Developments of the Generative Topographic Mapping

Christopher M. Bishop
Microsoft Research
St. George House, 1 Guildhall Street, Cambridge CB2 3NH
cmbishop@microsoft.com

Markus Svensén
Neural Computing Research Group
Aston University, Birmingham, B4 7ET, U.K.
svensen@cns.mpg.de*

Christopher K. I. Williams
Neural Computing Research Group
Aston University, Birmingham, B4 7ET, U.K.
C.K.I.Williams@aston.ac.uk

April 30, 1998

Abstract

The Generative Topographic Mapping (GTM) model was introduced by Bishop *et al.* (1998) as a probabilistic re-formulation of the self-organizing map (SOM). It offers a number of advantages compared with the standard SOM, and has already been used in a variety of applications. In this paper we report on several extensions of the GTM, including an incremental version of the EM algorithm for estimating the model parameters, the use of local subspace models, extensions to mixed discrete and continuous data, semi-linear models which permit the use of high-dimensional manifolds whilst avoiding computational intractability, Bayesian inference applied to hyper-parameters, and an alternative framework for the GTM based on Gaussian processes. All of these developments directly exploit the probabilistic structure of the GTM, thereby allowing the underlying modelling assumptions to be made explicit. They also highlight the advantages of adopting a consistent probabilistic framework for the formulation of pattern recognition algorithms.

1 Introduction

Probability theory provides a powerful, consistent framework for dealing quantitatively with uncertainty (Cox, 1946). It is therefore ideally suited as a theoretical foundation for pattern

*Current address: Max-Plank-Institute of Cognitive Neuroscience, Inselstraße 22–26, D-04103, Leipzig, Germany.

recognition. Recently, the self-organizing map (SOM) of Kohonen (1982) was re-formulated within a probabilistic setting (Bishop *et al.*, 1998) to give the GTM (Generative Topographic Mapping). In going to a probabilistic formulation, several limitations of the SOM were overcome, including the absence of a cost function and the lack of a convergence proof.

A further advantage of the probabilistic formulation of the GTM is that extensions to the basic model can be formulated in a principled manner in which the corresponding modelling assumptions are made explicit. In this paper we present several extensions of the GTM, all of which build on its probabilistic formulation. We first show, in Section 2, how a generalized form of EM algorithm can be used to derive an incremental version in which data points are presented one at a time, while preserving the convergence guarantees of the batch version. Next we show in Section 3 how the Gaussian components of the GTM can be generalized from an isotropic distribution to one which reflects the local subspace properties of the underlying manifold. Then in Section 4 we show how the GTM can be extended to allow for discrete as well as continuous data variables. A generalization of the GTM which permits the use of high-dimensional manifolds without running into computational intractability is described in Section 5. Next, in Section 6 we provide a Bayesian treatment of the hyper-parameters in the GTM. Finally, in Section 7 we demonstrate the use of Gaussian processes in place of standard regression models to define the non-linear manifold.

We begin with a brief, self-contained, review of the GTM.

1.1 The Generative Topographic Mapping

The Generative Topographic Mapping is a probability density model which describes the distribution of data in a space of several dimensions in terms of a smaller number of latent (or hidden) variables. By using a discrete grid of points in latent space, analogous to the nodes of the SOM, it is able to use a non-linear relationship between the latent space and the data space while remaining tractable. A detailed derivation of the GTM can be found in Bishop *et al.* (1998). Here we simply describe the resulting density model and summarize the parameter estimation (or training) procedure.

Our description of the GTM starts by defining a q-dimensional latent space, with coordinates $\mathbf{u} = (u_1, \ldots, u_q)$, as shown schematically on the left-hand side of Figure 1. For the purposes of this paper we shall be primarily interested in $q = 1$ or $q = 2$. Within the latent space we introduce a regular array of nodes, labelled by the index $i = 1, \ldots, K$. These are analogous to the nodes of the SOM. Next we introduce a set of M fixed non-linear basis functions $\boldsymbol{\phi}(\mathbf{u}) = \{\phi_j(\mathbf{u})\}$, where $j = 1, \ldots, M$, which form a non-orthogonal basis set. The $\{\phi_j\}$ might consist, for example, of a regular array of Gaussian or sigmoidal functions. Using these basis functions we define a non-linear transformation from the latent space to the data space given by a linear combination of the basis functions so that each point $\mathbf{u}$ in latent space is mapped to a corresponding point $\mathbf{y}$ in the D-dimensional data space given by

$$\mathbf{y} = \mathbf{W}\boldsymbol{\phi}(\mathbf{u}) \tag{1}$$

where $\mathbf{W}$ is a $D \times M$ matrix of weight parameters.

If we denote the node locations in latent space by $\mathbf{u}_i$, then (1) defines a corresponding set of

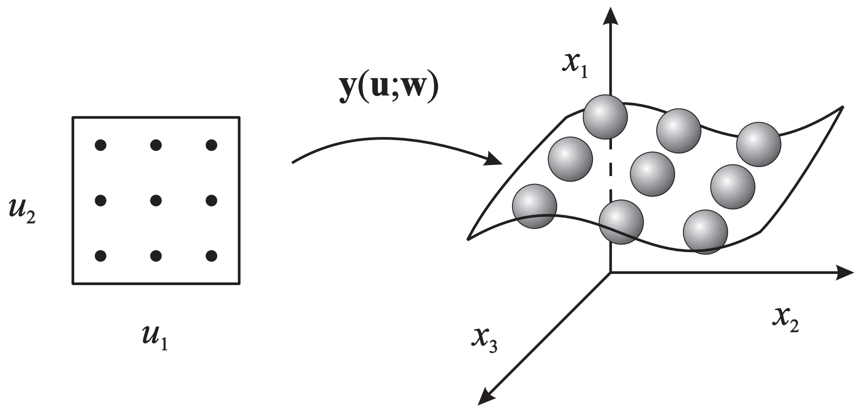


Figure 1: In order to formulate a latent variable model which is similar in spirit to the SOM, we consider a prior distribution $p(\mathbf{u})$ consisting of a superposition of delta functions located at the nodes of a regular grid in latent space. Each node $\mathbf{u}_i$ is mapped to a corresponding point $\mathbf{m}_i = \mathbf{y}(\mathbf{u}_i; \mathbf{W})$ in data space, and forms the centre of a corresponding Gaussian distribution.

'reference vectors' given by

$$\mathbf{m}_i = \mathbf{W}\boldsymbol{\phi}(\mathbf{u}_i). \tag{2}$$

Each of the reference vectors then forms the centre of an isotropic Gaussian distribution in data space, whose inverse variance we denote by β, so that

$$p(\mathbf{x}|i) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|\mathbf{m}_i - \mathbf{x}\|^2\right\}. \tag{3}$$

Finally, the probability density function for the GTM model is obtained by summing over all of the Gaussian components, to give

$$p(\mathbf{x}|\mathbf{W},\beta) = \sum_{i=1}^{K} P(i)p(\mathbf{x}|i) = \sum_{i=1}^{K} \frac{1}{K}\left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|\mathbf{m}_i - \mathbf{x}\|^2\right\} \tag{4}$$

where K is the total number of components (equal to the number of grid points in latent space), and we have taken the prior probabilities of each of the components to be constant and equal to $1/K$. To summarize, we can regard the GTM model as a constrained mixture of Gaussians, as illustrated schematically in Figure 1, in which the Gaussian components are isotropic with an inverse variance β and have centres given by (2). The GTM is an example of a latent variable model, in which the probability distribution of the observed data variables $\mathbf{x}$ is expressed in terms of an integration over the distribution of a set of latent, or hidden, variables $\mathbf{u}$ whose values are unobserved. The regular grid of points in latent space corresponds to a particular choice of latent space distribution for which the integration is tractable. Since the transformation from latent space to data space is non-linear, the GTM is representing the distribution of data in terms of a q-dimensional non-Euclidean manifold in data space. The Gaussian distribution (3) represents a noise model and allows for the fact that the data will not be confined precisely to such a q-dimensional manifold.

The adaptive parameters of the model are $\mathbf{W}$ and β. Since the GTM represents a constrained mixture model, the centres of the Gaussian components cannot be adapted to the data independently, but instead are adjusted indirectly through changes to the weight matrix $\mathbf{W}$.

We denote the data space variables by $\mathbf{x} = x_1, \ldots, x_D$, and we shall assume that the data set has been normalized to zero mean (equivalently we can include a constant basis function $\phi_0(\mathbf{u}) = 1$

in the mapping (1)). Since the GTM represents a parametric probability density model, it can be fitted to a data set $\{\mathbf{x}_n\}$, where $n = 1, \ldots, N$, by maximum likelihood. The log likelihood function is given by

$$\mathcal{L}(\mathbf{W}, \beta) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n | \mathbf{W}, \beta) \tag{5}$$

where $p(\mathbf{x}|\mathbf{W}, \beta)$ is given by (4), and we have assumed independent, identically distributed data. We can maximize this log likelihood function by finding expressions for its derivatives and using these in a standard non-linear optimization algorithm such as conjugate gradients.

Alternatively, we can exploit the latent-variable structure of the model and use the expectation-maximization (EM) algorithm (Dempster *et al.*, 1977; Bishop, 1995). In the E-step, we use the current values of the parameters $\mathbf{W}$ and β to evaluate the posterior probability, or *responsibility*, which each component i takes for every data point $\mathbf{x}_n$, which, using Bayes' theorem, is given by

$$R_{ni} \equiv p(i|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|i)}{\sum_j p(\mathbf{x}_n|j)} \tag{6}$$

in which the prior probabilities $P(i) = 1/K$ have cancelled between numerator and denominator. Using (3) we can rewrite this in the form

$$R_{ni} = \frac{\exp\left\{-\frac{\beta}{2}\|\mathbf{m}_i - \mathbf{x}_n\|^2\right\}}{\sum_j \exp\left\{-\frac{\beta}{2}\|\mathbf{m}_j - \mathbf{x}_n\|^2\right\}}. \tag{7}$$

Then in the M-step we use the responsibilities to re-estimate the weight matrix $\mathbf{W}$ by solving the following system of linear equations

$$(\mathbf{\Phi}^{\mathrm{T}}\mathbf{G}\mathbf{\Phi})\mathbf{W}_{\mathrm{new}}^{\mathrm{T}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{X} \tag{8}$$

which follow by maximization of the expected complete-data log likelihood. In (8) $\mathbf{\Phi}$ is a $K \times M$ matrix with elements $\Phi_{ij} = \phi_j(\mathbf{u}_i)$, $\mathbf{X}$ is an $N \times D$ matrix with elements x_{nk}, $\mathbf{R}$ is a $K \times N$ matrix with elements R_{ni}, and $\mathbf{G}$ is a $K \times K$ diagonal matrix with elements $G_{ii} = \sum_n R_{ni}$. The inverse variance parameter is also re-estimated in the M-step using

$$\frac{1}{\beta_{\mathrm{new}}} = \frac{1}{ND} \sum_{n=1}^{N} \sum_{i=1}^{K} R_{ni} \|\mathbf{W}_{\mathrm{new}} \boldsymbol{\phi}(\mathbf{u}_i) - \mathbf{x}_n\|^2. \tag{9}$$

A detailed derivation of the EM algorithm for the GTM can be found in Bishop *et al.* (1998).

We can initialize the parameters $\mathbf{W}$ so that the GTM model initially approximates principal component analysis (PCA). To do this, we first evaluate the data covariance matrix and obtain the eigenvectors corresponding to the q largest eigenvalues, and then we determine $\mathbf{W}$ by minimizing the sum-of-squares error between the projections of the latent points into data space by the GTM model and the corresponding projections obtained from PCA. The value of β^{-1} is initialized to be the larger of either the $q + 1$ eigenvalue from PCA (representing the variance of the data away from the PCA sub-space) or the square of half of the grid spacing of the PCA-projected latent points in data space.

The latent space of the GTM is generally chosen to have a low dimensionality (typically $q = 2$). Although it is straightforward to formulate the GTM for latent spaces of any dimension, the model becomes computationally intractable if q becomes large, since the number of nodes in the regular grid grows exponentially with q (as does the number of basis functions). The same problem arises for the SOM. One approach to solving this problem is discussed in Section 5.

The batch SOM can be related to the GTM by considering the limit in which the inverse variance parameter $\beta \to \infty$. This is analogous to the relation between a Gaussian mixture model trained by EM and the K-means algorithm (which can be obtained from the Gaussian mixture model by taking the limit in which the component variances go to zero). For large data sets in many dimensions, the dominant computational cost of the GTM arises in the E-step due to the evaluation of the quantities $\|\mathbf{m}_i - \mathbf{x}_n\|^2$ corresponding to the Euclidean distances between each reference vector and each data point. Since this same computation must also be performed for the self-organizing map, the computational efficiency of the GTM and the batch SOM, for large data sets in high dimensions, are roughly comparable. Many of the techniques used to speed up the learning phase of the SOM can also be adapted to the GTM model.

One useful modification to the standard GTM is to use penalized maximum likelihood by adding a regularization term to the log likelihood in (5). The simplest example is a quadratic regularizer of the form

$$\frac{1}{2}\alpha\|\mathbf{w}\|^2 \tag{10}$$

where $\mathbf{w}$ is a column vector consisting of the concatenation of the successive columns of $\mathbf{W}$, and the hyperparameter α is a fixed constant. Techniques for treating α probabilistically are discussed in Section 6, where the regularizer (10) will be interpreted as the logarithm of a Gaussian prior distribution over the weights. Inclusion of the regularizer (10) leads to a simple modification to the M-step (8) of the EM algorithm to give

$$\left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{G}\mathbf{\Phi} + \frac{\alpha}{\beta}\mathbf{I}\right)\mathbf{W}_{\mathrm{new}}^{\mathrm{T}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{X} \tag{11}$$

where $\mathbf{I}$ is the $M \times M$ unit matrix.

A more complete discussion of the GTM model, and of its relation to the SOM, is given in Bishop *et al.* (1998). Papers relating to the GTM, and a software implementation of the GTM in Matlab, are available from

`http://www.ncrg.aston.ac.uk/GTM/`.

2 Incremental Learning

The version of the GTM described in Section 1.1 uses batch learning in which all of the data points are used together to update the model parameters. For large data sets this may become computationally wasteful since the M-step is performed only after all of the data points have been considered. Significant computational savings could potentially be obtained by updating the parameters incrementally using data points one at a time, or in small batches. This is particularly advantageous if there is significant redundancy in the data set. We therefore consider a sequential EM algorithm for the GTM and provide an outline proof of its convergence.

Suppose that, at a given stage of the algorithm, we have current estimates for $\{R_{ni}\}$ as well as for $\mathbf{W}$ and β. If the next data point is $\mathbf{x}_m$ then we can use (6) to evaluate the corresponding value for R_{mi}^{new}, while leaving the remaining R_{ni} for $n \neq m$ unchanged. Then we can revise our estimate of $\mathbf{G}$ using $G_{ii}^{\text{new}} = G_{ii} + R_{mi}^{\text{new}} - R_{mi}$ and similarly revise our estimate of $\mathbf{RX}$ using $(\mathbf{RX})_i^{\text{new}} = (\mathbf{RX})_i + (R_{mi}^{\text{new}} - R_{mi})\mathbf{x}_m$. We then solve (8) to find $\mathbf{W}^{\text{new}}$ and subsequently obtain β^{new} using

$$\frac{1}{\beta_{\text{new}}} = \frac{1}{\beta} + \frac{1}{ND}\sum_{i=1}^{K} R_{im}^{\text{new}} \|\mathbf{W}_{\text{new}}\boldsymbol{\phi}(\mathbf{u}_i) - \mathbf{x}_m\|^2 - \frac{1}{ND}\sum_{i=1}^{K} R_{im}\|\mathbf{W}\boldsymbol{\phi}(\mathbf{u}_i) - \mathbf{x}_m\|^2 \tag{12}$$

which follows from (9).

This incremental EM algorithm is effectively performing a partial E-step since we are updating only one of the $\{R_{ni}\}$. A general proof that such algorithms still have guaranteed convergence properties was demonstrated by Neal and Hinton (1998). Here we give an outline of their proof in the context of the GTM. Consider the function

$$\mathcal{F}(\{R_{ni}\}, \mathbf{W}, \beta) = \sum_n \sum_i R_{ni} \ln\left\{\frac{1}{K} p(\mathbf{x}|i, \mathbf{W}, \beta)\right\} - \sum_n \sum_i R_{ni} \ln R_{ni} \tag{13}$$

in which the $\{R_{ni}\}$ are regarded as arbitrary non-negative numbers satisfying $\sum_i R_{ni} = 1$ for all n. The quantity $\mathcal{F}$ is analogous to the (negative) free energy in statistical physics. If we maximize (13) with respect to the R_{ni}, using Lagrange multipliers to take account of the summation constraints, we obtain the result (6). If we then subsequently maximize over $\mathbf{W}$ and β keeping the R_{ni} fixed, we recover the standard M-step equations (8) and (9). Our partial E-step corresponds to maximizing $\mathcal{F}$ with respect to R_{mi} while keeping the remaining R_{ni} for $n \neq m$ fixed. It is easily shown (Neal and Hinton, 1998) that a (local or global) maximum of $\mathcal{F}$ corresponds to a (local or global) maximum of the true log likelihood. Thus our algorithm is guaranteed to increase $\mathcal{F}$ until we reach a maximum likelihood solution.

Comparison of the incremental EM algorithm with the standard batch approach for a simple Gaussian mixture model by Neal and Hinton (1998) demonstrated the potential for substantial improvements in speed of convergence. In the case of the GTM, each M-step requires the solution of a set of coupled linear equations given by (8) and the computational cost of doing so may offset much of the gain of using an incremental approach involving data points considered one at a time. This is easily resolved by taking batches of data points and using these to update the corresponding responsibilities before performing the M-step, thereby keeping the overhead of the M-step small while still ensuring that the overall cost of one EM cycle does not scale with the size of the data set. Again, at each iteration the function $\mathcal{F}$ in (13) is increased, thereby providing a guarantee of convergence. For sufficiently large data sets it will always be computationally efficient to use an incremental approach, for which there will exist an optimal batch size.

We see that the initialization of the R_{ni} does not have to be consistent with the initial values of $\mathbf{W}$ and β, so that we can, for instance, simply set all of the $R_{ni} = 1/K$. Other variations of the EM algorithm are also possible. For instance, it will often be the case that many of the responsibilities take very small values, particularly in later stages of the optimization. If these values are frozen (i.e. not recomputed when the corresponding data points are presented) then the analysis based on (13) again shows that the value of $\mathcal{F}$ will not decrease and so a stable algorithm will result.

3 A Manifold-Aligned Noise Model

The noise model (3) was introduced primarily to account for the variance of the data away from the underlying non-Euclidean manifold. However, since the latent points are discrete it also has to account for variance locally along the directions of the manifold. Depending on the distribution of the data, and the density of points on the manifold, these variances may have quite different values. We can accommodate this effect by generalizing (3) to allow for different variances in directions which are (locally) parallel and perpendicular to the manifold, as illustrated in Figure 2. In particular, we would like to ensure that the variance of the noise

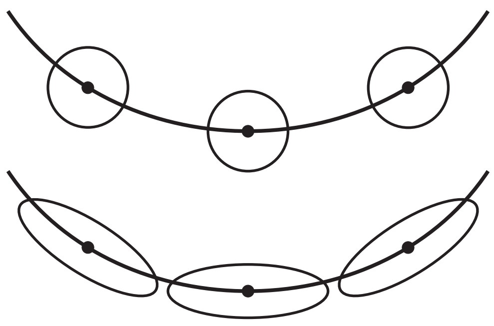


Figure 2: Illustration of the generalization of the noise model to allow for different variances parallel and perpendicular to the GTM manifold. The top figure shows the standard GTM model, with the manifold shown as a curve and the Gaussian component densities represented as circles. In the bottom figure the noise model is generalized to a manifold-aligned non-isotropic covariance model.

distribution in directions tangential to the manifold is never significantly less than the square of the typical distance between neighbouring nodes, so that there is a smooth distribution along the manifold even when the noise variance perpendicular to the manifold becomes small.

We can construct a suitable covariance matrix as follows. The derivatives of the mapping function $\mathbf{y}(\mathbf{u}; \mathbf{W})$ with respect to the latent space coordinates $u_1, \ldots, u_q$ represent linearly independent vectors lying tangentially to the manifold at the point $\mathbf{u}$. The covariance matrix can then be constructed in the form

$$\mathbf{C}_i = \frac{1}{\beta}\mathbf{I} + \eta \sum_{l=1}^{q} \frac{\partial \mathbf{y}}{\partial u_{il}} \frac{\partial \mathbf{y}^{\mathrm{T}}}{\partial u_{il}} \tag{14}$$

where u_{il} is the lth component of $\mathbf{u}_i$, and η is a scaling factor equal to (some multiple of) the distance between neighbouring nodes in latent space. The required derivatives are easily calculated since

$$\frac{\partial \mathbf{y}}{\partial u_{il}} = \mathbf{W}\boldsymbol{\psi}_{il} \tag{15}$$

where $\boldsymbol{\psi}_{il}$ are the (fixed) partial derivatives of the basis functions $\boldsymbol{\phi}(\mathbf{u}_i)$ with respect to u_{il}.

This modification to the model results in a more complex E-step since the component densities of the mixture distribution take the form

$$p(\mathbf{x}|i) = \left(\frac{1}{2\pi}\right)^{D/2} |\mathbf{C}_i|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{m}_i - \mathbf{x})^{\mathrm{T}}\mathbf{C}_i^{-1}(\mathbf{m}_i - \mathbf{x})\right\} \tag{16}$$

and hence require that the inverse and the determinant of each covariance matrix be evaluated for each latent point. The inverse is efficiently computed using q successive applications of the matrix inversion lemma

$$(\mathbf{A} + \mathbf{v}\mathbf{v}^{\mathrm{T}})^{-1} = \mathbf{A}^{-1} - \frac{(\mathbf{A}^{-1}\mathbf{v})(\mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1})}{1 + \mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{v}} \tag{17}$$

which is easily verified by multiplying both sides by $(\mathbf{A} + \mathbf{v}\mathbf{v}^{\mathrm{T}})$. Similarly, the M-step equations become more complex since the covariance matrices now depend on the weight matrix $\mathbf{W}$. For the $\mathbf{W}$-update we therefore approximate the re-estimation formulae for $\mathbf{W}$ by replacing $\mathbf{C}$ with $\beta^{-1}\mathbf{I}$ thereby recovering the standard M-step updates (8) and (9).

As an illustration of this model we use a 1D toy problem in 2D, similar to the one used in Bishop *et al.* (1998). The training data, together with the converged model, are shown in Figure 3.

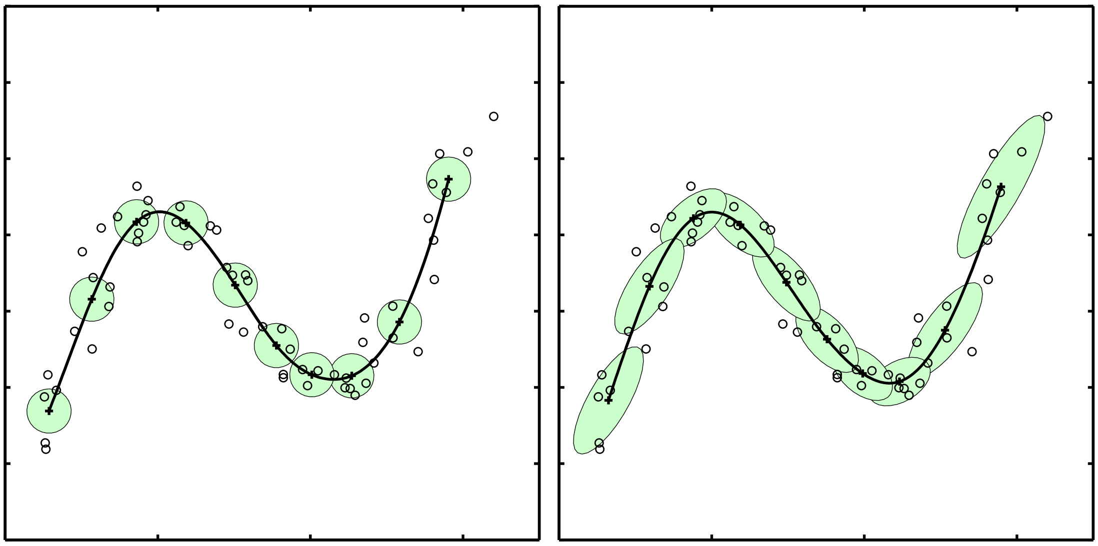


Figure 3: Application of the manifold-aligned noise model to a toy problem. The plots show the data space, with the training data plotted as $\circ$, and the GTM manifold shown as a curve connecting the mixture component centres. Mixture components are plotted as ellipsoids, corresponding to unit Mahalanobis distance, with '+' marking the centres. The left hand plot shows a standard GTM model (giving a log likelihood of -58.9) while the right hand plot shows the modified form of the GTM having a manifold-aligned noise model (giving a log likelihood of -48.3).

This version of the GTM has some similarities to the adaptive subspace SOM model (Kohonen, 1995), since each mixture component now represents a local linear sub-space. A related form of Gaussian mixture model, with general covariance matrices and constrained centres, was described by Williams *et al.* (1993).

4 Discrete Data

The original version of the GTM, as discussed in Section 1, was formulated for the case of data variables which are continuous[1]. We now extend the model to account for discrete data and for

[1]The SOM model is also formulated for continuous variables.

combinations of discrete and continuous variables. Consider first the case of a set of binary data variables $x_k \in \{0,1\}$. As for the case of continuous variables, we assume that the components of $\mathbf{x}$ are conditionally independent, given the latent space label i. We can then express the conditional distribution of the binary vector $\mathbf{x}$, given i, using a binomial (Bernoulli) distribution of the form

$$p(\mathbf{x}|i) = \prod_k m_{ik}^{x_k}(1 - m_{ik})^{1-x_k} \tag{18}$$

where the conditional means m_{ik} are given by $m_{ik} = \sigma(\mathbf{w}_k^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{u}_i))$, $\sigma(a) = (1 + \exp(-a))^{-1}$ is the logistic sigmoid function, and $\mathbf{w}_k$ is the k^{th} column of $\mathbf{W}$. Note that in (18) there is no analogue of the noise parameter β.

Next, suppose instead that the D data variables represent membership of one of D mutually exclusive classes. Again, the data values are binary, but for a given pattern all values are zero except for one component which identifies the class (this is called a 1-of-D coding scheme). In this case we can represent the conditional distribution of the data variables using a multi-nomial distribution (Bishop, 1995) of the form

$$p(\mathbf{x}|i) = \prod_{k=1}^{D} m_{ik}^{x_k} \tag{19}$$

where m_{ik} are defined by a softmax, or normalized exponential, transformation (Bishop, 1995) of the form

$$m_{ik} = \frac{\exp(\mathbf{w}_k^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{u}_i))}{\sum_j \exp(\mathbf{w}_k^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{u}_j))}. \tag{20}$$

Finally, if we have a data set consisting of a combination of continuous, binary and categorical variables, we can formulate the appropriate model by writing the conditional distribution $p(\mathbf{x}|i)$ as a product of Gaussian, binomial and multi-nomial distributions. This represents the standard conditional independence framework (used in many latent variable models) in which the observed variables are independent given the latent variables.

We can again estimate the parameters in such models using the EM algorithm. The E-step again takes the form (6). However, the M-step now requires non-linear optimization, although this may be performed efficiently using the iterative re-weighted least squares (IRLS) algorithm (McCullagh and Nelder, 1989). Note that it is not necessary to perform an exact optimization in the M-step, and indeed it will typically be computationally efficient to perform only a partial optimization, corresponding to the generalized EM (GEM) algorithm (Dempster *et al.*, 1977).

5 A Semi-linear Model

We have already noted that the computational cost of the standard GTM grows exponentially with the number of latent dimensions (as is also the case for the SOM). One approach to dealing with high-dimensional latent spaces would simply be to consider a random sampling of the latent space, as used by MacKay (1995) in the 'density network' model. However, such sampling effectively becomes very sparse as the dimensionality of the latent space increases, so again this approach is limited to low values of q.

An alternative approach is to introduce a *semi-linear* model, in which the data variables depend non-linearly on a small number of dimensions of latent space, and depend linearly on the remaining dimensions. Linear latent variable models include factor analysis (Everitt, 1984) and probabilistic principal component analysis (Tipping and Bishop, 1997c). The GTM can be regarded as one possible non-linear generalization of such models.

First, suppose we consider a model in which the data variables are purely linear functions of the latent variables, so that (1) becomes $\mathbf{y} = \mathbf{V}\mathbf{u} + \boldsymbol{\mu}$ in which $\mathbf{V}$ is a $D \times q$ matrix, and for convenience we have introduced an explicit mean vector $\boldsymbol{\mu}$. Instead of using a finite discrete grid in latent space, we can consider a prior distribution over $\mathbf{u}$ given by a zero mean, unit covariance Gaussian. The marginal distribution of the data variables is then given by the convolution of two Gaussian functions and can be evaluated analytically, with the result that the overall density model is a Gaussian with mean $\boldsymbol{\mu}$ and covariance

$$\mathbf{C} = \beta^{-1}\mathbf{I} + \mathbf{V}\mathbf{V}^{\mathrm{T}}. \tag{21}$$

An important property of this model, demonstrated by Tipping and Bishop (1997b,1997c), is that the maximum likelihood solution for $\mathbf{V}$, β and $\boldsymbol{\mu}$ can be found in closed form. The solution for $\boldsymbol{\mu}_{\mathrm{ML}}$ is straightforward and is given by the sample mean. If we now introduce the sample covariance matrix given by

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\mathrm{ML}})^{\mathrm{T}} \tag{22}$$

then the maximum likelihood solution for $\mathbf{V}$ is given by

$$\mathbf{V}_{\mathrm{ML}} = \mathbf{U}(\boldsymbol{\Lambda} - \beta^{-1}\mathbf{I})^{1/2} \tag{23}$$

where $\mathbf{U}$ is a $D \times q$ matrix whose columns are the principal eigenvectors of $\mathbf{S}$ (i.e. the eigenvectors corresponding to the q largest eigenvalues) with corresponding eigenvalues in the diagonal $q \times q$ matrix $\boldsymbol{\Lambda}$. Finally the maximum-likelihood estimator of β is given by

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{D-q}\sum_{j=q+1}^{D}\lambda_j \tag{24}$$

where we have ordered the eigenvalues such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D$. The result (24) has a clear interpretation as the variance 'lost' in the projection, averaged over the lost dimensions. This model therefore represents a *probabilistic* formulation of standard principal components analysis.

Now we consider a semi-linear formulation of the GTM in which the data variables depend non-linearly on a few discretized latent variables and linearly on the remaining Gaussian latent variables. Marginalizing over all of the latent variables we obtain the following density model

$$p(\mathbf{x}|\mathbf{W},\mathbf{V},\beta) = \sum_{i=1}^{K}\frac{1}{K}\left(\frac{1}{2\pi}\right)^{D/2}|\mathbf{C}|^{-1/2}\exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}-\mathbf{m}_i)\right\} \tag{25}$$

where $\mathbf{C}$ is given by (21). The density model (25) can be interpreted as a mixture of probabilistic PCA models with equal covariance matrices and with means $\mathbf{m}_i$ lying on the GTM manifold. In order to maximize the corresponding log likelihood, we could treat both the discretized and the continuous latent variables as jointly missing data and apply the EM algorithm. However, we can make use of the above result for probabilistic PCA by treating only the discrete latent variables (corresponding to the 'non-linear dimensions') as missing. The E-step of the corresponding EM algorithm involves the evaluation of the responsibilities R_{ni} which are given by

$$R_{ni} = \frac{\exp\left\{-\frac{1}{2}(\mathbf{x}_n - \mathbf{m}_i)^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}_n - \mathbf{m}_i)\right\}}{\sum_j \exp\left\{-\frac{1}{2}(\mathbf{x}_n - \mathbf{m}_j)^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}_n - \mathbf{m}_j)\right\}}. \tag{26}$$

In the M-step we must maximize the expected complete-data log likelihood (Dempster *et al.*, 1977; Bishop, 1995) given by

$$\langle \mathcal{L}_{\mathrm{C}} \rangle = \sum_{n=1}^{N}\sum_{i=1}^{K} R_{ni}\left\{-\ln K - \frac{D}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{C}| - \frac{1}{2}(\mathbf{x}_n - \mathbf{m}_i)^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}_n - \mathbf{m}_i)\right\}. \tag{27}$$

Thus we see that $\langle \mathcal{L}_{\mathrm{C}} \rangle$ depends on the data only through the weighted covariance matrix

$$\mathbf{S} = \sum_{n=1}^{N}\sum_{i=1}^{K} R_{ni}(\mathbf{x}_n - \mathbf{m}_i)(\mathbf{x}_n - \mathbf{m}_i)^{\mathrm{T}}. \tag{28}$$

Maximizing (27) jointly over $\mathbf{W}$, $\mathbf{V}$ and β, for fixed R_{ni}, can then be accomplished as follows. We first note that the maximum over $\mathbf{W}$ does not depend $\mathbf{V}$ or β and is given by the solution to (8). We now use this new value for $\mathbf{W}$ to evaluate $\{\mathbf{m}_i\}$ and hence evaluate $\mathbf{S}$ given by (28). Finally we can find the eigenvector/eigenvalue decomposition of $\mathbf{S}$ and use this to solve for $\mathbf{V}$ and β using (23) and (24).

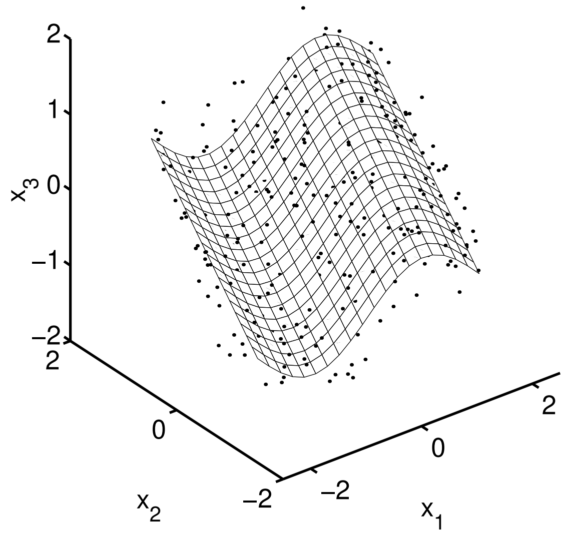
We have seen that, in the probabilistic PCA model, the solutions for $\mathbf{V}$ and β have explicit, closed-form solutions. However, it was noted by Tipping and Bishop (1997a) that, for problems in which the dimensionality D of the data space is high, it may be more efficient to treat the continuous latent variables as missing data and apply the EM algorithm. Although this results in an iterative optimization scheme, each step requires $O(ND)$ operations, compared with the $O(ND^2)$ operations needed to evaluate the covariance matrix. Provided the number of iterations of EM needed to reach satisfactory convergence is sufficiently smaller than D, there can be an overall computational saving, which typically improves as D increases. A derivation and discussion of this EM algorithm is given in Tipping and Bishop (1997c).

As a demonstration of this model, a simple data set was generated in a 3-dimensional space. The first two variables, x_1 and x_2 were discretized over a rectangular grid, while the third variable, x_3, was computed from x_1 and x_2 with the formula

$$x_3 = x_2 + 0.5\sin(0.5\pi x_1) \tag{29}$$

so that x_3 depends linearly on x_2 and non-linearly on x_1. Gaussian noise was then added to x_1, x_2 and x_3. A semi-linear GTM with one non-linear latent variable (using 10 nodes and 5 basis functions) and one linear latent variable was trained on this data set, starting from a PCA initialization. The trained model, shown in Figure 4, captures the structure of the data well. Note that this model appears to be fairly sensitive to the initialization of its parameters, and is relatively prone to finding local minima.

The semi-linear model for the latent space distribution can easily be combined with the type of mixed discrete-continuous distributions for the data space distribution discussed in Section 4.



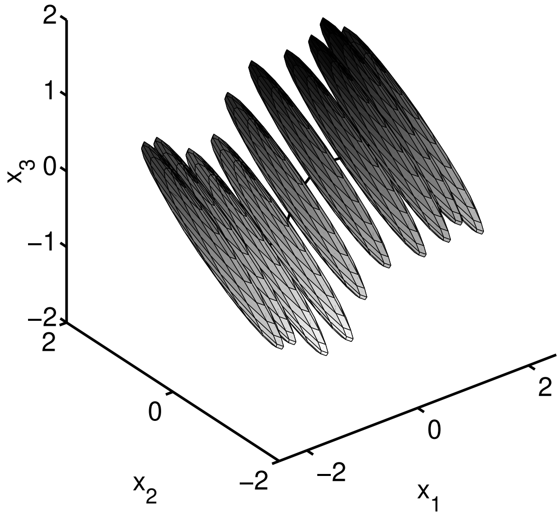


Figure 4: Demonstration of the semi-linear model. The left plot shows a 2-dimensional manifold embedded in data space, along with the data set generated by sampling points on the manifold and adding Gaussian noise. The right hand plot shows the result of fitting a semi-linear GTM model having one non-linear latent dimension with 10 latent points, and one linear dimension. The mixture components are plotted as ellipsoids corresponding to unit Mahalanobis distance.

6 Bayesian Inference for Hyperparameters

An important issue in maximum likelihood density estimation is that of model complexity, which in the context of the GTM is determined in large part by the 'stiffness' of the manifold. A more flexible manifold can provide a better fit to the training data, but if the effective complexity is too high the model may adapt too closely to the specific data set and thereby give a poorer representation of the underlying distribution from which the data was generated (a phenomenon known as *over-fitting*).

The effective model complexity in the GTM is controlled by the number and form of the basis functions as well as by the regularization coefficient. Although it would be possible to explore a range of model complexities by altering the number of basis functions, it is computationally more convenient to arrange for the complexity to be governed by one or more real-valued parameters, and to explore the corresponding continuous space. We shall denote such parameters generically by σ, which might, for example, represent a common width parameter in the case of Gaussian basis functions.

In the discussion of the GTM in Section 1.1 the parameters $\mathbf{W}$ and β were estimated from the data using maximum (penalized) likelihood, while the regularization coefficient α (as well as any parameters governing the basis functions) was assumed to be constant. Since the GTM represents a probabilistic model, it offers the possibility of a more comprehensive probabilistic treatment using a Bayesian formalism.

In this section it will be convenient to introduce a column vector $\mathbf{w}$ consisting of the concatenation of the successive columns of $\mathbf{W}$. From (4) and (5) the log likelihood function for the GTM is given by

$$\mathcal{L}(\mathbf{w}, \beta, \sigma) = \ln p(\{\mathbf{x}\}|\mathbf{w}, \beta, \sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{i=1}^{K} \frac{1}{K} \left(\frac{\beta}{2\pi} \right)^{D/2} \exp \left[-\frac{\beta}{2} \|\mathbf{m}_i - \mathbf{x}\|^2 \right] \right\}. \tag{30}$$

We now introduce a prior distribution over the weights, which for simplicity we choose to be an isotropic Gaussian distribution of the form

$$p(\mathbf{w}|\alpha) = \left(\frac{\alpha}{2\pi} \right)^{W/2} \exp \left\{ -\frac{\alpha}{2} \|\mathbf{w}\|^2 \right\} \tag{31}$$

where W is the total number of elements in $\mathbf{w}$. Since α controls the distribution of other parameters it is often called a hyperparameter, and we shall similarly use this terminology to describe β and σ also. A full Bayesian treatment would involve the introduction of prior distributions over α, β and σ followed by a marginalization over all the parameters and hyperparameters in the model. Instead we estimate values for the hyperparameters by maximizing their marginal likelihood $p(\{\mathbf{x}_n\}|\alpha, \beta, \sigma)$ in which we have integrated over $\mathbf{w}$. This corresponds to the type-II maximum likelihood procedure (Berger, 1985), also known as the evidence approximation (MacKay, 1992a; Bishop, 1995). Utsugi (1997) applies a similar Bayesian treatment to a generalized form of the elastic net (Durbin and Willshaw, 1987; Durbin *et al.*, 1989), which is also a probabilistic model having close connections to the SOM.

The marginal likelihood for the GTM model is given by

$$p(\{\mathbf{x}_n\}|\alpha, \beta, \sigma) = \int p(\{\mathbf{x}_n\}|\mathbf{w}, \beta, \sigma) p(\mathbf{w}|\alpha)\, d\mathbf{w}. \tag{32}$$

Since this integral is analytically intractable, we follow MacKay (1992b) and make a local Gaussian approximation to the posterior distribution over $\mathbf{w}$ in the neighbourhood of a mode. Maximizing the posterior distribution (for given values of α, β and σ) corresponds to maximizing the penalized log likelihood, and the solution for $\mathbf{w}$ was given in (11). Suppose we have found a maximum $\mathbf{w}_*$ of the posterior distribution. If we define

$$S(\mathbf{w}, \alpha, \beta, \sigma) = -\ln \left\{ p(\{\mathbf{x}_n\}|\mathbf{w}, \beta, \sigma) p(\mathbf{w}|\alpha) \right\} \tag{33}$$

then we can then write (32) in the form

$$\begin{aligned} p(\{\mathbf{x}_n\}|\alpha, \beta, \sigma) &= \int \exp \big\{ -S(\mathbf{w}, \alpha, \beta, \sigma) \big\}\, d\mathbf{w} \\ &\simeq \exp \big\{ -S(\mathbf{w}_*, \alpha, \beta, \sigma) \big\} \int \exp \left\{ -\frac{1}{2} (\mathbf{w} - \mathbf{w}_*)^{\mathrm{T}} \mathbf{A} (\mathbf{w} - \mathbf{w}_*) \right\} d\mathbf{w} \\ &= \exp \big\{ -S(\mathbf{w}_*, \alpha, \beta, \sigma) \big\} (2\pi)^{W/2} |\mathbf{A}|^{-1/2} \end{aligned} \tag{34}$$

where we we have performed a Taylor expansion of the logarithm of the integrand and retained terms up to second order. Note that the first order terms vanish since the integrand is proportional to the posterior distribution, through Bayes' theorem, and we are at a local maximum. We have also introduced the Hessian matrix $\mathbf{A}$ given by the second derivatives of S with respect to the elements of $\mathbf{w}$, evaluated at $\mathbf{w}_*$. Making use of (30) and (31) we then obtain the log-evidence for σ, α and β in the form

$$\ln p(\{\mathbf{x}_n\}|\alpha, \beta, \sigma) = \mathcal{L}(\mathbf{w}_*, \beta, \sigma) - \frac{\alpha}{2} \|\mathbf{w}_*\|^2 - \frac{1}{2} \ln |\mathbf{A}| + \frac{W}{2} \ln \alpha. \tag{35}$$

Although the Hessian matrix can be calculated exactly, the resulting expression is computationally expensive to evaluate. Here we consider an approximation obtained by neglecting terms involving derivatives of the responsibilities R_{ni} with respect to $\mathbf{w}$. This approximation becomes exact when the responsibility for each data point n is taken by just one of the mixture components (as is often effectively the case during the later stages of GTM training) so that $R_{ni} \in \{0, 1\}$. The Hessian matrix then takes a block diagonal form (with one block corresponding to each column from the original $\mathbf{W}$ matrix) in which all blocks are identical and have the form $\beta\mathbf{\Phi}^{\mathrm{T}}\mathbf{G}\mathbf{\Phi} + \alpha\mathbf{I}$. Note that this expression will already have been evaluated for use in the regularized M-step (11).

We can maximize (35) with respect to α and β by setting the respective derivatives to zero, yielding the update formulae

$$\alpha = \frac{\gamma}{\|\mathbf{w}_*\|^2} \tag{36}$$

and

$$\beta = \frac{ND - \gamma}{\sum_n \sum_i R_{ni} \|\mathbf{x}_n - \mathbf{m}_i\|^2} \tag{37}$$

where we have defined

$$\gamma = \sum_{i=1}^{W} \frac{\lambda_i - \alpha}{\lambda_i} \tag{38}$$

and λ_i are the eigenvalues of $\mathbf{A}$. Note that we have neglected terms involving derivatives of $\mathbf{w}_*$ with respect to α and β. Comparison of (37) with the corresponding maximum likelihood update (9) shows that they have identical form except for the appearance of γ which can be interpreted as the *effective* number of $\mathbf{w}$-parameters in the model (MacKay, 1992a).

In a practical implementation, maximization with respect to $\mathbf{W}$ using the EM algorithm is interleaved with re-estimation of α and β. Since the dependence of the marginal log likelihood on σ is more complex we do not obtain a simple re-estimation formula, but since we are now down to a single variable, we can simply evaluate (35) for a range of different σ values, while estimating α and β on-line, and select the model with the highest log-evidence score.

To evaluate this method, synthetic data was generated from a curved 2D manifold in a 3D space. 20 data sets were generated by adding random Gaussian noise with standard deviation 0.2 to 400 points drawn from a regular grid on the manifold. A corresponding test data set of 1024 points was also generated. A GTM model, with a 15×15 latent grid and a 5×5 grid of Gaussian basis functions with common width parameter σ, was trained on the 20 data sets, each time starting from a PCA initialization. The plots in Figure 5 show the log-evidence and log-likelihoods after training, plotted against $\log_2(\sigma)$. The data generating manifold is shown in the top left panel of Figure 6, together with a sample data set. Figure 6 also shows an example of the model with the highest log-evidence ($\sigma = 1$), together examples of models in which σ is fixed to values which are either too small or too large.

7 Gaussian Process Formulation

In the original GTM, described in Section 1.1, there is a hard constraint on the form of the latent-space to data-space mapping due to the finite number of basis functions used, as well as

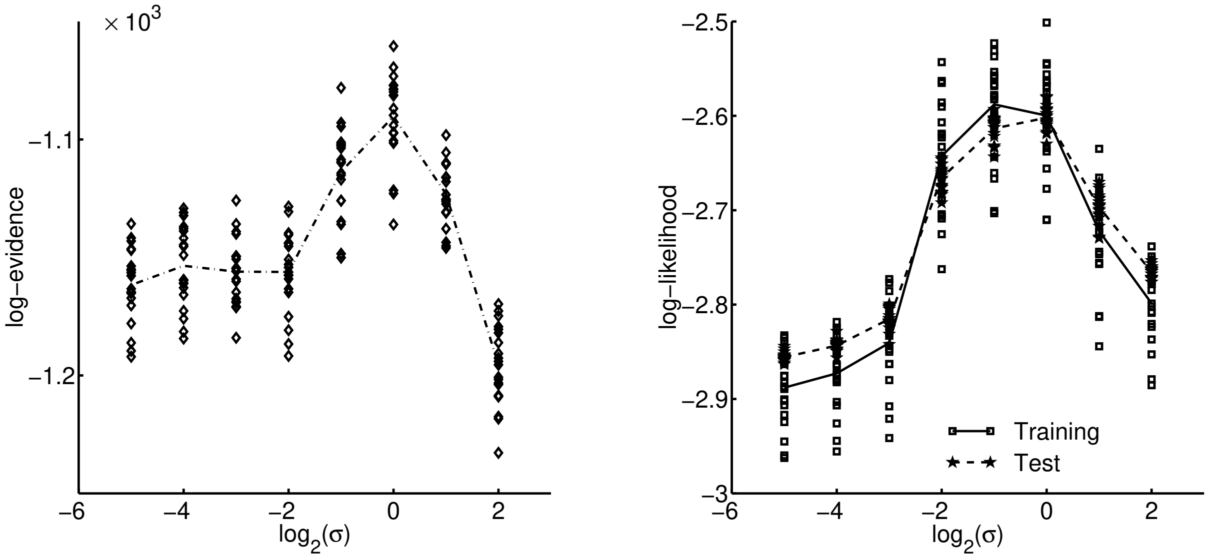


Figure 5: Log-evidence and log-likelihood plotted against $\log_2(\sigma)$. Each plot shows the individual results for the 20 data sets after training, together with a line plot summarizing the mean. The log-likelihood plot shows results for the training and test sets (normalized by the respective size of the data sets).

a soft constraint due to the regularization term (10). An alternative approach is to enforce the smoothness of this mapping entirely through regularization, using a *Gaussian process* prior over functions. For each dimension j in the data space ($j = 1, \ldots, D$), let $\mathbf{m}^{(j)}$ be a vector of length K consisting of the jth components of $\mathbf{m}_1$ through $\mathbf{m}_K$, so that if the column vectors $\mathbf{m}_i$ are arranged side-by-side, one obtains the $D \times K$ matrix $\mathbf{M}$ in which $\mathbf{m}^{(j)}$ is the jth row of this matrix. Consider a Gaussian prior distribution on the centre locations given by

$$p(\mathbf{M}) = \prod_{j=1}^{D} \frac{1}{(2\pi)^{K/2}|\mathbf{B}^{(j)}|^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{m}^{(j)\mathrm{T}}(\mathbf{B}^{(j)})^{-1}\mathbf{m}^{(j)}\right\} \tag{39}$$

where the $\mathbf{B}^{(j)}$'s are positive definite matrices. In practice it will usually not be necessary to use different covariance matrices for each dimension, and the $\mathbf{B}^{(j)}$'s will be denoted generically by $\mathbf{B}$.

The EM algorithm is now used to maximize the penalized log likelihood

$$\mathcal{L}_{\mathrm{p}}(\mathbf{M}, \beta) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\mathbf{M}, \beta) + \ln p(\mathbf{M}). \tag{40}$$

In the E-step, the usual responsibilities are calculated. With these fixed, the M-step involves the inversion of a $K \times K$ matrix (where K is the number of latent points). This should be contrasted with equation (8) which involves the inversion of an $M \times M$ matrix (in which M is the number of basis functions). The $\mathbf{m}$'s can be initialized via PCA, as in the standard GTM.

We now focus on the specification of $\mathbf{B}$. The theory of Gaussian process regression (Whittle, 1963; Williams and Rasmussen, 1996) or equivalently regularization networks (Poggio and Girosi, 1990) allows $\mathbf{B}$ to be quite general. The covariance between m_{kj} and m_{lj} can be taken to depend on the positions of their respective nodes $\mathbf{u}_k$ and $\mathbf{u}_l$, so that $B_{kl} = f(\mathbf{u}_k, \mathbf{u}_l)$, where $f(\cdot, \cdot)$ is a

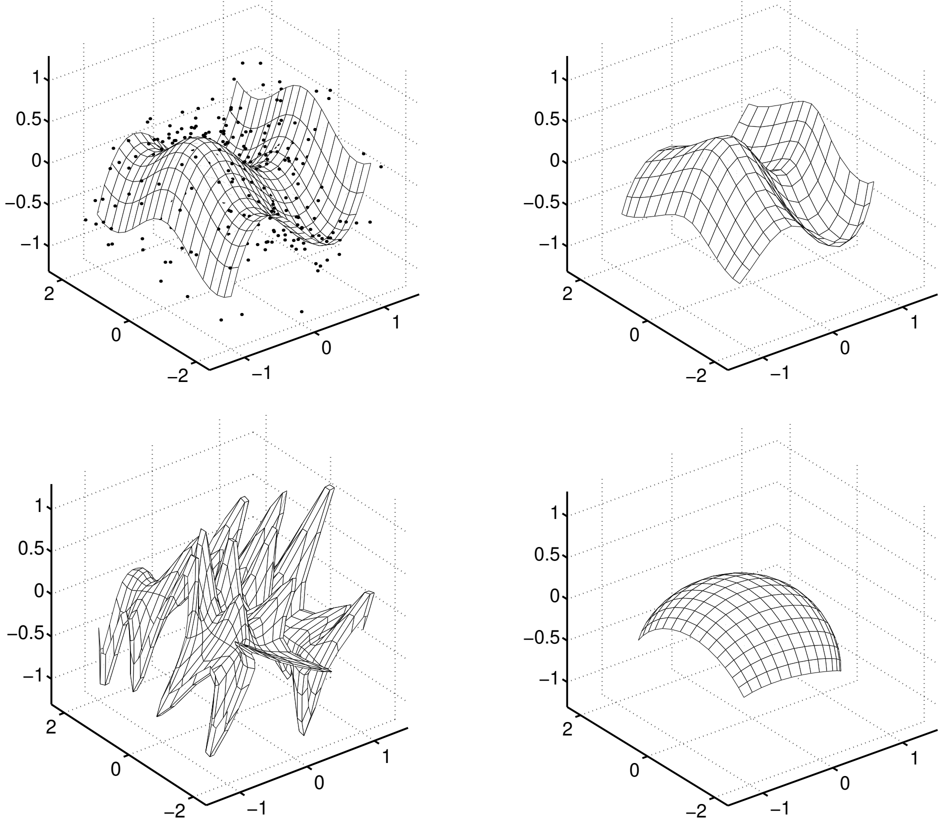


Figure 6: Illustration of Bayesian parameter re-estimation. The data generating manifold is shown in the top left plot, together with a sample data set. The top right plot shows the manifold of a GTM model trained on this data set, with σ fixed to 1 and α and β being re-estimated during training, final values being $\alpha = 9.2$ and $\beta = 18.3$. The bottom left plot shows a significantly more flexible model, $\sigma = 0.25$, trained using the standard GTM and no weight regularization; the final estimated value for β was 40.9. The bottom right plot, shows a much stiffer model, $\sigma = 2$, trained using the standard GTM and constant weight regularization of 50; the final estimated value for β was 10.1.

covariance function. For example, one can use

$$B_{kl} = v \exp\left\{-\frac{\| \mathbf{u}_k - \mathbf{u}_l \|^2}{2\lambda^2}\right\} \tag{41}$$

where λ is a length scale in the latent space and v sets the overall scale of $\mathbf{B}$. A wide variety of covariance functions can be used, and there is a substantial literature concerning valid covariance functions (see, for example, Yaglom (1987)).

In the original GTM, and in the Gaussian process formulation of the GTM outlined above, the overall regression problem decomposes into separate problems for each dimension in the data space. (These problems are coupled only through the responsibilities.) However, if the prior on $\mathbf{M}$ couples the various dimensions (as in the technique of co-kriging in geostatistics (Cressie, 1993)), then this would no longer be the case, and the M-step would involve a $KD \times KD$ matrix inversion.

Utsugi (1997) provides a similar analysis to that above, but uses a relatively simple covariance matrix $\mathbf{B}$ based on a discretized approximation to derivatives of the $\mathbf{M}$ surface[2]. He gives details of the EM algorithm which can easily be extended to the more general case, and also provides a Bayesian treatment of hyper-parameters which is similar in spirit to that given in Section 6. By specifying $\mathbf{B}$ through a covariance function we would expect to obtain rather better control over the prior on $\mathbf{M}$. For example, the length scale λ in (41) affords direct and readily understandable control over the flexibility of the latent-space to data-space mapping. One other important advantage of formulating the Gaussian process prior via the covariance function, rather than through a difference operator as in Utsugi (1997), is that it defines the manifold in the data space not just at the reference vectors but everywhere on the 2-d surface. This can be achieved because the machinery of Gaussian process regression predicts the data-space locations corresponding to new $\mathbf{u}$ points.

The use of spline smoothing for the M-step in work on principal curves (Hastie and Stuetzle, 1989; Tibshirani, 1992) is another example of the use of Gaussian process-type priors over functions in SOM-like models.

An advantage of the Gaussian process formulation of the GTM is that it emphasizes the similarities between the GTM and SOM, by eliminating the use of basis functions in the regression model. Furthermore, the update for $\mathbf{m}^{(j)}$ is given by

$$\mathbf{m}^{(j)} = (\mathbf{G} + \beta^{-1}(\mathbf{B}^{(j)})^{-1})^{-1}\mathbf{G}(\mathbf{G}^{-1}\mathbf{R}\mathbf{x}^{(j)}) \tag{42}$$

where $\mathbf{x}^{(j)}$ is the jth column of $\mathbf{X}$, and $\mathbf{G}^{-1}\mathbf{R}\mathbf{x}^{(j)} \stackrel{def}{=} \overline{\mathbf{x}}^{(j)}$ is the vector of weighted means of the data at the K points in latent space. In the SOM, the update for $\mathbf{m}^{(j)}$ is given by

$$\mathbf{m}^{(j)} = \mathbf{H}^{(j)}\overline{\mathbf{x}}^{(j)} \tag{43}$$

where $\mathbf{H}$ is the neighbourhood function evaluated between pairs of latent points. It is hard to draw an analogy between $\mathbf{H}^{(j)}$ and $\mathbf{B}^{(j)}$ because of the matrix inversions involved in (42).

Another advantage of the Gaussian process formulation is that it avoids issues of discrete model order selection that arise in the GTM concerning the number of basis functions used. However, the continuous parameters that control $p(\mathbf{M})$ through the covariance function still need to be addressed. Utsugi (1997) has discussed a MAP (maximum a-posteriori probability) treatment of these parameters, and a fully Bayesian treatment using Markov chain Monte Carlo methods would also be possible.

One disadvantage of using Gaussian processes to formulate the GTM model is that the matrices to be inverted will be larger than those in the parametric GTM case. However, using up to 1000 nodes in latent space should not present too many problems on modern workstations, and techniques for efficient approximate treatment of Gaussian processes for larger problems have been explored by Gibbs and MacKay (1997).

8 Conclusions

One of the many benefits of the probabilistic foundation of the GTM is that extensions of the model can be formulated in a principled manner, and we have explored a number of such extensions in this paper. There are many other ways in which the basic GTM model can be extended,

[2]In fact Utsugi's matrix is only positive semi-definite due to the presence of a linear null-space.

again by taking advantage of the probabilistic setting. For example, it is straightforward to construct a probabilistic *mixture* of GTM models. The parameters of the component models as well as the mixing coefficients between the models, can be determined by maximum likelihood using the EM algorithm, again retaining the attractive convergence properties. This can be further extended to hierarchical mixtures, as discussed in Bishop and Tipping (1998).

Another refinement of the basic model would be to allow the parameters σ and β to be continuous functions of the latent space variable $\mathbf{u}$, defined by a parametric transformation. Similarly, the individual nodes can be assigned adaptive mixing coefficients (fixed at $1/K$ in the original formulation of the GTM) and these could be independent variables (non-negative and summing to unity) or they could again be smooth functions of $\mathbf{u}$. In all such cases, there is a well-defined learning procedure based on maximization of the likelihood function, and the EM algorithm can be exploited to handle the hidden variables.

In many applications involving real-world data, the data set will suffer from missing values. Provided the values can be assumed to be 'missing at random' (i.e. the missingness is not itself informative) then maximum likelihood specifies that the correct procedure for treating such data is to marginalize over the missing values. For many of the distributions considered in this paper this marginalization is trivial to implement, and corresponds to simply ignoring the missing values, as has been done in the case of the SOM (Samard and Harp, 1992). For more complex distributions, such as the non-isotropic Gaussians considered in Section 3, the marginalization is more complex but still analytically tractable.

While both the SOM and the GTM represent the data in terms of an underlying two-dimensional structure, an elegant property of the GTM is that there exists an explicit manifold defined by the continuous non-linear mapping from latent space to data space specified by the basis functions. The corresponding *magnification factors*, which characterize the way in which portions of the latent space are stretched and distorted by the transformation to data space, can therefore be evaluated as continuous functions of the latent space coordinates using the techniques of differential geometry (Bishop *et al.*, 1997a). This technique can also be applied to the batch version of the SOM (Bishop *et al.*, 1997b), by exploiting the existence of a natural interpolating surface arising through a kernel smoothing interpretation (Mulier and Cherkassky, 1995).

Another role for the GTM is as the emission distribution of a hidden Markov model, leading to *GTM through time* as described in Bishop *et al.* (1997). Finally, we note that the technique of independent component analysis (ICA) (Jutten and Herault, 1991; Comon, 1994; Bell and Sejnowski, 1995) can be formulated as a latent variable model with a linear transformation from latent space to data space (MacKay, 1996; Pearlmutter and Parra, 1996). ICA can therefore be extended to allow non-linear transformations by employing the framework of the GTM (Pajunen and Karhunen, 1997).

In summary, the GTM retains the many appealing features of the SOM, and offers comparable computational speed, while its probabilistic formulation permits a wide variety of extensions to be developed in a theoretically well-founded setting.

Acknowledgements

This work was supported by EPSRC grant GR/K51808: *Neural Networks for Visualization of High-Dimensional Data.* We would like to thank Geoffrey Hinton, Iain Strachan and Michael Tipping for useful discussions, and Tim Cootes and Andreas Lanitis (University of Manchester) for a helpful conversation concerning semi-linear models. Also we wish to thank the anonymous referees for their helpful comments. Markus Svensén is grateful to the SANS group at the Royal Institute of Technology for their hospitality. Chris Bishop and Chris Williams would like to thank the Isaac Newton Institute for Mathematical Sciences in Cambridge for providing such a stimulating research environment during the *Neural Networks and Machine Learning* programme.

References

Bell, A. J. and T. J. Sejnowski (1995). An information maximization approach to blind separation and blind deconvolution. *Neural Computation* **7** (6), 1129–1159.

Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis* (Second ed.). New York: Springer-Verlag.

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

Bishop, C. M., G. E. Hinton, and I. G. D. Strachan (1997). GTM through time. In *Proceedings IEE Fifth International Conference on Artificial Neural Networks, Cambridge, U.K.*, pp. 111–116.

Bishop, C. M., M. Svensén, and C. K. I. Williams (1997a). Magnification factors for the GTM algorithm. In *Proceedings IEE Fifth International Conference on Artificial Neural Networks, Cambridge, U.K.*, pp. 64–69.

Bishop, C. M., M. Svensén, and C. K. I. Williams (1997b). Magnification factors for the SOM and GTM algorithms. In *Proceedings 1997 Workshop on Self-Organizing Maps, Helsinki University of Technology, Finland.*, pp. 333–338.

Bishop, C. M., M. Svensén, and C. K. I. Williams (1998). GTM: the Generative Topographic Mapping. *Neural Computation* **10** (1), 215–234.

Bishop, C. M. and M. E. Tipping (1998). A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (3), 281–293.

Comon, P. (1994). Independent component analysis: a new concept? *Signal Processing* **36**, 287–314.

Cox, R. T. (1946). Probability, frequency and reasonable expectation. *American Journal of Physics* **14** (1), 1–13.

Cressie, N. A. C. (1993). *Statistics for Spatial Data.* New York: John Wiley.

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.

Durbin, R., R. Szeliski, and A. Yuille (1989). An analysis of the elastic net approach to the travelling salesman problem. *Neural Computation* **1** (3), 348–358.

Durbin, R. and D. Willshaw (1987). An analogue approach to the travelling salesman problem. *Nature* **326**, 689–691.

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

Gibbs, M. and D. J. C. MacKay (1997). Efficient implementation of Gaussian processes. Draft manuscript, available from `http://wol.ra.phy.cam.ac.uk/mackay/homepage.html`.

Hastie, T. and W. Stuetzle (1989). Principal curves. *Journal of the American Statistical Association* **84** (406), 502–516.

Jutten, C. and J. Herault (1991). Blind separation of sources. *Signal Processing* **24**, 1–10.

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

Kohonen, T. (1995). *Self-Organizing Maps.* Berlin: Springer-Verlag.

MacKay, D. J. C. (1992a). Bayesian interpolation. *Neural Computation* **4** (3), 415–447.

MacKay, D. J. C. (1992b). A practical Bayesian framework for back-propagation networks. *Neural Computation* **4** (3), 448–472.

MacKay, D. J. C. (1995). Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research, A* **354** (1), 73–80.

MacKay, D. J. C. (1996). Maximum likelihood and covariant algorithms for independent component analysis. Draft manuscript, available from `http://wol.ra.phy.cam.ac.uk/mackay/homepage.html`.

McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models* (2 ed.). Chapman and Hall.

Mulier, F. and V. Cherkassky (1995). Self-organization as an iterative kernel smoothing process. *Neural Computation* **7** (6), 1165–1177.

Neal, R. M. and G. E. Hinton (1998). A new view of the EM algorithm that justifies incremental and other variants. In M. I. Jordan (Ed.), *Learning in Graphical Models.* Kluwer.

Pajunen, P. and J. Karhunen (1997). A maximum likelihood approach to nonlinear blind source separation. In *Proceedings 1997 International Conference on Artificial Neural Networks, ICANN'97*, pp. 541–546. Lausanne, Switzerland.

Pearlmutter, B. A. and L. C. Parra (1996). A context-sensitive generalization of ICA. In *International Conference on Neural Information Processing.*

Poggio, T. and F. Girosi (1990). Networks for approximation and learning. *Proceedings of the IEEE* **78** (9), 1481–1497.

Samard, T. and S. A. Harp (1992). Self-organization with partial data. *Network: Computation in Neural Systems* **3** (2), 205–212.

Tibshirani, R. (1992). Principal curves revisited. *Statistics and Computing* **2**, 183–190.

Tipping, M. E. and C. M. Bishop (1997a). Mixtures of principal component analysers. In *Proceedings IEE Fifth International Conference on Artificial Neural Networks, Cambridge, U.K., July.*, pp. 13–18.

Tipping, M. E. and C. M. Bishop (1997b). Mixtures of probabilistic principal component analysers. Technical Report NCRG/97/003, Neural Computing Research Group, Aston University, Birmingham, UK. Accepted for publication in Neural Computation.

Tipping, M. E. and C. M. Bishop (1997c). Probabilistic principal component analysis. Technical report, Neural Computing Research Group, Aston University, Birmingham, UK. Accepted for publication in the Royal Statistical Society, B.

Utsugi, A. (1997). Hyperparameter selection for self-organizing maps. *Neural Computation* **9** (3), 623–635.

Whittle, P. (1963). *Prediction and Regulation by Linear Least-square Methods.* English Universities Press.

Williams, C. K. I. and C. E. Rasmussen (1996). Gaussian processes for regression. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems*, Volume 8, pp. 514–520. MIT Press.

Williams, C. K. I., M. D. Revow, and G. E. Hinton (1993). Hand-printed digit recogntion using deformable models. In L. Harris and M. Jenkin (Eds.), *Spatial Vision in Humans and Robots.* Cambridge University Press.

Yaglom, A. M. (1987). *Correlation Theory of Stationary and Related Random Functions Volume I:Basic Results.* Springer-Verlag.