Chapter 1

VISUALISATION OF INCOMPLETE DATA USING CLASS INFORMATION CONSTRAINTS

Yi Sun, Peter Tiňo and Ian Nabney

Neural Computing Research Group

Aston University, Birmingham, B4 7ET, UK

suny,tinop,nabneyit@aston.ac.uk

Abstract We analyse how the training algorithm for the Generative Topographic Mapping (GTM) can be modified to use class information to improve results on incomplete data. The approach is based on an Expectation-Maximisation (EM) method which estimates the parameters of the mixture components and missing values at the same time; furthermore, if we know the class membership of each pattern, we can improve the generic algorithm by eliminating multi-modalities in the posterior distribution over the latent space centres. We evaluate the method on a toy problem and a realistic data set. The results show that our algorithm can help to construct informative visualisation plots, even when many of the training points are incomplete.

Keywords: Incomplete Data, GTM, EM, Class Information.

1. Introduction

Data visualisation plays a key role in developing good models for large quantities of data and can aid analysts in identifying groups of data points which have similar characteristics. It is often the case that the visualisation plot can be more helpful when data is labelled with class information (using colour, for example), and in many datasets, this class information is readily available.

In many applications the input data is incomplete. Therefore it is important to use available values and class information and to reconstruct the missing values. For example, in the field of drug discovery, scientists use computer modelling to 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.

Tresp et al. ([8] and [7]) developed techniques for coping with missing data in the context of supervised learning. In this paper, we investigate ways of dealing with missing data in the context of unsupervised and "semi-supervised" learning. In particularly, we concentrate on the generative topographic mapping GTM [2], which is a non-linear probabilistic visualisation technique. The problem considered here is how to train the GTM model with incomplete data and reconstruct the missing values using additional class information. In this way the data, including the missing components, can be shown in a visualisation plot that is as "faithful" as possible.

Our algorithm can be described briefly as follows. A density model (GTM) of the data is learned in an unsupervised way from the incomplete training data set using an EM algorithm where sufficient statistics for the missing data are estimated in the E-step. For visualisation purposes, the missing values are filled in by computing the mean of their conditional distribution. The generic algorithm [6] can be improved by considering class label information associated with each point in the training data set. In this case, we can obtain the probability that a particular class c was responsible for generating the latent space centre. By using Bayes' theorem, we can compute the class-conditional priors for all latent space centres, which can be imposed on the latent space centres when we calculate the posterior probabilities (or 'responsibilities') of the latent space centres for the incomplete data points.

The generic algorithm, GTM with incomplete data, is detailed in section 2. Section 3 gives a detailed description of how we incorporate class information into the generic algorithm. We illustrate the algorithm in section 4 with a toy data set and a high-dimensional data set. Section 5 discusses the results.

2. Generative Topographic Mapping

2.1 The Generative Topographic Mapping

The generative topographic mapping (GTM) [1] is a non-linear 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 [4].

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. A latent data point $\mathbf{x}$ is mapped to $\mathbf{y} = \mathbf{W}\mathbf{\Phi}(\mathbf{x})$ in data space by a radial basis function network (see e.g. [3]) with weights $\mathbf{W}$ and a basis function matrix $\mathbf{\Phi}$. The goal of the model is to represent the distribution $P(\mathbf{t})$ in terms of K latent points $\mathbf{x}_j$ $(j = 1, 2, ..., K)$ and corresponding spherical Gaussian kernels centred on $\mathbf{y}(\mathbf{x}_j; \mathbf{W})$ [2]. The data density is defined by

$$P(\mathbf{t}|\mathbf{W}, \beta) = \frac{1}{K} \sum_{j=1}^{K} P(\mathbf{t}|\mathbf{x}_j, \mathbf{W}, \beta), \tag{1.1}$$

where $P(\mathbf{t}|\mathbf{x}_j, \mathbf{W}, \beta)$ is the density given by the jth component, i.e. $P(\mathbf{t}|\mathbf{x}_j, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{\frac{D}{2}} \exp\left\{-\frac{\beta}{2}||\mathbf{t} - \mathbf{y}(\mathbf{x}_j, \mathbf{W})||^2\right\}$. The weights $\mathbf{W}$ and the inverse variance β can be fitted by maximum likelihood with the EM algorithm, while the mixing coefficients are fixed to $1/K$. The M-step includes solving linear equations for $\mathbf{W}$ and a simple weighted variance formula for β^{-1}

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, where the responsibility r_{nj} is given by

$$r_{nj} = \frac{\left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x}_j; \mathbf{W}) - \mathbf{t}_n\|^2\right\}}{\sum_{j'=1}^{K} \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x}_{j'}; \mathbf{W}) - \mathbf{t}_n\|^2\right\}}. \tag{1.2}$$

2.2 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 m and o denote subvectors and submatrices of the parameters matching the missing and observed components of the data and each data vector can have different patterns of missing components. Binary indicator variables z_{nj} are introduced in the usual way 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 $z_{nj} = 0$. The EM algorithm treats both the indicator variables z_{nj} and the missing inputs $\mathbf{t}_n^m$ as hidden variables.

For the GTM, In the E-step, the expectation of z_{nj} is $E[z_{nj}|\mathbf{t}_n^o, \theta_j] = r_{nj}$ (equation (1.2)) measured only on $\mathbf{t}_n^o$ (the observed dimensions of $\mathbf{t}_n$).

Following [5], we introduce

$$\hat{\mathbf{t}}_{nj}^m = \mathrm{E}(\mathbf{t}_n^m | z_{nj} = 1, \mathbf{t}_n^o, \theta_j) = (\mathbf{y}_j^m)^{old} + \Sigma_j^{mo} \Sigma_j^{oo^{-1}} (\mathbf{t}_n^o - (\mathbf{y}_j^o)^{old}), \quad (1.3)$$

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, $(\mathbf{y}_j^m)^{old} = (\mathbf{W}_{old}\mathbf{\Phi}(\mathbf{x}_j))^m$ and $(\mathbf{y}_j^o)^{old} = (\mathbf{W}_{old}\mathbf{\Phi}(\mathbf{x}_j))^o$. As the covariance matrix is isotropic, $\Sigma_j = \beta^{-1}\mathrm{I}$, and the covariance of missing and observed values Σ_j^{mo} is equal to 0. So we have:

$$\hat{\mathbf{t}}_{nj}^m = (\mathbf{y}_j^m)^{old}. \quad (1.4)$$

The missing values are filled in using the posterior means,

$$E[\mathbf{t}_n^m | \mathbf{t}_n^o, \theta_j] = \sum_{j=1}^{K} r_{nj} \hat{\mathbf{t}}_{nj}^m. \quad (1.5)$$

In the M-step, the new weights are updated to $\mathbf{W}_{new}$ as described in [2] for complete training data. The variance is updated by

$$\beta^{-1} = \frac{1}{\mathrm{ND}} \sum_{n=1}^{N} \sum_{j=1}^{K} r_{nj} \left(\|\mathbf{t}_n^o - \mathbf{y}_j^o\|^2 + \mathrm{E}[z_{nj} \|\mathbf{t}_n^m - \mathbf{y}_j^m\|^2] \right), \quad (1.6)$$

where

$$\begin{aligned} \mathrm{E}[z_{nj}\|\mathbf{t}_n^m - \mathbf{y}_j^m\|^2] &= M(\beta^{-1})^{old} + (\hat{\mathbf{t}}_{nj}^m)^T(\hat{\mathbf{t}}_{nj}^m) - 2(\hat{\mathbf{t}}_{nj}^m)^T \mathbf{y}_j^m \\ &\quad + (\mathbf{y}_j^m)^T \mathbf{y}_j^m, \end{aligned} \quad (1.7)$$

M is the number of missing values in $\mathbf{t}_n$; $\mathbf{y}_j^o$ and $\mathbf{y}_j^m$ is computed using $(\mathbf{W}_{new}\mathbf{\Phi}(\mathbf{x}_j))^o$ and $(\mathbf{W}_{new}\mathbf{\Phi}(\mathbf{x}_j))^m$. A more detailed derivation can be found in [6].

3. Learning with Class-Conditional Information

When visualising a set of data points $T = \{\mathbf{t}_1, ..., \mathbf{t}_n\}$ from a set of class-labelled points $T_c = \{(\mathbf{t}_1, c_1), (\mathbf{t}_2, c_2), ..., (\mathbf{t}_N, c_N)\}$, with class labels c_n from a set $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_I\}$, one can use the class information as a clue for reasoning about missing values in the corrupted data points $\mathbf{t}_n$. Given a corrupted point $\mathbf{t}_n$, instead of computing the responsibilities $r_{nj} = P(\mathbf{x}_j | \mathbf{t}_n^o)$, we determine the *class-conditional responsibilities*

$$r_{njc} = P(\mathbf{x}_j | \mathbf{t}_n^o, c_n) = \frac{P(\mathbf{x}_j, c_n | \mathbf{t}_n^o)}{\sum_{k=1}^{K} P(\mathbf{x}_k, c_n | \mathbf{t}_n^o)}. \quad (1.8)$$

By the (standard) assumption of conditional independence of observed variables, given the hidden ones, we have $P(\mathbf{t}_n^o, c_n|\mathbf{x}_j) = P(\mathbf{t}_n^o|\mathbf{x}_j)P(c_n|\mathbf{x}_j)$. Using a flat prior on the latent space centres $P(\mathbf{x}_k) = 1/K$, $k = 1, 2, ..., K$,

$$P(\mathbf{x}_j, c_n|\mathbf{t}_n^o) = \frac{P(\mathbf{t}_n^o, c_n|\mathbf{x}_j)P(\mathbf{x}_j)}{\sum_{k=1}^{K}\sum_{C_i \in \mathcal{C}} P(\mathbf{t}_n^o, C_i|\mathbf{x}_k)P(\mathbf{x}_k)} = r_{nj}\ P(c_n|\mathbf{x}_j). \quad (1.9)$$

The distribution of class labels, conditioned on the latent space centres $\mathbf{x}_j$, is computed by determining the "mass" of uncorrupted training points "explained" by $\mathbf{x}_j$ and belonging to a class $C_i \in \mathcal{C}$,

$$P(C_i|\mathbf{x}_j) = \frac{\sum_{\mathbf{t}_n \in T_{comp}; c_n = C_i} P(\mathbf{t}_n|\mathbf{x}_j)}{\sum_{\mathbf{t}_n \in T_{comp}} P(\mathbf{t}_n|\mathbf{x}_j)}, \quad (1.10)$$

where T_{comp} is a collection of uncorrupted training points. This means we are assuming that the mechanism for missing data does not depend on the class.

After some calculation, (1.8) can be rewritten as

$$r_{njc} = \frac{P(\mathbf{t}_n^o|\mathbf{x}_j)\ P(c_n|\mathbf{x}_j)}{\sum_{k=1}^{K} P(\mathbf{t}_n^o|\mathbf{x}_k)\ P(c_n|\mathbf{x}_k)} = \frac{P(\mathbf{t}_n^o|\mathbf{x}_j)\ P(\mathbf{x}_j|c_n)}{\sum_{k=1}^{K} P(\mathbf{t}_n^o|\mathbf{x}_k)\ P(\mathbf{x}_k|c_n)}, \quad (1.11)$$

where for $C_i \in \mathcal{C}$,

$$P(\mathbf{x}_j|C_i) = \frac{P(C_i|\mathbf{x}_j)}{\sum_{k=1}^{K} P(C_i|\mathbf{x}_k)}. \quad (1.12)$$

It follows from (1.11), that unlike the original latent centres' responsibilities r_{nj}, where a flat prior $P(\mathbf{x}_k) = 1/K$ is imposed, the class-conditional responsibilities, r_{njc}, are calculated using a *class-conditional prior on latent space*, $P(\mathbf{x}_j|C_i)$, $C_i \in \mathcal{C}$. When faced with corrupted data points, such class-conditional priors help us to eliminate (at least to some degree) multi-modalities in posterior over the latent space centres. This may be no longer a true EM algorithm. However, in our experiments there was no occasion on which the likelihood decreased, even with large numbers of missing values.

4. Experimental Results

In this section we compare three training algorithms for the GTM with missing data: **(I)** generic EM for missing data using class information as described in Section 3; **(II)** standard EM on the data first completed by class-conditional mean imputation. In this case, each missing component of a data point $\mathbf{t}_n$ from class C_i is filled by simply calculating the average of all observed values of the same component (dimension) in the points from the class C_i; **(III)** generic EM for missing data (i.e. with on-line estimation of missing values but not using class information).

4.1 Toy data

We first consider a toy data set, which consists of 80 training points lying on the curve $t_2 = \sin(t_1)$. The coordinate t_1 was generated uniformly in the interval $[-\pi/2, 7\pi/2]$, and spherical Gaussian noise with standard deviation 0.1 was added to the t_2 coordinate. We have defined 4 classes as shown in Figure 1.1. We also sampled a (complete) test data set of 200 data points from the same distribution.

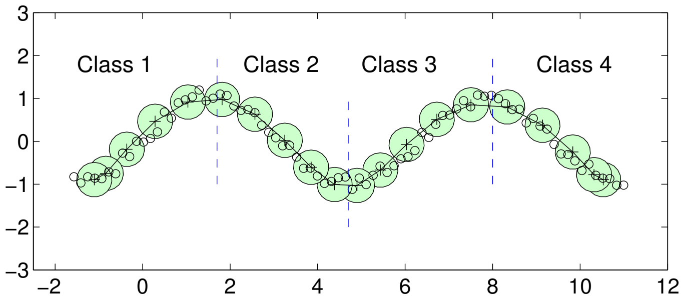


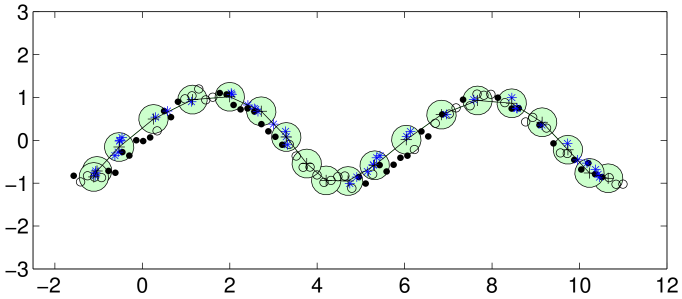
Figure 1.1. GTM at iteration 20 trained on complete data set

Some values of t_1 were removed at random from points in all classes. A straightforward application of the generic algorithm (III) gave unsatisfactory results because of multi-modalities in the responsibilities (1.2). For example, if $t_2 = 0$, t_1 could be $0, \pi, 2\pi$ or 3π with equal probability. The network was trained using Algorithms I and II. The proportion of missing values was varied from 0 to 70 percent. For the different percentage of missing values, the experiments were carried out 10 times deleting different values.

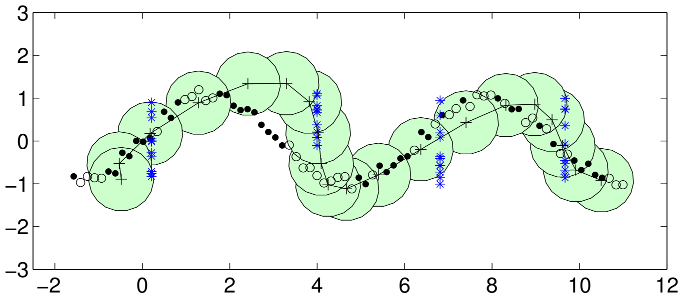
Plots (a) and (b) in Figure 1.2 display results using Algorithms I and II. It suggests that the EM algorithm combining the class-conditional information is highly successful, while the missing values estimated using class-conditional MI lie on straight lines. Note how the greater uncertainty in estimating missing data for Algorithm II leads to much larger variance in the trained GTM. Plot (c) displays the test set likelihood. It shows that the network can perform better using Algorithm I than using Algorithm II and still performs well with even up to 60% missing values in the training data set.

4.2 Oil Flow Data

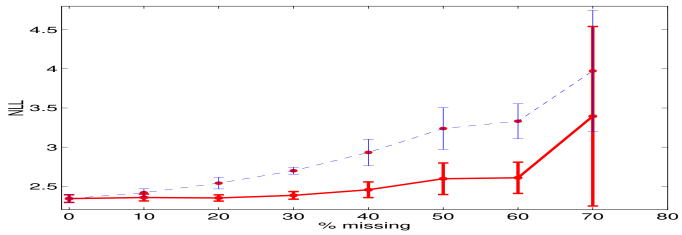
In this experiment, we chose an oil pipeline flow data set [1], which is a more realistic test case with 12 measured variables. There are three flow



(a) EM with class information (Algorithm I)



(b) Data completed using class-conditional mean imputation (Algorithm II)



(c) Testing error in the toy problem

Figure 1.2. Toy problem: illustrations of GTMs at iteration 20 ((a)–(b)) trained on data with 50% of t_1 values missing. Complete points are plotted as circles. Stars represent the estimated missing values, while the dots show the original values. The centres of the GMM are plotted as plus signs, and are joined by a line according to their ordering in the (one-dimensional) latent space (K = 20). The discs surrounding each plus sign represent two standard deviations' width of the noise model. (c) The average test set negative log likelihood (NLL) and standard deviation error bars for 10 repetitions of the training process. Algorithm I and II are represented by the thick solid and dashed lines, respectively.

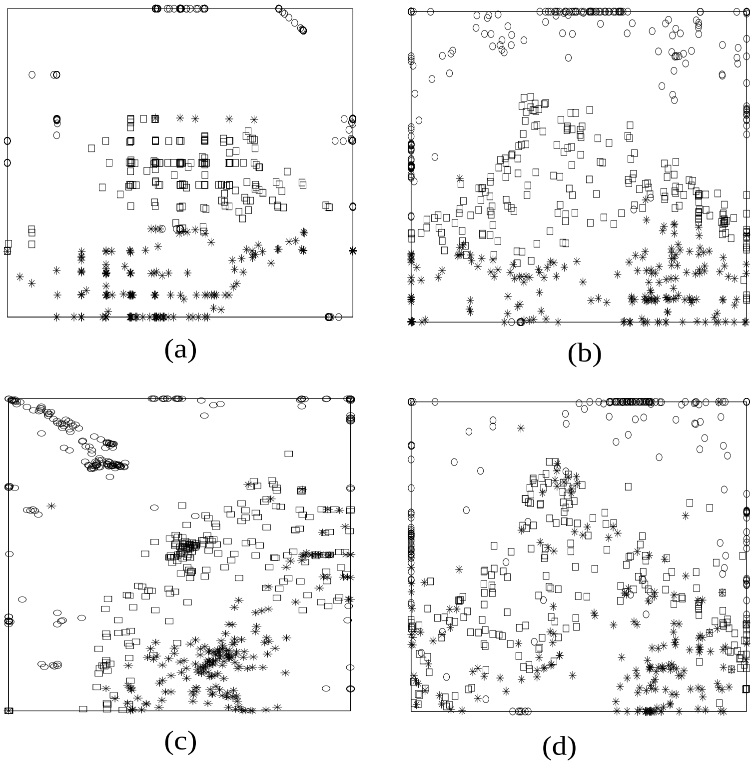


Figure 1.3. Oil flow data: (a) training on the complete data set. The remaining plots show results for models trained on incomplete data. (b) EM algorithm combining the class-conditional information (Algorithm I); (c) standard EM using conditional mean imputation (Algorithm II); (d) generic EM without class information (Algorithm III). The classes homogeneous, annular and laminar are represented by square, star and circle signs respectively.

classes in this data set: homogeneous, annular and laminar. A GTM with a two-dimensional latent space was used to model and visualise the data. In the training set, 50% of the data points in each class are incomplete, with between 6 and 9 values removed.

Figure 1.3 displays the results. It suggests that Algorithm I can be an improvement over the two other algorithms for missing data. Plot (b) shows better separation of classes and matches better to the result obtained from the complete data set (plot (a)). After using class-conditional MI, some strongly overlapped clusters appear in plot (c) since the same means are substituted for missing values of the same class. As for plot (d), which was obtained just by the generic algorithm, the homogeneous and annular classes are not separated well as we did not use the class-conditional prior knowledge in the training process.

5. Discussion

In this paper, we have shown how incomplete data and class information can both be included in the GTM training process. The new algorithm is preferable to the simple strategy of just filling in the missing values with conditional means. The problem of multi-modalities in the GTM responsibilities for missing data can be overcome (at least to some degree) by means of the class-conditional prior and this significantly improves both the visualisation plot and the fit of the model.

Acknowledgments

This research has been funded by Pfizer Central Research and BBSRC grant 92/BIO12093. The experiments were carried out with the NETLAB neural network toolbox, available from
`http://www.ncrg.aston.ac.uk/netlab`. The authors are most grateful to the anonymous reviewers for their insightful and helpful comments.

References

[1] 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.

[2] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: The Generative Topographic Mapping. *Neural Computation*, 10(1):215–235, 1998.

[3] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, New York, N.Y., 1995.

[4] 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.

[5] Z. Ghahramani and M. I. Jordan. Learning from incomplete data. Technical report, AI Laboratory, MIT, 1994.

[6] Y. Sun, P. Tiňo, and I. Nabney. GTM-based Data Visualisation with Incomplete Data. Technical Report NCRG/2001/013, NCRG, Aston University, 2001.

[7] V. Tresp, S. Ahmad, and R. Neuneier. Training neural networks with deficient data. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 128–135. Morgan Kaufmann Publishers, Inc., 1994.

[8] V. Tresp, R. Neuneier, and S. Ahmad. Efficient methods for dealing with missing data in supervised learning. In D. S. Touretzky G. Tesauro and T. K. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, 1995.