GTM-based Data Visualisation with Incomplete Data

Yi Sun, Peter Tiňo, and Ian Nabney

Neural Computing Research Group, Aston University,
Aston Triangle, Birmingham B4 7ET
United Kingdom
{suny,tinop,nabneyit}@aston.ac.uk
http://www.ncrg.aston.ac.uk/

Abstract. We analyse how the Generative Topographic Mapping (GTM) can be modified to cope with missing values in the training data. Our approach is based on an Expectation-Maximisation (EM) method which estimates the parameters of the mixture components and at the same time deals with the missing values. We incorporate this algorithm into a hierarchical GTM. We verify the method on a toy data set (using a single GTM) and a realistic data set (using a hierarchical GTM). The results show our algorithm can help to construct informative visualisation plots, even when some of the training points are corrupted with missing values.

1 Introduction

Data visualisation, which plays a key role in developing good models for large quantities of data, is an important aid in dimension reduction, gives information about local deviations in performance and provides a useful check for objective quantitative measures. However, in many applications the input data is incomplete. Therefore it is important to know how to use the available data and how to reconstruct the missing values. For example, in the pharmaceutical field, scientists use computer modelling to examine and analyse the molecular structure of compounds and high throughput screening to assess their interaction with biological targets. Many compounds are not screened against a complete set of targets, yet we do not want to exclude all such compounds from data analysis since that risks missing potential drugs.

The hierarchical generative topographic mapping (GTM) model is an interactive data visualisation technique, which enables the user to construct arbitrarily detailed projection plots. The basic building block is the GTM [1] . The problem considered here is to train the GTM model with incomplete data and reconstruct the missing values. This way the data, including the missing components, can be shown in a visualisation plot that is as "faithful" as possible. For hierarchical GTM, the incomplete data can be displayed at all levels of the hierarchy of visualisation plots.

Our algorithm can be described briefly as follows. A joint density model of the data is learned in an unsupervised way from the incomplete training data set by using an EM algorithm. For visualisation purposes, the missing data is filled in by computing the posterior mean. In [2], the GTM was trained only with complete data, and an additional condition was added to reconstruct the missing data. In contrast, our algorithm is more generic.

Since our algorithm is based on Gaussian mixture models (GMM) and the EM algorithm, in section 2 we briefly introduce the EM algorithm for GMMs. The GTM with incomplete data algorithm is detailed in section 3. Section 4 gives a basic introduction to hierarchical GTM. We illustrate the algorithm in section 5 with a toy data and a high dimensional data set from flow diagnostics of an oil pipeline. Section 6 discusses the result.

2 The EM Algorithm for Gaussian Mixture Models

The EM algorithm is especially relevant since it is a general method for parameter estimation in mixture models that is based on the idea of filling in missing data. This section introduces briefly the algorithm for finding the maximum likelihood parameters of a Gaussian mixture model [3].

We consider a mixture density

$$P(\mathbf{t}_n) = \sum_{j=1}^{K} P(\mathbf{t}_n | j; \theta_j) P(j), \tag{1}$$

which is generating the (i.i.d.) data $\mathbf{T} = \{\mathbf{t}_n\}_{n=1}^{N}$. In this case each component of the mixture is denoted by j and parametrised by θ_j, and $P(j)$ is the prior probability for the mixture component j. Then the log likelihood of the parameters given the data set is

$$\mathcal{L}(\theta) = \sum_{n=1}^{N} \log \sum_{j=1}^{K} P(\mathbf{t}_n | j; \theta_j) P(j). \tag{2}$$

The binary indicator variables z_{nj} are introduced to specify which component of the mixture generated the data point. $z_{nj} = 1$ if and only if $\mathbf{t}_n$ is generated by component j, otherwise $z_{nj} = 0$. Then equation (2) can be re-written as the complete data log likelihood function:

$$\mathcal{L}_c(\theta) = \sum_{n=1}^{N} \sum_{j=1}^{K} z_{nj} \log[P(\mathbf{t}_n | z_{nj}; \theta) P(z_{nj}; \theta)]. \tag{3}$$

Since z_{nj} is not known, the expectation $E[z_{nj} | \mathbf{t}_n, \theta_j]$ of z_{nj} given the current parameter values θ_j is computed. This is the probability that the Gaussian j generated the data point $\mathbf{t}_n$ and is denoted by r_{nj}. This is the E-step of the EM algorithm:

$$r_{nj} = \frac{|\Sigma_j|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{t}_n - \mu_j)^T \Sigma_j^{-1} (\mathbf{t}_n - \mu_j)\} P(j)}{\sum_{k=1}^{K} |\Sigma_k|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{t}_n - \mu_k)^T \Sigma_k^{-1} (\mathbf{t}_n - \mu_k)\} P(k)}. \tag{4}$$

The means μ_j and covariance matrices Σ_j of the jth component Gaussian are updated in the M-step using the data set weighted by the r_{nj}:

$$\mu_j^{t+1} = \frac{\sum_{n=1}^{N} r_{nj} \mathbf{t}_n}{\sum_{n=1}^{N} r_{nj}} \tag{5}$$

$$\Sigma_j^{t+1} = \frac{\sum_{n=1}^{N} r_{nj} (\mathbf{t}_n - \mu_j^{t+1})(\mathbf{t}_n - \mu_j^{t+1})^T}{\sum_{n=1}^{N} r_{nj}} \tag{6}$$

The equations above are for full covariance matrices, but there are similar equations for other covariance structures.

3 Generative Topographic Mapping and Incomplete Data

3.1 The Generative Topographic Mapping

The generative topographic mapping (GTM) [1] is a nonlinear latent variable model that uses latent (or hidden) variables to model a probability distribution in the data space. It is a constrained mixture of Gaussians whose parameters are optimised using the expectation-maximisation (EM) algorithm.

For the GTM, $\mathbf{t}$ denotes the data in a D-dimensional Euclidean space and $\mathbf{x}$ denotes the latent variables in an L-dimensional latent space. Considering a non-linear transformation from the latent space to the data space using a radial basis function network(see e.g. [4]), the latent data is mapped to data space by a radial basis function $\mathbf{y} = \mathbf{W\Phi}(\mathbf{x})$ with weights $\mathbf{W}$ and a basis function matrix $\mathbf{\Phi}$. The goal of the latent variable model is to find a representation for the distribution $p(\mathbf{t})$ in terms of a number K of latent points $\mathbf{x}_j (j = 1, 2, ...K)$ and corresponding Gaussian distributions centred on $\mathbf{y}(\mathbf{x_j}; \mathbf{W})$ [1]. The data density is defined by

$$P(\mathbf{t}|\mathbf{W}, \beta) = \frac{1}{K} \sum_{j=1}^{K} P(\mathbf{t}|\mathbf{x_j}, \mathbf{W}, \beta) \tag{7}$$

and

$$P(\mathbf{t}|\mathbf{x_j}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{ -\frac{\beta}{2} \|\mathbf{y}(\mathbf{x_j}; \mathbf{W}) - \mathbf{t}\|^2 \right\} \tag{8}$$

where $\mathbf{W}$ and the inverse variance β can be fitted by maximum likelihood with the EM algorithm.

The latent space representation of the point $\mathbf{t}_n$, i.e. the *projection* of $\mathbf{t}_n$, is taken to be the mean $\sum_{j=1}^{K} r_{nj}\ \mathbf{x}_j$ of the posterior distribution on the latent space.

3.2 Incorporating missing values into the EM algorithm for the GTM model

To handle missing values in the data set, we write data points $\mathbf{t}_n$ as $(\mathbf{t}_n^o, \mathbf{t}_n^m)$, where each data vector can have different patterns of missing components; m and

o denote subvectors and submatrices of the parameters matching the missing and observed components of the data. The EM algorithm treats both the indicator variables z_{nj} and the missing inputs $\mathbf{t}_n^m$ as hidden variables. For the GTM, as the covariance matrix is constrained to be isotropic, $\Sigma_j = \beta^{-1}\mathrm{I}$, the covariance of missing and observed values Σ_j^{mo} is equal to 0. The expected value in the E-step is taken with respect to both sets of hidden variables. If we knew the values of the indicator variables z_{nj}, we would write the negative log likelihood function as

$$\mathcal{L}(\mathrm{W},\beta) = \sum_{n=1}^{N}\sum_{j=1}^{K} z_{nj}\Big\{\frac{D}{2}\ln(2\pi) - \frac{D}{2}\ln\beta + \frac{\beta}{2}\Big[\parallel \mathbf{t}_n^o - \mathbf{y}_j^o \parallel^2 + \\ \parallel \mathbf{t}_n^m - \mathbf{y}_j^m \parallel^2 \Big]\Big\} \tag{9}$$

After taking the expectation, the sufficient statistics for the parameters include three unknown terms, z_{nj}, $z_{nj}\mathbf{t}_n^m$ and $z_{nj}\mathbf{t}_n^m\mathbf{t}_n^m$. So we must calculate the expectations for these three terms. Following [5], we introduce:

$$\hat{\mathbf{t}}_{nj}^m = \mathrm{E}(\mathbf{t}_n^m | z_{nj} = 1, \mathbf{t}_n^o, \theta_j) = (\mathbf{y}_j^m)^{old} \tag{10}$$

which is the least-squares regression between $\mathbf{t}_n^m$ and $\mathbf{t}_n^o$ predicted by Gaussian j, and 'old' denotes the value computed in the last M-step.

The expectation of z_{nj} is $E[z_{nj}|\mathbf{t}_n^o, \theta_j] = r_{nj}$ (equation (4)) measured only on the observed dimensions of $\mathbf{t}_n$. For the GTM, we calculate:

$$\begin{aligned}\mathrm{E}[z_{nj}\mathbf{t}_n^m|\mathbf{t}_n^o,\theta_j] = \mathrm{E}[z_{nj}|\mathbf{t}_n^o,\theta_j]\mathrm{E}[\mathbf{t}_n^m|z_{nj}=1,\mathbf{t}_n^o,\theta_j] &= r_{nj}\hat{\mathbf{t}}_{nj}^m \\ &= r_{nj}(\mathbf{y}_j^m)^{old}\end{aligned} \tag{11}$$

In the M-step, the missing values are expressed using the posterior means:

$$E[\mathbf{t}_n^m|\mathbf{t}_n^o,\theta_j] = \sum_{j=1}^{K} r_{nj} E[\mathbf{t}_n^m | z_{nj} = 1, \mathbf{t}_i^o, \theta_j] \tag{12}$$

and the weights are then updated to $\mathbf{W}_{new}$ as used way for GTM [1]. The variance is updated by:

$$\beta^{-1} = \frac{1}{\mathrm{ND}}\sum_{n=1}^{N}\sum_{j=1}^{K} r_{nj}\left(\|\mathbf{t}_n^o - \mathbf{y}_j^o\|^2 + \mathrm{E}[z_{nj}\|\mathbf{t}_n^m - \mathbf{y}_j^m\|^2]\right) \tag{13}$$

where

$$\begin{aligned}\mathrm{E}[z_{nj}\|\mathbf{t}_n^m - \mathbf{y}_j^m\|^2] &= \mathrm{E}[\|\mathbf{t}_n^m - \mathbf{y}_j^m\|^2 | z_{nj} = 1] \\ &= (\beta^{-1})^{old} + (\hat{\mathbf{t}}_{nj}^m)^T(\hat{\mathbf{t}}_{nj}^m) - 2(\hat{\mathbf{t}}_{nj}^m)^T\mathbf{y}_j^m + (\mathbf{y}_j^m)^T\mathbf{y}_j^m\end{aligned} \tag{14}$$

and $\mathbf{y}_j^m = (\mathbf{W}_{new}\mathbf{\Phi}(\mathbf{x_j}))^m$.

4 Hierarchical GTM

4.1 An introduction to hierarchical GTM

For a complex data set, a single two-dimensional visualisation plot may not be sufficient since it is difficult to capture all of the interesting aspects in the data set. Therefore a hierarchical visualisation system is desirable.

Given a training data set T $= \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\}$, the probability, assigned to this set by a hierarchy of GTMs organised in hierarchical tree $\mathcal{T}$, is calculated by considering the hierarchical GTM $\mathcal{T}$ as a mixture of GTMs [6], with mixture components being the leaves $\mathcal{M}$. The parameters of the hierarchy (weights $\mathbf{W}$, inverse variance β and parent-conditional mixture coefficients) are fitted by maximum likelihood using the EM algorithm. Mixture coefficients for the mixture components $\mathcal{M}$ are calculated recursively by multiplying parent-conditional mixture coefficients down the path from the root to $\mathcal{M}$.

Given a data point $\mathbf{t}_n$ and a submodel $\mathcal{M}$ in the hierarchy $\mathcal{T}$, we have three types of hidden variables: 1) Responsibility of Parent($\mathcal{M}$), the parent of $\mathcal{M}$, for generating t_n. 2) Parent-conditional responsibility for $\mathbf{t}_n$, given that Parent($\mathcal{M}$) generated t_n, and 3) Responsibility of latent space centres $\mathbf{x}_j$ of $\mathcal{M}$ for generating $\mathbf{t}_n$.

To avoid numerical problems arising from multiplication of small probabilities and to speed up the training process, the GTMs on deeper levels are trained only on data points for which the parent model has responsibility greater than some pre-set threshold ϵ. In our experiments $\epsilon = 10^{-3}$.

4.2 Parameter initialisation

Having trained GTMs down to level ℓ of the hierarchical tree $\mathcal{T}$, we choose a parent model $\mathcal{N}$ at level ℓ and, based on its visualisation plot, we select "regions of interest" for child GTMs $\mathcal{M}$ at level $\ell + 1$. More precisely, the visualisation plot of the parent GTM $\mathcal{N}$ shows low-dimensional representations in the latent space of data space points from the training set.

The regions of interest are selected as follows: The user first selects points $\mathbf{c}_i$, $i = 1, 2, ..., A$, in the latent space that correspond to "centres" of the subregions the user is interested in. The points $\mathbf{c}_i$ are then transformed via the map $\mathbf{y}_\mathcal{N}$ defined by the parent GTM $\mathcal{N}$ to the data space

$$\mathbf{y}_\mathcal{N}(\mathbf{c}_i) = \mathbf{W}_\mathcal{N}\ \mathbf{\Phi}_\mathcal{N}(\mathbf{c}_i) \tag{15}$$

The regions of interest are given by the Voronoi compartments [7] in the data space corresponding to the points $\mathbf{y}_\mathcal{N}(\mathbf{c}_i)$, $i = 1, 2, ..., A$:

$$V_i = \left\{ \mathbf{t} \in \Re^D |\ d\left(\mathbf{t}, \mathbf{y}_\mathcal{N}(\mathbf{c}_i)\right) = \min_j d\left(\mathbf{t}, \mathbf{y}_\mathcal{N}(\mathbf{c}_j)\right) \right\}, \tag{16}$$

where $d(\cdot, \cdot)$ is the Euclidean distance in the data space $\Re^D$. All points in V_i are allocated to the "centre" $\mathbf{y}_\mathcal{N}(\mathbf{c}_i)$.

We initialise the parameters $\mathbf{W}_{\mathcal{M}}$ of child GTMs $\mathcal{M}$, so that each GTM initially approximates principal component analysis (PCA) of the corresponding Voronoi compartment. For GTM $\mathcal{M}$ corresponding to a compartment V_i, we first evaluate the covariance matrix of training points in V_i and obtain the first L principal eigenvectors. Next, we determine $\mathbf{W}_{\mathcal{M}}$ by minimising the error

$$E = \frac{1}{2} \sum_{j=1}^{K_{\mathcal{M}}} ||\mathbf{W}_{\mathcal{M}} \; \mathbf{\Phi}_{\mathcal{M}}(\mathbf{x}_j^{\mathcal{M}}) \; - \; \mathbf{U} \; \mathbf{x}_j^{\mathcal{M}}||^2, \tag{17}$$

where the columns of $\mathbf{U}$ are the first L principal eigenvectors of the data covariance matrix (see [1]).

Following [1], parameter $\beta_{\mathcal{M}}$ is initialised to be the larger of the $L+1$ eigenvalue from PCA, that represents the variance of the data away from the PCA plane , or the square of half of the grid spacing of the PCA-projected latent data points in data space.

5 Experiment

In our experiments, GTM models were trained in two ways: (A1) the algorithm defined in section 3.2 and (A2) standard EM applied to a dataset with the missing values replaced by the unconditional mean.

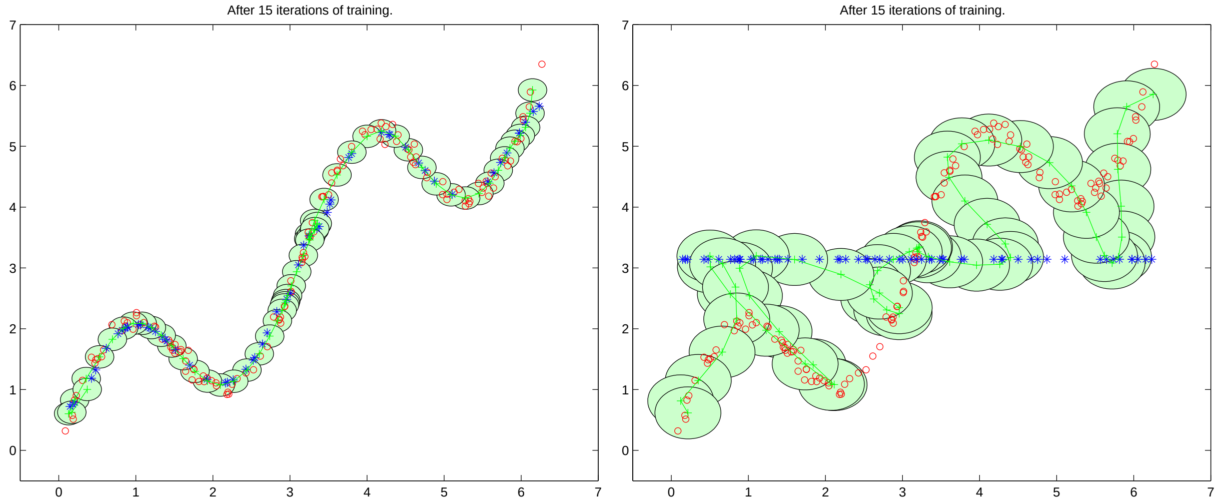
5.1 The toy data

200 training data points were generated randomly in the interval $[0, 2\pi]$ as $\mathbf{t}_1$. The variable $\mathbf{t}_2$ was then computed by the function $\mathbf{t}_2 = \mathbf{t}_1 + 1.25 \sin(2\mathbf{t}_1)$. A spherical Gaussian noise with standard deviation 0.1 was added to $\mathbf{t}_2$ coordinates. Then we deleted 30% of the values in $\mathbf{t}_2$ randomly. Figure 1 shows the result using A1 and A2. After training, the negative log likelihood is 1.62 and 2.66 per data respectively.

5.2 Oil data

This example arises from the problem of determining the fraction of oil in a multiphase pipeline carrying a mixture of oil, water and gas. The data set consists of 1000 12-dimensional points. Points in the data set are classified into three different multi-phase flow configurations: homogeneous, annular and laminar [8].

Figure 2 shows the visualisation results. A hierarchy of GTMs up to level 3 was trained on the data set. For every level, $15 \times 15 = 225$ latent data points were selected in the 2-dimensional latent space and the number of Gaussian basis functions is $4 \times 4 = 16$. The final visualisation plot for the complete (uncorrupted) data can be seen in figure 2(a). For the top level, after 10 training iterations, the negative log likelihood is -3.93 per data point.



(a)Using the EM based algorithm

(b)Using unconditional mean method

Fig. 1. The toy problem: the complete data points are plotted as circles while the centres of the Gaussian mixture are plotted as plus signs. The centres are joined by a line according to their ordering in the (one-dimensional) latent space (K = 60). The stars represent the missing values. The discs surrounding each plus sign represent two standard deviations' width of the noise model.

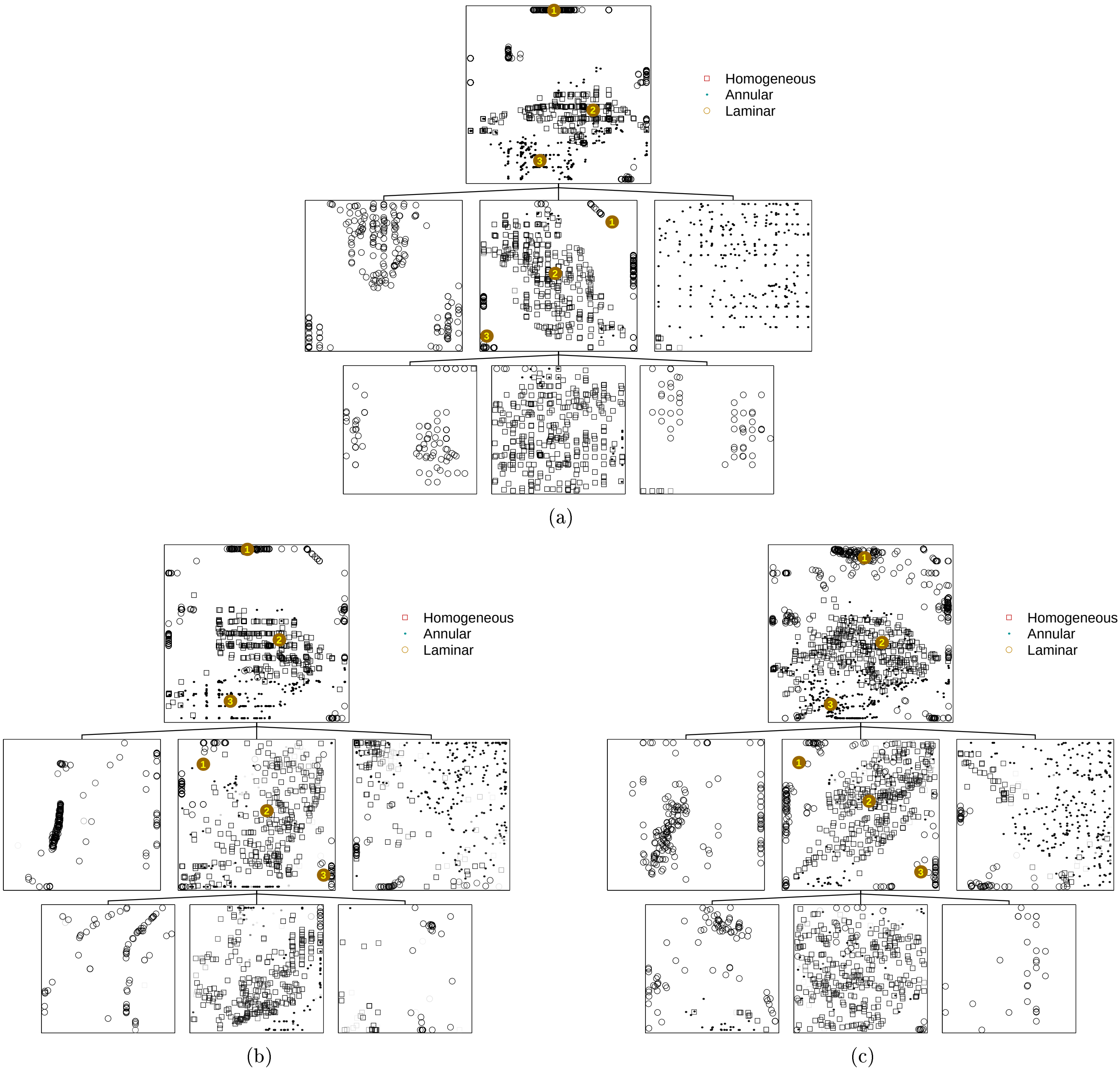
We randomly deleted 30% of values in the data set. The maximum number of corrupted coordinates per data point is 6. Again we compare the negative log likelihood of A1 and A2. Here we just measured the values of negative log likelihood for the top level GTM, since the likelihood for lower level models depends on where the "regions of interest" are selected. For the incomplete data set, after 10 training cycles, using the EM algorithm, the negative log likelihood is -3.39 per data point, while using unconditional mean filling in the missing data, the negative log likelihood is -1.31. Using our EM based algorithm for dealing with missing values can indeed be beneficial as it can be seen by comparing the top level (root) visualisation plots and the second visualisation plots on the second level of the hierarchy. These second-level plots show better separation of classes and match better to the models trained on the complete data set.

6 Conclusions

In this paper, we have shown how incomplete data can be included in the hierarchical GTM training. The algorithm for dealing with missing values based on the EM algorithm and Gaussian mixture models is a viable approach for data visualisation. It is preferable to the simple strategy of just filling-in the missing values with unconditional means.

References

1. C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: The Generative Topographic Mapping. *Neural Computation*, 10(1):215–235, 1998.
2. M. Á. Carreira-Perpiñán. Reconstruction of Sequential Data with Probabilistic Models and Continuity Constraints. In Sara A. Solla, Todd K. Leen, and Klaus-Robert Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. The MIT Press, 2000.
3. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Roy. Stat. Soc. B*, 39:1–38, 1977.
4. C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, New York, N.Y., 1995.
5. Z. Ghahramani and M. I. Jordan. Learning from incomplete data. Technical report, AI Laboratory, MIT, 1994.
6. P. Tino and I. Nabney. Constructing localized non-linear projection manifolds in a principled way: hierarchical Generative Topographic Mapping. Technical report, 2000.
7. F. Aurenhammer. Voronoi diagrams - survey of a fundamental geometric data structure. *ACM Computing Surveys*, 3:345–405, 1991.
8. C. M. Bishop and G. D. James. Analysis of Multi-phase Flows Using Dual-energy Gamma Densitometry and Neural Networks. *Nuclear Instruments and Methods in Physics Research, A*, 327:580–593, 1993.



(a)

(b)

(c)

Fig. 2. Data visualisation for oil data by using hierarchical GTM. Plot (a) shows the result of training on the complete data set. Plot (b) shows the result of using the EM algorithm learning from incomplete data, while plot (c) shows the same data set visualised using the unconditional mean to fill in the missing data.