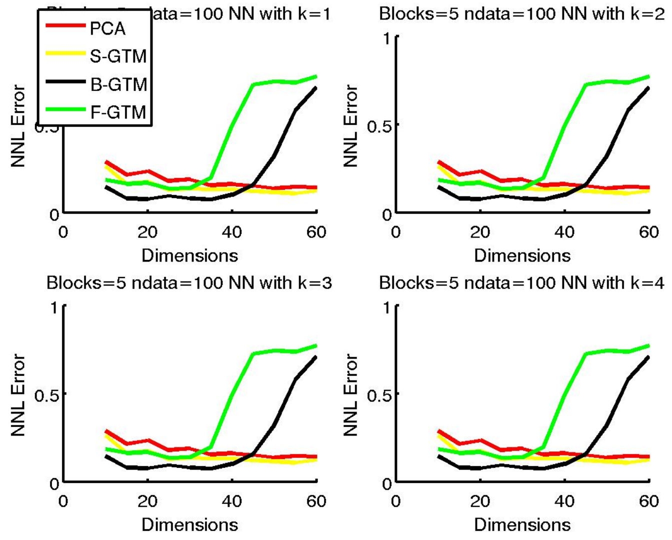


Figure 19: Comparing the different methods with 5 blocks in the covariance. The NNL Error is calculated for different values of k.

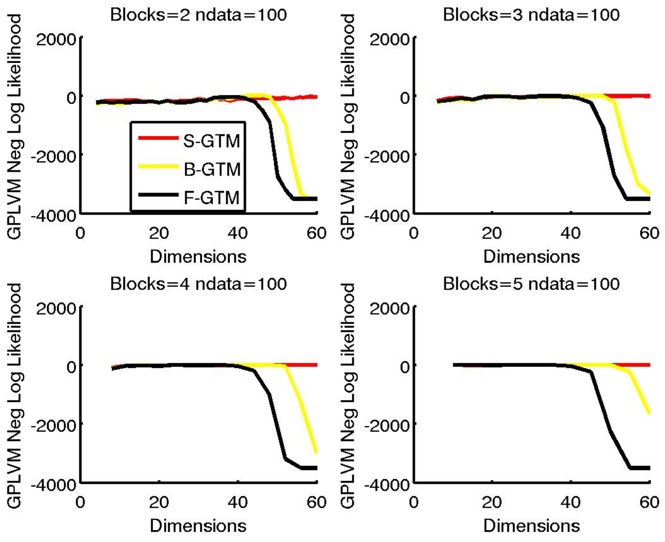


Figure 20: Comparing the different methods using the GP-LVM selection likelihood. The different block sizes in the covariance matrix range from 2 to 5.

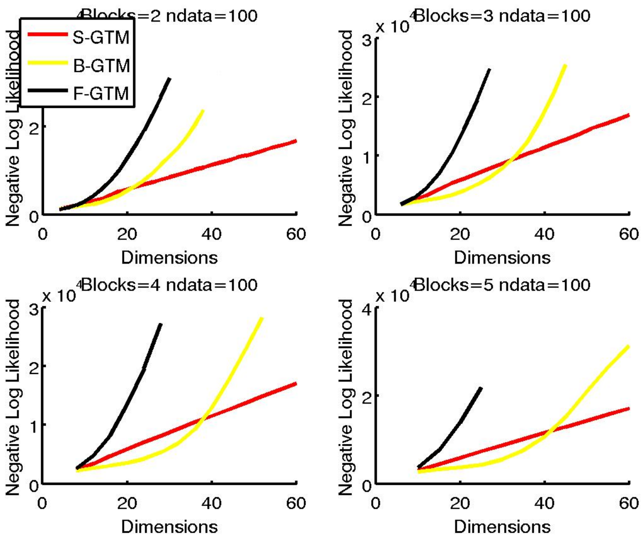


Figure 21: Comparing the different methods using the average model log likelihood. The different block sizes in the covariance matrix range from 2 to 5.

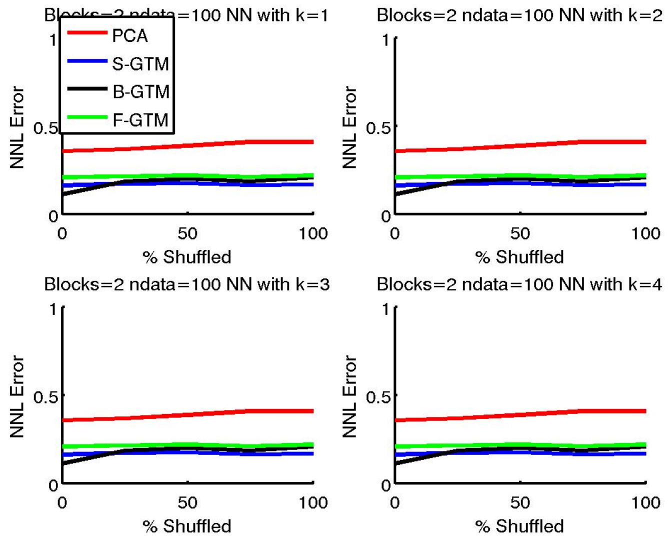


Figure 22: Shuffle Experiment to compare the different methods with 2 blocks in the covariance. The NNL Error is calculated for different values of k

4.5 Shuffle Experiment

The shuffle experiment was conducted on a 24-dimensional highly structured data set and thus it has a covariance of 20 for the blocks in the single Gaussian and a standard deviation of 7.55 for the whole data set. Figures (22 to 24) show the results for block sizes from 2 to 4. With 0 percent shuffle, block GTM is first, spherical GTM second, full GTM third and PCA performs worst. The performance of the last three models stays constant as expected, only being altered by the random effects due to the small size of repetitions when rerunning the experiment with different random shuffle patterns. The performance of the block GTM however deteriorates as the number of shuffled variables increases, as expected. Interestingly, the performance tends towards the level of spherical GTM or slightly worse. Similar behaviour can be seen for the log likelihood in Figure (25), with the only exception that block GTM has a worse likelihood than spherical GTM in the case of 2 blocks. In this case the likelihood approaches a constant level as well, which is slightly higher than the one for spherical GTM. The GP-LVM likelihood in Figure (26) is again somewhat hard to interpret though the general trend is the same; however it indicates an even stronger advantage for the block GTM.

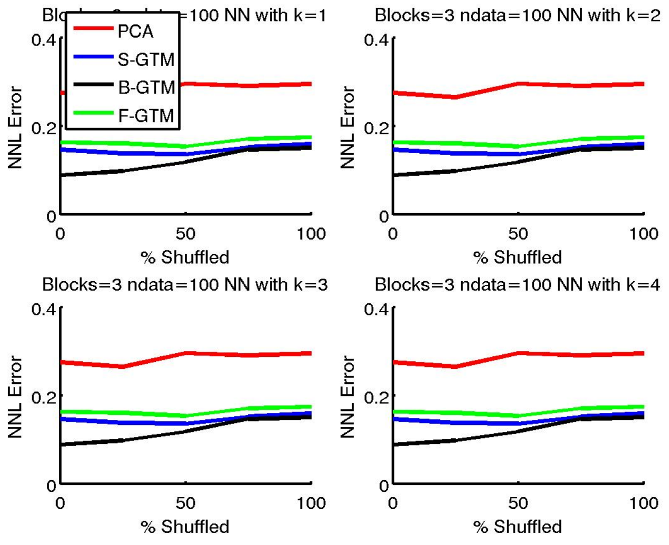


Figure 23: Shuffle Experiment to compare the different methods with 3 blocks in the covariance. The NNL Error is calculated for different values of k

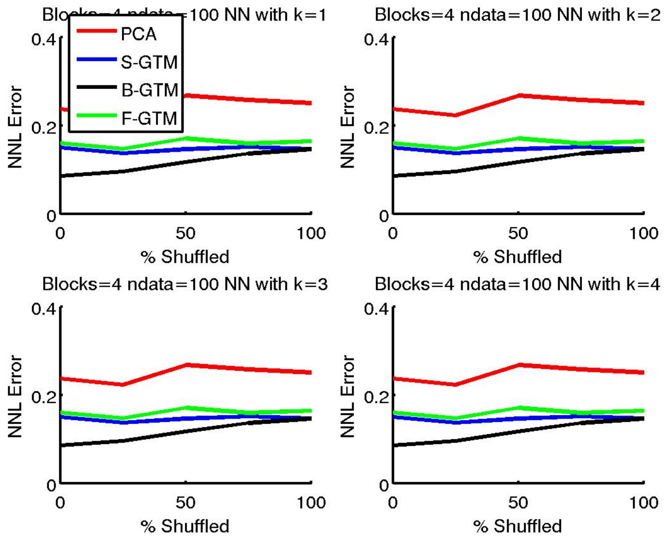


Figure 24: Shuffle Experiment to compare the different methods with 4 blocks in the covariance. The NNL Error is calculated for different values of k

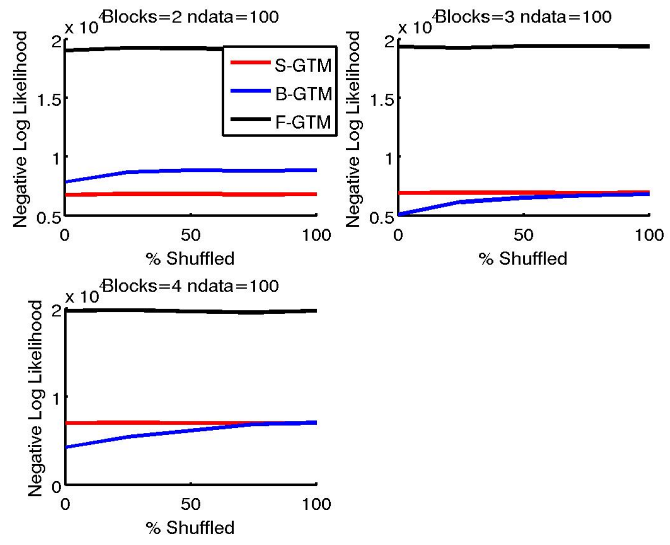


Figure 25: Shuffle Experiment to compare the different methods using the average model log likelihood. The different block sizes in the covariance matrix range from 2 to 4.

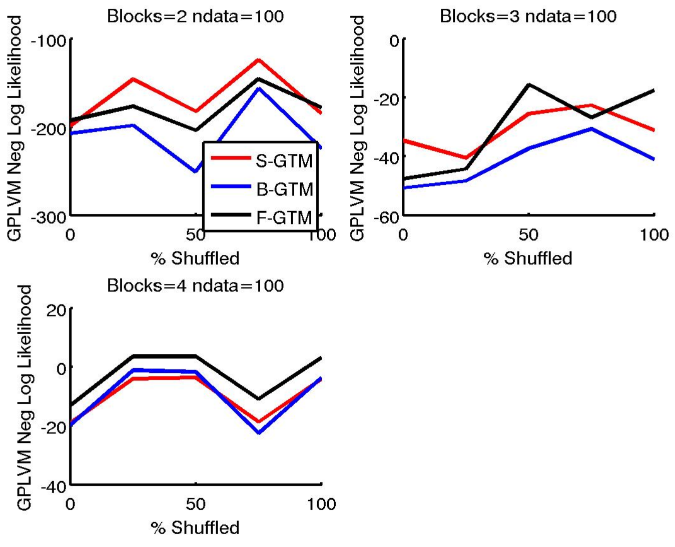


Figure 26: Shuffle Experiment to compare the different methods using the GP-LVM selection likelihood. The different block sizes in the covariance matrix range from 2 to 4.

5 Conclusions

The experiments show that the block GTM is very promising. The main limitation is the abrupt breakdown when the data set dimensionality is too great. Empirical results not shown in this report indicate that the sample size of the data does not change the boundary at 40 dimensions. If one has 40 or fewer dimensions it better to include information about the grouping of the variables. In the worst cases, when there is no strong expression of this grouping in the data or a misspecification of the grouping, the block GTM will perform at the same level as spherical GTM. The experiments also show that the use of the GP-LVM framework to assess the quality of the model has to be considered very carefully. It failed to detect when the methods break down and essentially project all the data onto one point. This seems to be optimal for fitting the GP-LVM framework to the data, when optimising the hyper parameters and thus causes the rapid decrease in likelihood. It also seems to be very prone to slight fluctuation in the data since the sample size was not large enough to correct for random effects in case of the shuffle experiments. However it is one of the few helpfull tools to assess unsupervised learning, especially when one cannot engineer synthetic data to test the method or wants to compare against non-probabilistic data projection methods.

6 Future Work

Future work in this area will be aimed at assessing the possibility of including methods like Bayesian Correlation Estimation [8] into the algorithm in order to learn the correlation structure rather than rely on it being imposed *a priori*. This would require a variational formulation of the GTM algorithm. It would also be interesting to find out what part of the algorithm is causing the break down of performance at 40 dimensions, since this does not seem to be related to the sample size. Further we had severe problems when trying to extend the algorithm for block GTM to handle missing data. The EM algorithm became highly unstable and we are currently investigating this problem.

References

[1] C. M. Bishop, M. Svensen, and C. K. I. Williams. Developments of the generative topographic mapping. *Neurocomputing*, 21:203–224, 1998.

[2] I. Borg and P Groenen. *Modern Multidimensional Scaling: theory and applications*. Springer-Verlag New York, 2005.

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

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

[5] 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.

[6] Stefan Harmeling. Exploring model selection techniques for nonlinear dimensionality reduction. Technical report, Edinburgh University, Scotland, 2007.

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

[8] Merrill W. Liechty John C. Liechty and Peter Mller. Bayesian correlation estimation. *Biometrika*, 91:1–14, 2004.

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

[10] Neil D. Lawrence. A scaled conjugate gradient algorithm for fast supervised learning. *Journal of Machine Learning Research 6*, page 1783?1816, 2005.

[11] 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.

[12] S.T. Roweis and L.K. Saul. Locally linear embedding. *Science*, 290:2323–2326, 2000.

[13] Chong Ho Yu. Resampling methods: concepts, applications, and justification. *Practical Assessment, Research and Evaluation*, 8, 2003.