GTM Through Time

Christopher M. Bishop[†], Geoffrey E. Hinton[*] and Iain G. D. Strachan[†]

[†] Neural Computing Research Group, Aston University, U.K.
[*] Dept. of Computer Science, University of Toronto, Canada.

Abstract

The standard GTM (generative topographic mapping) algorithm assumes that the data on which it is trained consists of independent, identically distributed (i.i.d.) vectors. For time series, however, the i.i.d. assumption is a poor approximation. In this paper we show how the GTM algorithm can be extended to model time series by incorporating it as the emission density in a hidden Markov model. Since GTM has discrete hidden states we are able to find a tractable EM algorithm, based on the forward-backward algorithm, to train the model. We illustrate the performance of GTM through time using flight recorder data from a helicopter.

1 Introduction

Latent variable models provide a representation for the distribution of data in a multi-dimensional space in terms of a reduced number of latent, or hidden, variables (Everitt, 1984). A well-known example of a latent variable algorithm is factor analysis which is based on a linear transformation between latent space and data space. The technique of principal component analysis can also be understood within the same framework and again involves a linear transformation from the hidden variables to the data variables. Recently there has been considerable interest in non-linear latent variable models in applications such as pattern recognition and data visualization. In particular, the *Generative Topographic Mapping* algorithm (Bishop *et al.*, 1996a; Bishop *et al.*, 1996b) has been introduced as a non-linear latent variable model which provides a principled alternative to the self-organizing map (SOM) algorithm of Kohonen (1982). Unlike the SOM, the GTM model defines a genuine probability density and thereby overcomes many of the limitations of the SOM.

In common with many models for density estimation, the GTM algorithm treats the data as independent, identically distributed (i.i.d.). While this is a valid assumption for some applications, there are many situations in which we need a more general framework. In particular, the i.i.d. assumption is clearly inappropriate for time series data, for which we typically expect data values at neighbouring time steps to be highly correlated and hence far from independent. In this paper we show how the GTM algorithm can be extended to deal with time series. We illustrate the technique using flight recorder data taken from a helicopter operating in a variety of different flight regimes. Finally, we discuss some extensions of the algorithm.

2 The Generative Topographic Mapping

We begin by reviewing the GTM algorithm for the standard case of i.i.d. data. The goal of GTM is to model the probability distribution of data living in a d-dimensional space in terms of L latent variables where $L < d$ and

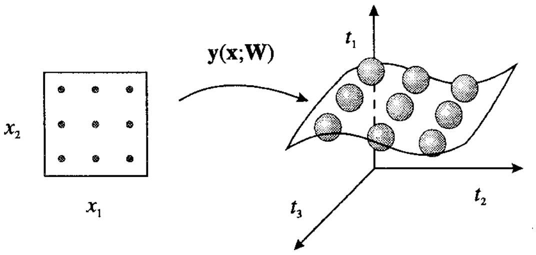

Figure 1: The non-linear mapping $\mathbf{y}(\mathbf{x};\mathbf{W})$ from the L-dimensional latent space $\mathbf{x}$ to the D-dimensional data space $\mathbf{t}$ defines an L-dimensional non-Euclidean manifold. A regular grid of points in latent space will then be mapped to an array of points on the manifold, with each point forming the centre of one of the Gaussian components in the density model.

where the transformation from latent variables to data variables can be non-linear. In applications involving data visualization it is convenient to consider a two-dimensional latent space, and we shall assume that $L = 2$ throughout this paper.

We denote the coordinates of the data space by $\mathbf{t} = (t_1, \ldots, t_D)^{\mathrm{T}}$ and those of the latent space by $\mathbf{x} = (x_1, \ldots, x_L)^{\mathrm{T}}$. The mapping from latent space to data space takes the form

$$\mathbf{y}(\mathbf{x};\mathbf{W}) = \mathbf{W}\boldsymbol{\phi}(\mathbf{x}) \qquad (1)$$

where $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_M)^{\mathrm{T}}$ represents a set of M fixed non-linear basis functions, and $\mathbf{W}$ is a $D \times M$ matrix of parameters. This mapping defines a non-Euclidean manifold embedded in data space, as illustrated in Figure 1. The form of the mapping (1) is chosen to simplify the training algorithm as discussed below. Note that (1) can approximate any continuous mapping to arbitrary accuracy provided we have sufficiently many basis functions $\phi_j(\mathbf{x})$ of an appropriate form.

A latent variable density model is defined by specifying a *prior* distribution $p(\mathbf{x})$ over latent space, together with a conditional distribution $p(\mathbf{t}|\mathbf{x})$ in data space, conditioned on the latent variables. The resulting density model is then obtained from the sum rule of probability by the convolution of these densities in the form

$$p(\mathbf{t}) = \int p(\mathbf{t}|\mathbf{x})p(\mathbf{x})\,d\mathbf{x}. \qquad (2)$$

For non-linear mappings this integral will in general be analytically intractable. In the GTM algorithm we therefore choose a specific form for the prior consisting of a superposition of delta functions given by

$$p(\mathbf{x}) = \frac{1}{K}\sum_{l=1}^{K}\delta(\mathbf{x} - \mathbf{x}_l) \qquad (3)$$

where $\{\mathbf{x}_l\}$ is a set of points on a regular grid in latent space (analogous to the 'feature-space' nodes in the SOM). This choice allows the integral in (2) to be evaluated analytically. It also has important implications for the extension to time-varying data discussed in Section 4.

From (2) and (3) we obtain a density model given by

$$p(\mathbf{t}) = \frac{1}{K}\sum_{l=1}^{K}p(\mathbf{t}|\mathbf{x}_l). \qquad (4)$$

We now choose the conditional density $p(\mathbf{t}|\mathbf{x})$ to be a radially symmetric Gaussian of the form

$$p(\mathbf{t}|\mathbf{x}) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|\mathbf{y} - \mathbf{t}\|^2\right\} \qquad (5)$$

where $\mathbf{y} = \mathbf{y}(\mathbf{x};\mathbf{W})$. The density (4) then represents a Gaussian mixture model in which the centres of the Gaussians are constrained to lie on a non-Euclidean manifold embedded in data space. Each latent space point is mapped to a corresponding point $\mathbf{y}(\mathbf{x}_l;\mathbf{W})$ lying on the manifold in data space which forms the centre of one of the components. Changes to the centres can only be made indirectly through changes in the parameters $\mathbf{W}$ describing the manifold. The model is also constrained in that the mixing proportions are fixed at $1/K$ (this is easily generalized if desired) and the Gaussian components have a common variance β^{-1}. For the standard GTM model trained on i.i.d. data, the parameter values can be determined using the EM algorithm.

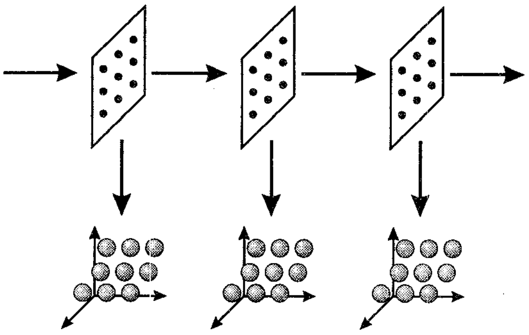

Figure 2: The temporal version of GTM consists of a hidden Markov model in which the hidden states are given by the latent points of the GTM model, and the emission probabilities are governed by the GTM mixture distribution. Note that the parameters of the GTM model, as well as the transition probabilities between states, are tied to common values across all time steps.

3 GTM Through Time

For data vectors $\mathbf{t}_n$ which take the form of a time series it is no longer appropriate to assume that the vectors are independent. Typically, vectors corresponding to nearby times will be highly correlated. Such effects can be captured using the hidden Markov model (HMM) formalism (Rabiner, 1989). Here we show how GTM can be extended within the HMM framework to represent temporal data.

The structure of the model is illustrated in Figure 2, in which the hidden states of the model at each time step are labelled by the index l corresponding to the latent points $\{\mathbf{x}_l\}$. We introduce a set of transition probabilities p_{ij} corresponding to the probability of making a transition to state j given that the current state is i. The emission density for the hidden Markov model is then given by the GTM density model (4). It should be noted that both the transition probabilities p_{ij} and the parameters $\mathbf{W}$ and β governing the GTM model are common to all time steps, so that the number of adaptive parameters in the model is independent of the length of the time series. We also allow separate prior probabilities π_l on each of the latent points at the first time step of the algorithm.

If we allow a fully connected matrix of independent transition probabilities connecting every state at time n to every state at time $n+1$, then the number of independent parameters would be prohibitively large. If we have, for example, 100 hidden states in the GTM model (a relatively small number) then we would have 10^4 independent transition probability parameters to be determined (slightly less in fact due to the constraint that probabilities must sum to one). This would require an excessive amount of training data.

Also it fails to capture any prior knowledge which we might possess about the nature of the transitions between different time steps. In many applications we expect different regions of the latent space to correspond to different regimes. We also expect smooth changes in latent space within a regime and relatively rare jumps to other regimes. An approximate way to capture this prior knowledge is to allow groups of transitions to be governed by a common parameter. We denote the kth group from state i by $\mathcal{G}_{ik}$ and we introduce indicator variables C_{ikj} which equal 1 if state j is in group $\mathcal{G}_{ik}$ and 0 otherwise. The transition probability from state i to a state in group k will be denoted η_{ik}, and these satisfy $\sum_k \eta_{ik} = 1$. The transition probability from state i to state j is then given by

$$p_{ij} = \sum_k \eta_{ik} C_{ikj} N_{ik}^{-1} \qquad (6)$$

where N_{ik} denotes the number of states in group $\mathcal{G}_{ik}$.

4 EM Algorithm

The model is trained using a set of N data vectors $\mathbf{t}_1, \ldots, \mathbf{t}_N$, in which the parameters $\mathbf{W}$ and β, as well as the transition probabilities, are determined by maximum likelihood. To find the correct likelihood function we note that the model represents a generative distribution for time series data as follows. At the first time step we select a latent point i with probability π_i and then generate the first data vector $\mathbf{t}_1$ by sampling from the corresponding Gaussian component $p(\mathbf{t}|\mathbf{x}_i)$ of the GTM model. Next we make a transition to a new state j with probability p_{ij} and again generate a data point from the corresponding component $p(\mathbf{t}|\mathbf{x}_j)$. From this we see that the likelihood function for a given observed sequence of vectors $\mathbf{t}_1, \ldots, \mathbf{t}_N$ can be written

$$\sum_{i_1} \cdots \sum_{i_N} \pi_{i_1} p(\mathbf{t}_1|\mathbf{x}_{i_1}) p_{i_1 i_2} \cdots p(\mathbf{t}_N|\mathbf{x}_{i_N}) \qquad (7)$$

where i_n denotes the state at step n. The summations correspond to a sum over all possible trajectories through the hidden states of the model. At first sight it would therefore appear that the evaluation and optimization of (7) would be an extremely complex undertaking since the number of paths through the hidden states grows exponentially with N. However, because of the discrete nature of the hidden states, we can obtain an efficient algorithm for training this model.

We can regard the identity of the component responsible for generating each data point as a missing variable, and use the EM (expectation-maximization) algorithm to maximize the likelihood (Dempster *et al.*, 1977; Bishop, 1995). In the context of hidden Markov models this is generally known as the Baum-Welch algorithm. To obtain the EM algorithm for this model we first introduce a set of binary indicator variables z_{ni} which denote the state i of the system at step n. We shall regard the z_{ni} as missing variables. If the z_{ni} were given, then the complete-data likelihood would take the form

$$L_c = \prod_{n=1}^{N-1} \prod_{i_n} \{\pi_{i_1} p(\mathbf{t}_1|\mathbf{x}_{i_1})\}^{z_{1i_1}} \{p_{i_n,i_{n+1}} p(\mathbf{t}_{n+1}|\mathbf{x}_{i_{n+1}})\}^{z_{ni_n} z_{ni_{n+1}}} (8)$$

The algorithm involves first making an initial guess for the parameters $\mathbf{W}$, β and η_{ik}. We next take the expectation of the logarithm of the complete-data log likelihood function (8) with respect to the posterior distribution of the z_{ni} (evaluated using the current values of the parameters), and use $\langle z_{ni} z_{nj}\rangle = \xi_n(i,j)$, where $\xi_n(i,j)$ denotes the joint posterior probability of being in state i at time n and state j at time $n+1$, to give

$$\langle \ln L_c \rangle = \sum_{n=1}^{N-1} \sum_{i_n} \xi_n(i_n, i_{n+1}) \ln\{p_{i_n,i_{n+1}} p(\mathbf{t}_n|\mathbf{x}_{i_n})\}. \qquad (9)$$

The posterior probabilities $\xi_n(i,j)$ are obtained in the E-step using the standard forward-backward algorithm (Rabiner, 1989). Maximizing (9) with respect to the η_{ik}, and using a Lagrange multiplier to enforce the constraint $\sum_k \eta_{ik} = 1$, we obtain

$$\eta_{ik} = \frac{\sum_n \sum_{j \in \mathcal{G}_k} \xi_n(i,j)}{\sum_k \sum_n \sum_{j \in \mathcal{G}_k} \xi_n(i,j)}. \qquad (10)$$

Similarly we can maximize (9) with respect to $\mathbf{W}$ to obtain the M-step equation

$$\mathbf{\Phi}^{\mathrm{T}} \mathbf{G}_{\mathrm{old}} \mathbf{\Phi} \mathbf{W}_{\mathrm{new}}^{\mathrm{T}} = \mathbf{\Phi}^{\mathrm{T}} \mathbf{R}_{\mathrm{old}} \mathbf{T} \qquad (11)$$

where $R_{in} = \sum_j \xi_n(i,j)$ denotes the posterior probability of state i at step n, $\mathbf{\Phi}$ is a $K \times M$ matrix with elements $\Phi_{ij} = \phi_j(\mathbf{x}_i)$, $\mathbf{T}$ is a $N \times D$ matrix with elements t_{nk}, $\mathbf{R}$ is a $K \times N$ matrix with elements R_{in}, and $\mathbf{G}$ is a $K \times K$ diagonal matrix with elements

$$G_{ii} = \sum_{n=1}^{N} R_{in}(\mathbf{W}, \beta). \qquad (12)$$

We can solve (11) for $\mathbf{W}_{\mathrm{new}}$ using standard matrix inversion techniques, based on singular value decomposition to allow for possible ill-conditioning. Note that the matrix $\mathbf{\Phi}$ is constant throughout the algorithm, and so need only be evaluated once at the start.

Finally, maximizing (9) with respect to β gives

$$\frac{1}{\beta_{\text{new}}} = \frac{1}{ND}\sum_{n=1}^{N}\sum_{i=1}^{K} R_{in}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \\ \|\mathbf{y}(\mathbf{x}_i; \mathbf{W}_{\text{new}}) - \mathbf{t}_n\|^2 . \quad (13)$$

After a complete M-step, the new parameter values are used in the next E-step to re-evaluate the posterior probabilities, and so on to convergence.

5 Helicopter Flight Data

We now present simulation results using the temporal GTM model applied to real data derived from helicopter test flights. The motivation behind this application is in determining the accumulated stress on the helicopter airframe. Different flight modes, and transitions between flight modes, cause different levels of stress, and at present maintenance intervals are determined using an assumed usage spectrum. The ultimate goal in this application would be to segment each flight into its distinct regimes, together with the transitions between those regimes, and hence evaluate the overall integrated stress.

The data used in this simulation was gathered from the flight recorder over four test flights, and consists of 9 variables (sampled every two seconds) measuring quantities such as acceleration, rate of change of heading, speed, altitude and engine torque. A sample of the data is shown in Figure 3.

We consider a GTM model having a 15×15 grid of states in latent space. For each latent state i, the transition probabilities to states at the next time step are collected into 10 separate groups, in which 9 of the groups correspond to those states j which are within a distance of ± 1 units in latent space from state i, while the 10th group consists of all remaining stages j. We expect that, in the trained model, different regions of the latent space will correspond to different flight regimes. The first group of transition probabilities then allows evolution of the data vector within a given flight regime to be modelled, while transitions to distant flight regimes can be represented by the second group. Note that this does not capture all of the likely behaviour of the time series. For instance, if transitions between two particular regimes are much more frequent than between some other pair of regimes, this cannot be represented in the model just described.

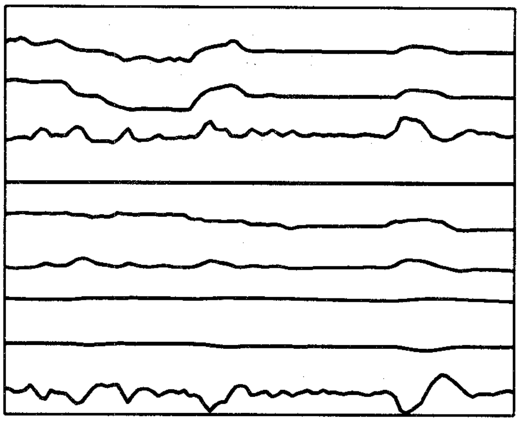

Figure 3: Sample of the helicopter data.

Figure 4 shows the posterior probability distribution in latent space for a trained temporal GTM model, in which the posterior probabilities for a given temporal sequence have been evaluated using the forward-backward algorithm as described earlier. Currently, we are

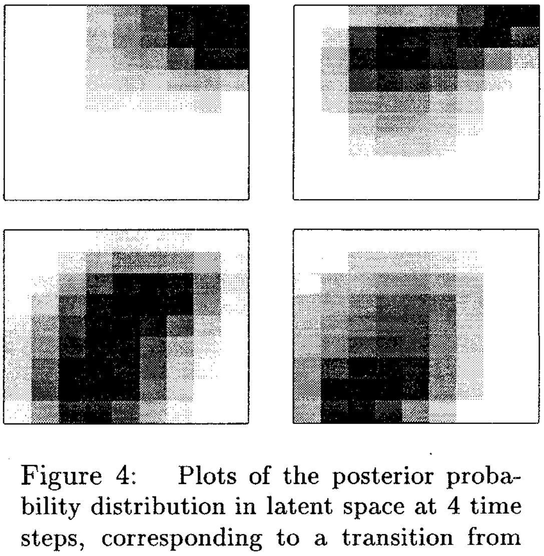

Figure 4: Plots of the posterior probability distribution in latent space at 4 time steps, corresponding to a transition from one flight regime to another.

exploring models with a more complex transition probability structure in order to discover the relative probabilities of different transitions.

6 Discussion

We have presented a latent variable model which can capture temporal dynamics and which also permits a non-linear transformation from latent to data space.

Since the algorithm is based on GTM, rather than the self-organizing map, it defines a true density model, which brings a number of important advantages. For example, real data sets frequently suffer from missing values. Provided the missing values are 'missing at random' then we can still find the optimal maximum likelihood parameter values by extending the EM algorithm to deal with the missing values.

Similarly, we can consider a *mixture* of GTM models at each time step and still retain the tractable EM algorithm. In this case there are two classes of transition probability, one governing transitions within each of the GTM sub-models and one governing transitions between sub-models.

Finally, we can use the model to perform novelty detection by finding sequences of states which have a small probability under the trained model. This highlights the difference between the i.i.d. and temporal models, since, if the data passed through a set of familiar states but in an unfamiliar order, the temporal GTM model would recognise the sequence as novel, whereas an i.i.d. approach would not.

Acknowledgements

This work was supported by EPSRC grant GR/K51792: *Validation and Verification of Neural Network Systems.* We would like to thank Markus Svensén and Christopher Williams for useful discussions.

References

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press.

Bishop, C. M., M. Svensén, and C. K. I. Williams (1996a). GTM: a principled alternative to the Self-Organizing Map. In *Proceedings 1996 International Conference on Artificial Neural Networks, ICANN'96*, pp. 165–170. Bochum, Germany: Springer-Verlag.

Bishop, C. M., M. Svensén, and C. K. I. Williams (1996b). GTM: the generative topographic mapping. Accepted for publication in *Neural Computation.* Available as NCRG/96/015 from `http://www.ncrg.aston.ac.uk/`.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B* **39** (1), 1–38.

Everitt, B. S. (1984). *An Introduction to Latent Variable Models.* London: Chapman and Hall.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics* **43**, 59–69.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* **77** (2), 257–285.