LATENT VARIABLE MODELS

CHRISTOPHER M. BISHOP

Microsoft Research
7 J. J. Thomson Avenue,
Cambridge CB3 0FB, U.K.

Published in *Learning in Graphical Models,* M. I. Jordan (Ed.),
MIT Press (1999), 371–403.

Abstract. A powerful approach to probabilistic modelling involves supplementing a set of observed variables with additional latent, or hidden, variables. By defining a joint distribution over visible and latent variables, the corresponding distribution of the observed variables is then obtained by marginalization. This allows relatively complex distributions to be expressed in terms of more tractable joint distributions over the expanded variable space. One well-known example of a hidden variable model is the mixture distribution in which the hidden variable is the discrete component label. In the case of continuous latent variables we obtain models such as factor analysis. The structure of such probabilistic models can be made particularly transparent by giving them a graphical representation, usually in terms of a directed acyclic graph, or Bayesian network. In this chapter we provide an overview of latent variable models for representing continuous variables. We show how a particular form of linear latent variable model can be used to provide a *probabilistic* formulation of the well-known technique of principal components analysis (PCA). By extending this technique to mixtures, and hierarchical mixtures, of probabilistic PCA models we are led to a powerful interactive algorithm for data visualization. We also show how the probabilistic PCA approach can be generalized to non-linear latent variable models leading to the Generative Topographic Mapping algorithm (GTM). Finally, we show how GTM can itself be extended to model temporal data.

1. Density Modelling

One of the central problems in pattern recognition and machine learning is that of density estimation, in other words the construction of a model of a probability distribution given a finite sample of data drawn from that distribution. Throughout this chapter we will consider the problem of modelling the distribution of a set of continuous variables $t_1, \ldots, t_d$ which we will collectively denote by the vector $\mathbf{t}$.

A standard approach to the problem of density estimation involves parametric models in which a specific form for the density is proposed which contains a number of adaptive parameters. Values for these parameters are then determined from an observed data set $D = \{\mathbf{t}_1, \ldots, \mathbf{t}_N\}$ consisting of N data vectors. The most widely used parametric model is the normal, or Gaussian, distribution given by

$$p(\mathbf{t}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-d/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{t}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\mathbf{t}-\boldsymbol{\mu})^{\mathrm{T}}\right\} \tag{1}$$

where $\boldsymbol{\mu}$ is the mean, $\boldsymbol{\Sigma}$ is the covariance matrix, and $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$. One technique for setting the values of these parameters is that of maximum likelihood which involves consideration of the log probability of the observed data set given the parameters, i.e.

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \ln p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln p(\mathbf{t}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{2}$$

in which it is assumed that the data vectors $\mathbf{t}_n$ are drawn independently from the distribution. When viewed as a function of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, the quantity $p(D|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is called the *likelihood* function. Maximization of the likelihood (or equivalently the log likelihood) with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ leads to the set of parameter values which are most likely to have given rise to the observed data set. For the normal distribution (1) the log likelihood (2) can be maximized analytically, leading to the intuitive result [1] that the maximum likelihood solutions $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ are given by

$$\widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{t}_n \tag{3}$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{t}_n - \widehat{\boldsymbol{\mu}})(\mathbf{t}_n - \widehat{\boldsymbol{\mu}})^{\mathrm{T}} \tag{4}$$

corresponding to the sample mean and sample covariance respectively.

As an alternative to maximum likelihood, we can define priors over $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ use Bayes' theorem, together with the observed data, to determine the posterior distribution. An introduction to Bayesian inference for the normal distribution is given in [5].

While the simple normal distribution (1) is widely used, it suffers from some significant limitations. In particular, it can often prove to be too flexible in that the number of independent parameters in the model can be excessive. This problem is addressed through the introduction of continuous latent variables. On the other hand, the normal distribution can also be insufficiently flexible since it can only represent uni-modal distributions. A more general family of distributions can be obtained by considering mixtures of Gaussians, corresponding to the introduction of a discrete latent variable. We consider each of these approaches in turn.

1.1. LATENT VARIABLES

Consider the number of free parameters in the normal distribution (1). Since $\mathbf{\Sigma}$ is symmetric, it contains $d(d+1)/2$ independent parameters. There are a further d independent parameters in $\boldsymbol{\mu}$, making $d(d+3)/2$ parameters in total. For large d this number grows like d^2, and excessively large numbers of data points may be required to ensure that the maximum likelihood solution for $\mathbf{\Sigma}$ is well determined. One way to reduce the number of free parameters in the model is to consider a diagonal covariance matrix, which has just d free parameters. This, however, corresponds to a very strong assumption, namely that the components of $\mathbf{t}$ are statistically independent, and such a model is therefore unable to capture the correlations between different components.

We now show how the number of degrees of freedom within the model can be controlled, while still allowing correlations to be captured, by introducing *latent* (or 'hidden') variables. The goal of a latent variable model is to express the distribution $p(\mathbf{t})$ of the variables $t_1, \ldots, t_d$ in terms of a smaller number of latent variables $\mathbf{x} = (x_1, \ldots, x_q)$ where $q < d$. This is achieved by first decomposing the joint distribution $p(\mathbf{t}, \mathbf{x})$ into the product of the marginal distribution $p(\mathbf{x})$ of the latent variables and the conditional distribution $p(\mathbf{t}|\mathbf{x})$ of the data variables given the latent variables. It is often convenient to assume that the conditional distribution factorizes over the data variables, so that the joint distribution becomes

$$p(\mathbf{t}, \mathbf{x}) = p(\mathbf{x})p(\mathbf{t}|\mathbf{x}) = p(\mathbf{x}) \prod_{i=1}^{d} p(t_i|\mathbf{x}). \tag{5}$$

This factorization property can be expressed graphically in terms of a Bayesian network, as shown in Figure 1.

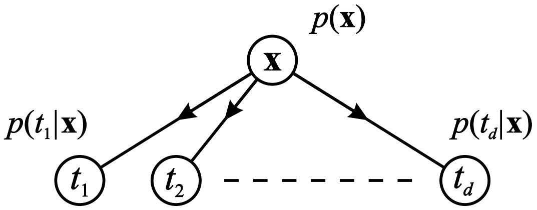

Figure 1. Bayesian network representation of the latent variable distribution given by (5), in which the data variables $t_1, \ldots, t_d$ are independent given the latent variables $\mathbf{x}$.

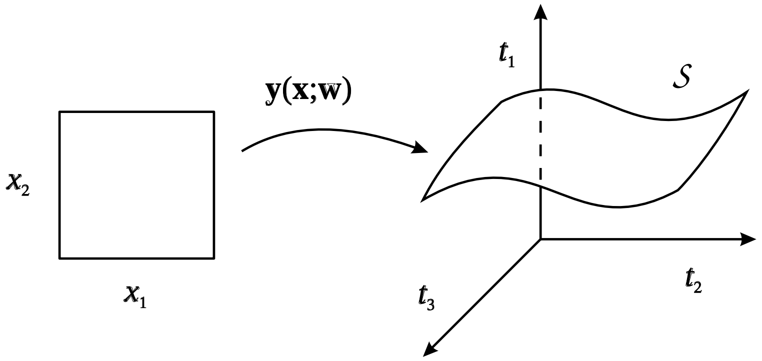

Figure 2. The non-linear function $\mathbf{y}(\mathbf{x}; \mathbf{w})$ defines a manifold $\mathcal{S}$ embedded in data space given by the image of the latent space under the mapping $\mathbf{x} \to \mathbf{y}$.

We next express the conditional distribution $p(\mathbf{t}|\mathbf{x})$ in terms of a mapping from latent variables to data variables, so that

$$\mathbf{t} = \mathbf{y}(\mathbf{x}; \mathbf{w}) + \mathbf{u} \tag{6}$$

where $\mathbf{y}(\mathbf{x}; \mathbf{w})$ is a function of the latent variable $\mathbf{x}$ with parameters $\mathbf{w}$, and $\mathbf{u}$ is an $\mathbf{x}$-independent noise process. If the components of $\mathbf{u}$ are uncorrelated, the conditional distribution for $\mathbf{t}$ will factorize as in (5). Geometrically the function $\mathbf{y}(\mathbf{x}; \mathbf{w})$ defines a manifold in data space given by the image of the latent space, as shown in Figure 2.

The definition of the latent variable model is completed by specifying the distribution $p(\mathbf{u})$, the mapping $\mathbf{y}(\mathbf{x}; \mathbf{w})$, and the marginal distribution $p(\mathbf{x})$. As we shall see later, it is often convenient to regard $p(\mathbf{x})$ as a *prior* distribution over the latent variables.

The desired model for the distribution $p(\mathbf{t})$ of the data is obtained by marginalizing over the latent variables

$$p(\mathbf{t}) = \int p(\mathbf{t}|\mathbf{x})p(\mathbf{x})\, d\mathbf{x}. \tag{7}$$

This integration will, in general, be analytically intractable except for specific forms of the distributions $p(\mathbf{t}|\mathbf{x})$ and $p(\mathbf{x})$.

One of the simplest latent variable models is called *factor analysis* [3, 4] and is based on a linear mapping $\mathbf{y}(\mathbf{x};\mathbf{w})$ so that

$$\mathbf{t} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \mathbf{u}, \tag{8}$$

in which $\mathbf{W}$ and $\boldsymbol{\mu}$ are adaptive parameters. The distribution $p(\mathbf{x})$ is chosen to be a zero-mean unit covariance Gaussian distribution $\mathcal{N}(\mathbf{0},\mathbf{I})$, while the noise model for $\mathbf{u}$ is also a zero mean Gaussian with a covariance matrix $\boldsymbol{\Psi}$ which is diagonal. Using (7) it is easily shown that the distribution $p(\mathbf{t})$ is also Gaussian, with mean $\boldsymbol{\mu}$ and a covariance matrix given by $\boldsymbol{\Psi} + \mathbf{W}\mathbf{W}^{\mathrm{T}}$.

The parameters of the model, comprising $\mathbf{W}$, $\boldsymbol{\Psi}$ and $\boldsymbol{\mu}$, can again be determined by maximum likelihood. There is, however, no longer a closed-form analytic solution, and so their values must be determined by iterative procedures. For q latent variables, there are $q \times d$ parameters in $\mathbf{W}$ together with d in $\boldsymbol{\Psi}$ and d in $\boldsymbol{\mu}$. There is some redundancy between these parameters, and a more careful analysis shows that the number of independent degrees of freedom in this model is given by

$$(d+1)(q+1) - q(q+1)/2. \tag{9}$$

The number of independent parameters in this model therefore only grows linearly with d, and yet the model can still capture the dominant correlations between the data variables. We consider the nature of such models in more detail in Section 2.

1.2. MIXTURE DISTRIBUTIONS

The density models we have considered so far are clearly very limited in terms of the variety of probability distributions which they can model since they can only represent distributions which are uni-modal. However, they can form the basis of a very general framework for density modelling, obtained by considering probabilistic *mixtures* of M simpler parametric distributions. This leads to density models of the form

$$p(\mathbf{t}) = \sum_{i=1}^{M} \pi_i p(\mathbf{t}|i) \tag{10}$$

in which the $p(\mathbf{t}|i)$ represent the individual components of the mixture and might consist, for example, of normal distributions of the form (1) each with its own independent mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. The parameters π_i in (10) are called *mixing coefficients* and satisfy the requirements $0 \leq \pi_i \leq 1$ and $\sum_i \pi_i = 1$ so that $p(\mathbf{t})$ will be non-negative and will integrate to unity (assuming the individual component densities also have these properties). We can represent the mixture distribution (10) as a simple Bayesian network, as shown in Figure 3.

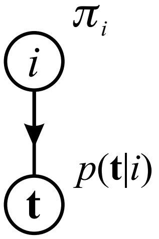

Figure 3. Bayesian network representation of a simple mixture distribution.

The mixing coefficients can be interpreted as *prior* probabilities for the values of the label i. For a given data point $\mathbf{t}_n$ we can then use Bayes' theorem to evaluate the corresponding posterior probabilities, given by

$$R_{ni} \equiv p(i|\mathbf{t}_n) = \frac{\pi_i p(\mathbf{t}_n|i)}{\sum_j \pi_j p(\mathbf{t}_n|j)}. \tag{11}$$

The value of $p(i|\mathbf{t}_n)$ can be regarded as the *responsibility* which component i takes for 'explaining' data point $\mathbf{t}_n$. Effectively this is using Bayes' theorem to reverse the direction of the arrow in Figure 3.

The log likelihood for the mixture distribution takes the form

$$\mathcal{L}(\{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}) = \sum_{n=1}^{N} \ln \left\{ \sum_{i=1}^{M} \pi_i p(\mathbf{t}|i) \right\}. \tag{12}$$

Maximization of this log likelihood is more complex than for a single component due to the presence of the sum inside the logarithm. An elegant and powerful technique for performing this optimization called the expectation-maximization (EM) algorithm [11], and an introductory account of EM in the context of mixture distributions is given in [5]. The EM algorithm is based on the observation that, if we were given a set of indicator variables z_{ni} specifying which component i was responsible for generating each data point $\mathbf{t}_n$, then the log likelihood would take the form

$$\mathcal{L}_{\text{comp}}(\{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}) = \sum_{n=1}^{N} \sum_{i=1}^{M} z_{ni} \ln \{\pi_i p(\mathbf{t}|i)\} \tag{13}$$

and its optimization would be straightforward, with the result that each component is fitted independently to the corresponding group of data points, and the mixing coefficients are given by the fractions of points in each group.

The $\{z_{ni}\}$ are regarded as 'missing data', and the data set $\{\mathbf{t}_n\}$ is said to be 'incomplete'. Combining $\{\mathbf{t}_n\}$ and $\{z_{ni}\}$ we obtain the corresponding 'complete' data set, with a log likelihood given by (13). Of course, the values of $\{z_{ni}\}$ are unknown, but their posterior distribution can be computed using Bayes' theorem, and the expectation of z_{ni} under this distribution is just the set of responsibilities R_{ni} given by (11). The EM algorithm is based on the maximization of the expected complete-data log likelihood given from (13) by

$$\langle \mathcal{L}_{\mathrm{comp}}(\{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\})\rangle = \sum_{n=1}^{N} \sum_{i=1}^{M} R_{ni} \ln \{\pi_i p(\mathbf{t}|i)\}\,. \tag{14}$$

It alternates between the E-step, in which the R_{ni} are evaluated using (11), and the M-step in which (14) is maximized with respect to the model parameters to give a revised set of parameter values. At each cycle of the EM algorithm the true log likelihood is guaranteed to increase unless it is already at a local maximum [11].

The EM algorithm can also be applied to the problem of maximizing the likelihood for a single latent variable model of the kind discussed in Section 1.1. We note that the log likelihood for such a model takes the form

$$\mathcal{L}(\mathbf{W}, \boldsymbol{\mu}, \boldsymbol{\Psi}) = \sum_{n=1}^{N} \ln p(\mathbf{t}_n) = \sum_{n=1}^{N} \ln \left\{ \int p(\mathbf{t}_n|\mathbf{x}_n) p(\mathbf{x}_n)\, d\mathbf{x}_n \right\}. \tag{15}$$

Again, this is difficult to treat because of the integral inside the logarithm. In this case the values of $\mathbf{x}_n$ are regarded as the missing data. Given the prior distribution $p(\mathbf{x})$ we can consider the corresponding posterior distribution obtained through Bayes' theorem

$$p(\mathbf{x}_n|\mathbf{t}_n) = \frac{p(\mathbf{t}_n|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{t}_n)} \tag{16}$$

and the sufficient statistics for this distribution are evaluated in the E-step. The M-step involves maximization of the expected complete-data log likelihood and is generally much simpler than the direct maximization of the true log likelihood. For simple models such as the factor analysis model discussed in Section 1.1 this maximization can be performed analytically. The EM (expectation-maximization) algorithm for maximizing the likelihood function for standard factor analysis was derived by Rubin and Thayer [23].

We can combine the technique of mixture modelling with that of latent variables, and consider a mixture of latent-variable models. The corresponding Bayesian network is shown in Figure 4. Again, the EM algorithm provides a natural framework for determination of the model parameters, and allows both the values of the component label i and of the latent variable $\mathbf{x}$ to be treated together as missing data.

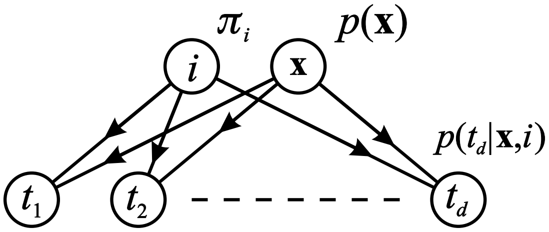

Figure 4. Bayesian network representation of a mixture of latent variable models. Given the values of i and $\mathbf{x}$, the variables $t_1, \ldots, t_d$ are conditionally independent.

In the subsequent sections of this chapter we shall see how the concepts of latent variables and mixture distributions can be used in a fruitful partnership to obtain a range of powerful algorithms for density modelling, pattern classification and data visualization.

2. Probabilistic Principal Component Analysis

Principal component analysis is a well-established technique for dimensionality reduction, and a chapter on the subject may be found in practically every text on multivariate analysis. Examples of its many applications include data compression, image processing, data visualization, exploratory data analysis, and pattern recognition.

The most common derivation of PCA is in terms of a standardized linear projection which maximizes the variance in the projected space [14]. For a set of observed d-dimensional data vectors $\{\mathbf{t}_n\}$, $n \in \{1 \ldots N\}$, the q *principal axes* $\mathbf{v}_j$, $j \in \{1, \ldots, q\}$, are those orthonormal axes onto which the retained variance under projection is maximal. It can be shown that the vectors $\mathbf{v}_j$ are given by the q dominant eigenvectors (i.e. those with the largest associated eigenvalues λ_j) of the sample covariance matrix

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{t}_n - \widehat{\boldsymbol{\mu}})(\mathbf{t}_n - \widehat{\boldsymbol{\mu}})^{\mathrm{T}} \tag{17}$$

such that $\mathbf{S}\mathbf{v}_j = \lambda_j \mathbf{v}_j$. Here $\widehat{\boldsymbol{\mu}}$ is the sample mean, given by (3). The q principal components of the observed vector $\mathbf{t}_n$ are given by the vector

$\mathbf{u}_n = \mathbf{V}^{\mathrm{T}}(\mathbf{t}_n - \widehat{\boldsymbol{\mu}})$, where $\mathbf{V}^{\mathrm{T}} = (\mathbf{v}_1, \ldots, \mathbf{v}_q)^{\mathrm{T}}$, in which the variables u_j are decorellated such that the covariance matrix for $\mathbf{u}$ is diagonal with elements $\{\lambda_j\}$.

A complementary property of PCA, and that most closely related to the original discussions of Pearson [20], is that, of all orthogonal linear projections $\mathbf{x}_n = \mathbf{V}^{\mathrm{T}}(\mathbf{t}_n - \widehat{\boldsymbol{\mu}})$, the principal component projection minimizes the squared reconstruction error $\sum_n \|\mathbf{t}_n - \hat{\mathbf{t}}_n\|^2$, where the optimal linear reconstruction of $\mathbf{t}_n$ is given by $\hat{\mathbf{t}}_n = \mathbf{V}\mathbf{x}_n + \widehat{\boldsymbol{\mu}}$.

One serious disadvantage of both these definitions of PCA is the absence of a probability density model and associated likelihood measure. Deriving PCA from the perspective of density estimation would offer a number of important advantages, including the following:

- The corresponding likelihood measure would permit comparison with other density–estimation techniques and would facilitate statistical testing.
- Bayesian inference methods could be applied (e.g. for model comparison) by combining the likelihood with a prior.
- If PCA were used to model the class–conditional densities in a classification problem, the posterior probabilities of class membership could be computed.
- The value of the probability density function would give a measure of the novelty of a new data point.
- The single PCA model could be extended to a mixture of such models.

In this section we review the key result of Tipping and Bishop [25], which shows that principal component analysis may indeed be obtained from a probability model. In particular we show that the maximum-likelihood estimator of $\mathbf{W}$ in (8) for a specific form of latent variable models is given by the matrix of (scaled and rotated) principal axes of the data.

2.1. RELATIONSHIP TO LATENT VARIABLES

Links between principal component analysis and latent variable models have already been noted by a number of authors. For instance Anderson [2] observed that principal components emerge when the data is assumed to comprise a systematic component, plus an independent error term for each variable having common variance σ^2. Empirically, the similarity between the columns of $\mathbf{W}$ and the principal axes has often been observed in situations in which the elements of $\boldsymbol{\Psi}$ are approximately equal [22]. Basilevsky [4] further notes that when the model $\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$ is exact, and therefore equal to $\mathbf{S}$, the matrix $\mathbf{W}$ is identifiable and can be determined analytically through eigen-decomposition of $\mathbf{S}$, without resort to iteration.

As well as assuming that the model is exact, such observations do not consider the maximum-likelihood context. By considering a particular case of the factor analysis model in which the noise covariance is isotropic so that $\mathbf{\Psi} = \sigma^2\mathbf{I}$, we now show that even when the data covariance matrix cannot be expressed exactly using the form $\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$, the maximum-likelihood estimator $\mathbf{W}_{\mathrm{ML}}$ is that matrix whose columns are the scaled and rotated principal eigenvectors of the sample covariance matrix $\mathbf{S}$ [25]. An important consequence of this derivation is that PCA may be expressed in terms of a probability density model, which we shall refer to as probabilistic principal component analysis (PPCA).

2.2. THE PROBABILITY MODEL

For the isotropic noise model $\mathbf{u} \sim N(\mathbf{0}, \boldsymbol{\sigma^2}\mathbf{I})$, equations (6) and (8) imply a probability distribution over $\mathbf{t}$-space for a given $\mathbf{x}$ given by

$$p(\mathbf{t}|\mathbf{x}) = (2\pi\sigma^2)^{-d/2} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{t} - \mathbf{W}\mathbf{x} - \boldsymbol{\mu}\|^2\right\}. \tag{18}$$

In the case of an isotropic Gaussian prior over the latent variables defined by

$$p(\mathbf{x}) = (2\pi)^{-q/2} \exp\left\{-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{x}\right\} \tag{19}$$

we then obtain the marginal distribution of $\mathbf{t}$ in the form

$$\begin{aligned} p(\mathbf{t}) &= \int p(\mathbf{t}|\mathbf{x})p(\mathbf{x})d\mathbf{x} && (20) \\ &= (2\pi)^{-d/2}|\mathbf{C}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{t} - \boldsymbol{\mu})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{t} - \boldsymbol{\mu})\right\} && (21) \end{aligned}$$

where the model covariance is

$$\mathbf{C} = \sigma^2\mathbf{I} + \mathbf{W}\mathbf{W}^{\mathrm{T}}. \tag{22}$$

Using Bayes' theorem, the *posterior* distribution of the latent variables $\mathbf{x}$ given the observed $\mathbf{t}$ is given by

$$\begin{aligned} p(\mathbf{x}|\mathbf{t}) = {} & (2\pi)^{-q/2}|\sigma^2\mathbf{M}|^{-1/2} \times \\ & \exp\left\{-\frac{1}{2}(\mathbf{x} - \langle\mathbf{x}\rangle)^{\mathrm{T}}(\sigma^2\mathbf{M})^{-1}(\mathbf{x} - \langle\mathbf{x}\rangle)\right\} \end{aligned} \tag{23}$$

where the posterior covariance matrix is given by

$$\sigma^2\mathbf{M} = \sigma^2(\sigma^2\mathbf{I} + \mathbf{W}^{\mathrm{T}}\mathbf{W})^{-1} \tag{24}$$

and the mean of the distribution is given by

$$\langle \mathbf{x} \rangle = \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{t} - \boldsymbol{\mu}). \tag{25}$$

Note that $\mathbf{M}$ has dimension $q \times q$ while $\mathbf{C}$ has dimension $d \times d$.

The log-likelihood for the observed data under this model is given by

$$\begin{aligned}\mathcal{L} &= \sum_{n=1}^{N} \ln\{p(\mathbf{t}_n)\} \\ &= -\frac{Nd}{2}\ln(2\pi) - \frac{N}{2}\ln|\mathbf{C}| - \frac{N}{2}\mathrm{Tr}\left\{\mathbf{C}^{-1}\mathbf{S}\right\}\end{aligned} \tag{26}$$

where the sample covariance matrix $\mathbf{S}$ of the observed $\{\mathbf{t}_n\}$ is given by (17).

In principle, we could determine the parameters for this model by maximizing the log-likelihood $\mathcal{L}$ using the EM algorithm of Rubin and Thayer [23]. However, we now show that, for the case of an isotropic noise covariance of the form we are considering, there is an exact analytical solution for the model parameters.

2.3. PROPERTIES OF THE MAXIMUM-LIKELIHOOD SOLUTION

Our key result is that the log-likelihood (26) is maximized when the columns of $\mathbf{W}$ span the principal subspace of the data. To show this we consider the derivative of (26) with respect to $\mathbf{W}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = N(\mathbf{C}^{-1}\mathbf{S}\mathbf{C}^{-1}\mathbf{W} - \mathbf{C}^{-1}\mathbf{W}) \tag{27}$$

which may be obtained from standard matrix differentiation results (see [19], pp 133). In [25] it is shown that, with $\mathbf{C}$ given by (22), the only non-zero stationary points of (27) occur for:

$$\mathbf{W} = \mathbf{U}_q(\boldsymbol{\Lambda}_q - \sigma^2\mathbf{I})^{1/2}\mathbf{R} \tag{28}$$

where the q column vectors in $\mathbf{U}_q$ are eigenvectors of $\mathbf{S}$, with corresponding eigenvalues in the diagonal matrix $\boldsymbol{\Lambda}_q$, and $\mathbf{R}$ is an arbitrary $q \times q$ orthogonal rotation matrix. Furthermore, it is also shown that the stationary point corresponding to the *global maximum* of the likelihood occurs when $\mathbf{U}_q$ comprises the *principal* eigenvectors of $\mathbf{S}$ (i.e. the eigenvectors corresponding to the q largest eigenvalues) and that all other combinations of eigenvectors represent saddle-points of the likelihood surface. Thus, from (28), the columns of the maximum-likelihood estimator $\mathbf{W}_{\mathrm{ML}}$ contain the principal eigenvectors of $\mathbf{S}$, with scalings determined by the corresponding eigenvalues together with the parameter σ^2, and with arbitrary rotation.

It may also be shown that for $\mathbf{W} = \mathbf{W}_{\mathrm{ML}}$, the maximum-likelihood estimator for σ^2 is given by

$$\sigma^2_{\mathrm{ML}} = \frac{1}{d-q} \sum_{j=q+1}^{d} \lambda_j \tag{29}$$

which has a clear interpretation as the variance 'lost' in the projection, averaged over the lost dimensions. Note that the columns of $\mathbf{W}_{\mathrm{ML}}$ are not orthogonal since

$$\mathbf{W}_{\mathrm{ML}}^{\mathrm{T}} \mathbf{W}_{\mathrm{ML}} = \mathbf{R}^{\mathrm{T}} (\mathbf{\Lambda}_q - \sigma^2 \mathbf{I}) \mathbf{R}, \tag{30}$$

which in general is not diagonal. However, the columns of $\mathbf{W}$ will be orthogonal for the particular choice $\mathbf{R} \neq \mathbf{I}$.

In summary, we can obtain a probabilistic principal components model by finding the q principal eigenvectors and eigenvalues of the sample covariance matrix. The density model is then given by a Gaussian distribution with mean $\boldsymbol{\mu}$ given by the sample mean, and a covariance matrix $\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2 \mathbf{I}$ in which $\mathbf{W}$ is given by (28) and σ^2 is given by (29).

3. Mixtures of Probabilistic PCA

We now extend the latent variable model of Section 2 by considering a mixture of probabilistic principal component analysers [24], in which the model distribution is given by (10) with component densities given by (22). It is straightforward to obtain an EM algorithm to determine the parameters π_i, $\boldsymbol{\mu}_i$, $\mathbf{W}_i$ and σ_i^2. The E-step of the EM algorithm involves the use of the current parameter estimates to evaluate the responsibilities of the mixture components i for the data points $\mathbf{t}_n$, given from Bayes' theorem by

$$R_{ni} = \frac{p(\mathbf{t}_n|i)\pi_i}{p(\mathbf{t}_n)}. \tag{31}$$

In the M-step, the mixing coefficients and component means are re-estimated using

$$\begin{aligned} \widetilde{\pi}_i &= \frac{1}{N} \sum_{n=1}^{N} R_{ni} && (32) \\ \widetilde{\boldsymbol{\mu}}_i &= \frac{\sum_{n=1}^{N} R_{ni} \mathbf{t}_n}{\sum_{n=1}^{N} R_{ni}} && (33) \end{aligned}$$

while the parameters $\mathbf{W}_i$ and σ_i^2 are obtained by first evaluating the weighted covariance matrices given by

$$\mathbf{S}_i = \frac{\sum_{n=1}^{N} R_{ni} (\mathbf{t}_n - \widetilde{\boldsymbol{\mu}})(\mathbf{t}_n - \widetilde{\boldsymbol{\mu}})^{\mathrm{T}}}{\sum_{n=1}^{N} R_{ni}} \tag{34}$$

and then applying (28) and (29).

3.1. EXAMPLE APPLICATION: HAND-WRITTEN DIGIT CLASSIFICATION

One potential application for high-dimensional density models is handwritten digit recognition. Examples of gray-scale pixel images of a given digit will generally lie on a lower-dimensional smooth continuous manifold, the geometry of which is determined by properties of the digit such as rotation, scaling and thickness of stroke. One approach to the classification of such digits (although not necessarily the best) is to build a model of each digit separately, and classify unseen digits according to the model to which they are most 'similar'.

Hinton *et al.* [12] discussed the problem of handwritten digit problem, and applied a 'mixture' of conventional PCA models, using soft reconstruction-based clustering, to the classification of scaled and smoothed 8-by-8 gray-scale images taken from the CEDAR U.S. postal service database [15]. The models were constructed using an 11,000-digit subset of the 'br' data set (which was further split into training and validation sets), and the 'bs' test set was classified according to which model best reconstructed each digit. We repeated the experiment with the same data using the probabilistic PCA mixture approach utilizing the same choice of parameter values ($M = 10$ and $q = 10$). The same method of classification was used, and the best model on the validation set misclassified 4.64% of the digits in the test set, while Hinton *et al.* [12] reported an error of 4.91%. We would expect the improvement to be a result partly of the localized clustering of the PPCA model, but also the use of individually-estimated values of σ_i^2 for each component, rather than a single, arbitrarily-chosen, global value used in [12].

One of the advantages of the PPCA methodology is that the definition of the density model permits the posterior probabilities of class membership to be computed for each digit and utilized for subsequent classification. After optimizing the parameters M and q for each model to obtain the best performance on the validation set, the model misclassified 4.61% of the test set. An advantage of the use of posterior probabilities is that it is possible to reject (using an optimal criterion) a proportion of the test samples about which the classifier is most 'unsure', and thus improve the classification performance on the remaining data. Using this approach to reject 5% of the test examples resulted in a misclassification rate of 2.50%.

4. Hierarchical Mixtures for Data Visualization

An interesting application for the PPCA model, and mixtures of PPCA models, is to the problem of data visualization. By considering a further extension to a hierarchical mixture model, we are led to a powerful interactive algorithm for visualization which retains a probabilistic framework and which can provide considerable insight into the structure of data in spaces of high dimensionality [10].

4.1. VISUALIZATION USING PROBABILISTIC PCA

Consider first the use of a single PPCA model for data visualization. In standard principal component analysis, the data points are visualized by orthogonal projection onto the principal components plane (spanned by the two leading eigenvectors). For our probabilistic PCA model this projection is modified slightly. From (23) and (25) it may be seen that the *posterior mean* projection of $\mathbf{t}_n$ is given by $\langle \mathbf{x}_n \rangle = \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{t}_n - \widehat{\boldsymbol{\mu}})$. When $\sigma^2 \to 0$, $\mathbf{M}^{-1} \to (\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-1}$ and $\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}$ then becomes an orthogonal projection, and so PCA is recovered (although the density model then becomes singular, and thus undefined). For $\sigma^2 > 0$, the projection onto the manifold is shrunk towards the origin as a result of the prior over $\mathbf{x}$. Because of this, $\mathbf{W}\langle \mathbf{x}_n \rangle$ is *not* an orthogonal projection of $\mathbf{t}_n$. We note, however, that information is not lost because of this shrinkage, since each data point may still be optimally reconstructed from the latent variable by taking the shrinkage into account. With $\mathbf{W} = \mathbf{W}_{\mathrm{ML}}$ the required reconstruction is given by

$$\hat{\mathbf{t}}_n = \mathbf{W}_{\mathrm{ML}}\{\mathbf{W}_{\mathrm{ML}}^{\mathrm{T}}\mathbf{W}_{\mathrm{ML}}\}^{-1}\mathbf{M}\langle \mathbf{x}_n \rangle, \tag{35}$$

and is derived in [25]. Thus the latent variables convey the necessary information to reconstruct the original data vector optimally, even in the case of $\sigma^2 > 0$.

The data set can therefore be visualized by mapping each data point onto the corresponding posterior mean $\langle \mathbf{x}_n \rangle$ in the two-dimensional latent space, as illustrated in Figure 5. Note that this type of visualization plot satisfies a topographic property in that points in data space which are sufficiently close will map to points in latent space which are also close.

We illustrate the visualization properties of this model using a toy data set consisting of 450 data points generated from a mixture of three Gaussians in three-dimensional space. Each Gaussian is relatively flat (has small variance) in one dimension, and two of these clusters are closely spaced with their principal planes parallel to each other, while the third is well separated from the first two. The structure of this data set has been chosen order to demonstrate the benefits of the interactive hierarchical approach developed in Section 4.3. A single two-dimensional latent variable model is

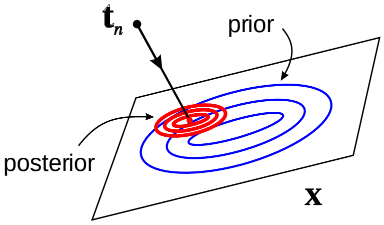

Figure 5. Illustration of the projection of a data vector $\mathbf{t}_n$ onto the point on the principal subspace corresponding to the posterior mean.

trained on this data set, and the result of plotting the posterior means of the data points is shown in Figure 6.

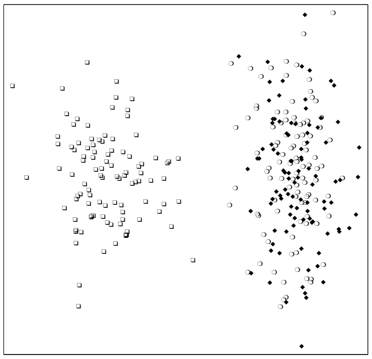

Figure 6. Plot of the posterior means of the data points from the toy data set, obtained from the probabilistic PCA model, indicating the presence of (at least) two distinct clusters.

4.2. MIXTURE MODELS FOR DATA VISUALIZATION

Next we consider the application of a simple mixture of PPCA models to data visualization. Once a mixture of probabilistic PCA models has been fitted to the data set, the procedure for visualizing the data points involves plotting each data point $\mathbf{t}_n$ on each of the two-dimensional latent spaces at the corresponding posterior mean position $\langle \mathbf{x}_{ni} \rangle$ given by

$$\langle \mathbf{x}_{ni} \rangle = (\mathbf{W}_i^{\mathrm{T}} \mathbf{W}_i + \sigma_i^2 \mathbf{I})^{-1} \mathbf{W}_i^{\mathrm{T}} (\mathbf{t}_n - \boldsymbol{\mu}_i) \tag{36}$$

as illustrated in Figure 7.

As a further refinement, the density of 'ink' for each data point $\mathbf{t}_n$ is weighted by the corresponding responsibility R_{ni} of model i for that data point, so that the total density of 'ink' is distributed by a partition of

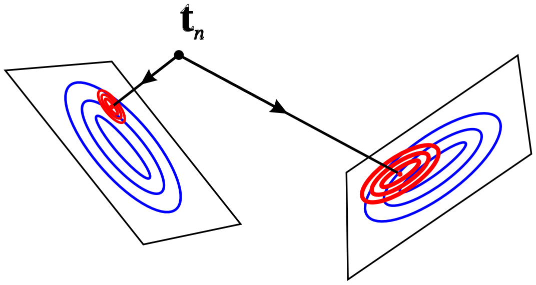

Figure 7. Illustration of the projection of a data vector onto two principal surfaces in a probabilistic PCA mixture model.

unity across the plots. Thus, each data point is plotted on every component model projection, while if a particular model takes nearly all of the posterior probability for a particular data point, then that data point will effectively be visible only on the corresponding latent space plot.

We shall regard the single PPCA plot introduced in Section 4.1 as the top level in a hierarchical visualization model, in which the mixture model forms the second level. Extensions to further levels of the hierarchy will be developed in Section 4.3.

The model can be extended to provide an interactive data exploration tool as follows. On the basis of the single top-level plot the user decides on an appropriate number of models to fit at the second level, and selects points $\mathbf{x}^{(i)}$ on the plot, corresponding, for example, to the centres of apparent clusters. The resulting points $\mathbf{y}^{(i)}$ in data space, obtained from $\mathbf{y}^{(i)} = \mathbf{W}\mathbf{x}^{(i)} + \boldsymbol{\mu}$, are then used to initialize the means $\boldsymbol{\mu}_i$ of the respective sub-models. To initialize the matrices $\mathbf{W}_i$ we first assign the data points to their nearest mean vector $\boldsymbol{\mu}_i$ and then compute the corresponding sample covariance matrices. This is a hard clustering analogous to K-means and represents an approximation to the posterior probabilities R_{ni} in which the largest posterior probability is replaced by 1 and the remainder by 0. For each of these clusters we then find the eigenvalues and eigenvectors of the sample covariance matrix and hence determine the probabilistic PCA density model. This initialization is then used as the starting point for the EM algorithm.

Consider the application of this procedure to the toy data set introduced in Section 4.1. At the top level we observed two apparent clusters, and so we might select a mixture of two models for the second level, with centres initialized somewhere near the centres of the two clusters seen at the top level. The result of fitting this mixture by EM leads to the two-level visualization plot shown in Figure 8.

The visualization process can be enhanced further by providing infor-

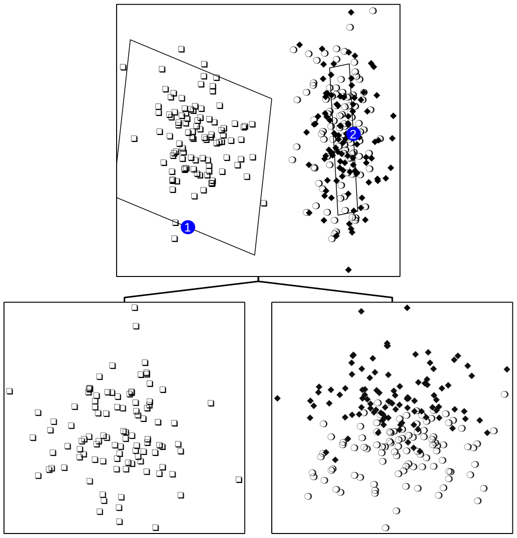

Figure 8. The result of applying the two-level visualization algorithm to the toy data set. At the second level a mixture of two latent variable models has been fitted and the data plotted on each latent space using the approach described in the text. In addition, the two latent planes have been visualized by projection back onto the top-level model. The left-hand plane at the second level is almost perpendicular to the top-level plane (as can be seen by its projection) giving further insight into why the two clusters which appear well separated on the left-hand second-level model appear to be overlapping at the top level.

mation at the top level on the location and orientation of the latent spaces corresponding to the second level, as shown in Figure 8. This is achieved by considering the orthogonal projection of the latent plane in data space onto the corresponding plane of the parent model.

4.3. HIERARCHICAL MIXTURE MODELS

We now extend the mixture representation of Section 1.2 to give a hierarchical mixture model. Our formulation will be quite general and can be applied to hierarchical mixtures of any parametric density. So far we have considered a two-level system consisting of a single latent variable model at the top level and a mixture of M_0 such models at the second level. We can now extend the hierarchy to a third level by associating a group $\mathcal{G}_i$ of latent variable models with each model i in the second level. The corresponding probability density can be written in the form

$$p(\mathbf{t}) = \sum_{i=1}^{M_0} \pi_i \sum_{j \in \mathcal{G}_i} \pi_{j|i} p(\mathbf{t}|i, j) \tag{37}$$

where $p(\mathbf{t}|i, j)$ again represent independent latent variable models, and $\pi_{j|i}$ correspond to sets of mixing coefficients, one set for each i, which satisfy $0 \leq \pi_{j|i} \leq 1$ and $\sum_j \pi_{j|i} = 1$. Thus each level of the hierarchy corresponds to a generative model, with lower levels giving more refined and detailed representations. This model is illustrated in Figure 9.

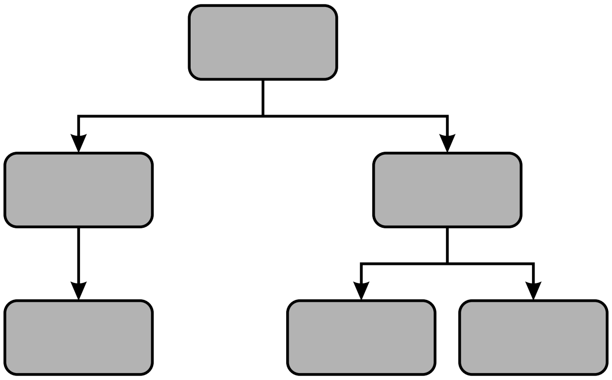

Figure 9. An example structure for the hierarchical mixture model.

Hierarchical mixtures of *conditional* density estimators were introduced by Jordan and Jacobs [16] in which all of the component distributions, as well as the various mixing coefficients, are conditioned on an observed 'input' vector. However, we are interested in hierarchical mixtures of unconditional density models. In this case a mixture of mixtures would be equivalent to a simple flat mixture and nothing would be gained from the hierarchy. In order to achieve the goal of hierarchical modelling we need to constrain the parameters of the model.

To see the appropriate form for the constraint, we note that the determination of the parameters of the models at the third level can again be viewed as a missing data problem in which the missing information corresponds to labels specifying which model generated each data point. When no information about the labels is provided the log likelihood for the model (37) would take the form

$$\mathcal{L} = \sum_{n=1}^{N} \ln \left\{ \sum_{i=1}^{M_0} \pi_i \sum_{j \in \mathcal{G}_i} \pi_{j|i} p(\mathbf{t}|i, j) \right\} \tag{38}$$

and the model would collapse to a simple mixture model. If, however, we were given a set of indicator variables z_{ni} specifying which model i at the second level generated each data point $\mathbf{t}_n$ then the log likelihood would become

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{i=1}^{M_0} z_{ni} \ln \left\{ \pi_i \sum_{j \in \mathcal{G}_i} \pi_{j|i} p(\mathbf{t}|i, j) \right\}. \tag{39}$$

In fact we only have partial, probabilistic, information in the form of the posterior responsibilities R_{ni} for each model i having generated the data points $\mathbf{t}_n$, obtained from the second level of the hierarchy. The corresponding log likelihood is obtained by taking the expectation of (39) with respect to the posterior distribution of the z_{ni} to give

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{i=1}^{M_0} R_{ni} \ln \left\{ \pi_i \sum_{j \in \mathcal{G}_i} \pi_{j|i} p(\mathbf{t}|i, j) \right\} \tag{40}$$

in which the R_{ni} are treated as constants. In the particular case in which the R_{ni} are all 0 or 1, corresponding to complete certainty about which model in the second level is responsible for each data point, the log likelihood (40) reduces to the form (39).

Maximization of (40) can again be performed using the EM algorithm, as shown in [10]. This has the same form as the EM algorithm for a simple mixture, discussed in Section 1.2, except that in the E-step, the posterior probability that model (i, j) generated data point $\mathbf{t}_n$ is given by

$$R_{ni,j} = R_{ni} R_{nj|i} \tag{41}$$

in which

$$R_{nj|i} = \frac{\pi_{j|i} p(\mathbf{t}_n|i, j)}{\sum_{j'} \pi_{j'|i} p(\mathbf{t}_n|i, j')}. \tag{42}$$

This result automatically satisfies the relation

$$\sum_{j \in \mathcal{G}_i} R_{ni,j} = R_{ni} \tag{43}$$

so that the responsibility of each model at the second level for a given data point n is shared by a partition of unity between the corresponding group of offspring models at the third level. It is straightforward to extend this hierarchical approach to any desired number of levels.

The result of applying this approach to the toy data set is shown in Figure 10.

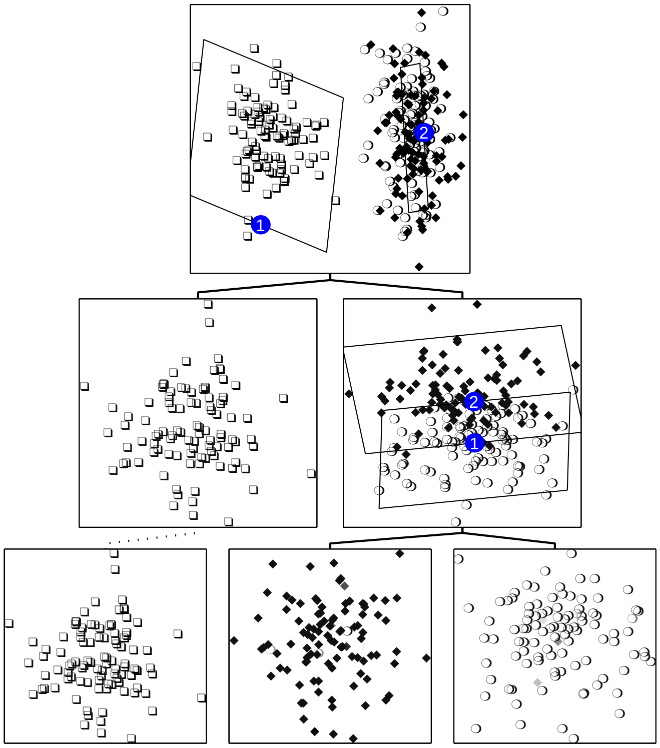

Figure 10. Plot of the complete three-level hierarchy for the toy data set. At the third level the three clusters have been almost perfectly separated. The structure of this particular hierarchical model is as shown in Figure 9.

4.4. EXAMPLE: OIL FLOW DATA

We now illustrate the application of the hierarchical visualization algorithm by considering an example data set arising from a non-invasive monitoring system used to determine the quantity of oil in a multi-phase pipeline containing a mixture of oil, water and gas [7]. The diagnostic data is collected from a set of three horizontal and three vertical beam-lines along which gamma rays at two different energies are passed. By measuring the degree of attenuation of the gammas, the fractional path length through oil and water (and hence gas) can readily be determined, giving 12 diagnostic measurements in total. In practice the aim is to solve the inverse problem of determining the fraction of oil in the pipe. The complexity of the problem arises from the possibility of the multi-phase mixture adopting one of a number of different geometrical configurations. Our goal is to visualize the structure of the data in the original 12-dimensional space. A data set consisting of 1000 points is obtained synthetically by simulating the physical processes in the pipe, including the presence of noise determined by photon statistics. Locally, the data is expected to have an intrinsic dimensionality of 2 corresponding to the 2 degrees of freedom given by the fraction of oil and the fraction of water (the fraction of gas being redundant). However, the presence of different configurations, as well as the geometrical interaction between phase boundaries and the beam paths, leads to numerous distinct clusters. It would appear that a hierarchical approach of the kind discussed here should be capable of discovering this structure. Results from fitting the oil flow data using a 3-level hierarchical model are shown in Figure 11.

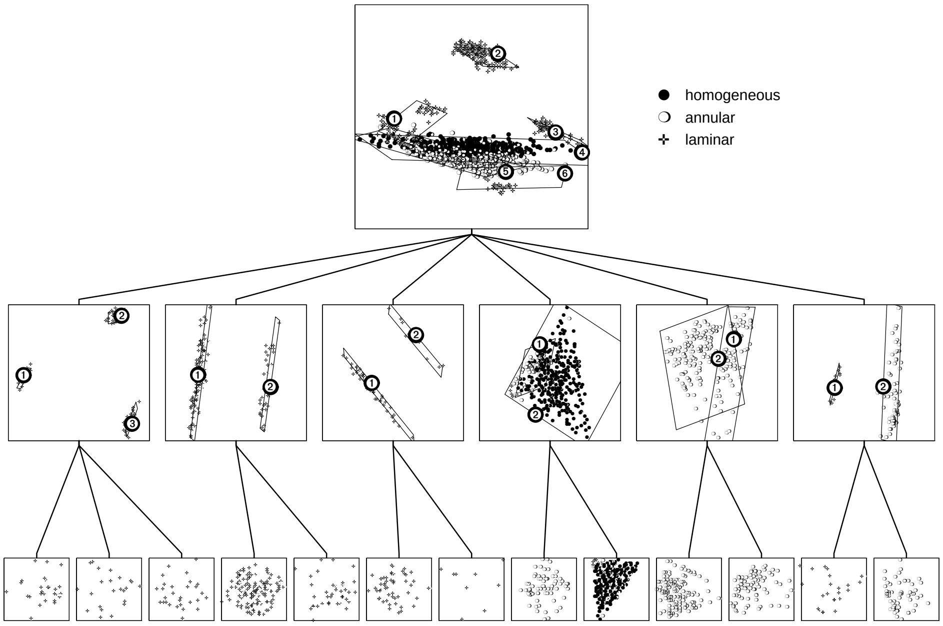

Figure 11. Results of fitting the oil data. The symbols denote different multi-phase flow configurations corresponding to homogeneous (●), annular (○) and laminar (+). Note, for example, how the apparently single cluster, number 2, in the top level plot is revealed to be two quite distinct clusters at the second level.

In the case of the toy data, the optimal choice of clusters and subclusters is relatively unambiguous and a single application of the algorithm is sufficient to reveal all of the interesting structure within the data. For more complex data sets, it is appropriate to adopt an exploratory perspective and investigate alternative hierarchies through the selection of differing numbers of clusters and their respective locations. The example shown in Figure 11 has clearly been highly successful. Note how the apparently single cluster, number 2, in the top-level plot is revealed to be two quite distinct clusters at the second level. Also, data points from the 'homogeneous' configuration have been isolated and can be seen to lie on a two-dimensional triangular structure in the third level. Inspection of the corresponding value of σ^2 confirms that this cluster is confined to a nearly planar sub-space, as expected from the physics of the diagnostic data for the homogeneous configurations.

5. Non-linear Models: The Generative Topographic Mapping

The latent variable models we have considered so far are based on a mapping from latent variables to data variables of the form (6) in which the function $\mathbf{y}(\mathbf{x};\mathbf{w})$ is linear in $\mathbf{x}$. Thus the manifold $\mathcal{S}$ in data space, shown in Figure 2 is a hyperplane. Data living on a manifold which is not hyperplanar (for example the hand-written digits data considered in Section 3.1) can then be approximated using a mixture of linear latent variable models. An alternative approach, however, would be to consider a latent variable model which is *non-linear*.

The difficulty with using a non-linear mapping function $\mathbf{y}(\mathbf{x};\mathbf{w})$ in (6) is that in general the integration over $\mathbf{x}$ in (7) will become analytically intractable. However, by making careful model choices a tractable, non-linear model, called the *Generative Topographic Mapping* or GTM, can be derived [9].

The central concept is to introduce a prior distribution $p(\mathbf{x})$ given by a sum of delta functions centred on the nodes of a regular grid in latent space

$$p(\mathbf{x}) = \frac{1}{K}\sum_{i=1}^{K}\delta(\mathbf{x}-\mathbf{x}_i) \tag{44}$$

in which case the integral in (7) can be performed analytically even for non-linear functions $\mathbf{y}(\mathbf{x};\mathbf{w})$. The conditional distribution $p(\mathbf{t}|\mathbf{x})$ is chosen to be an isotropic Gaussian with variance σ^2. (Note that this is easily generalized to deal with mixed continuous and categorical data by considering the corresponding product of Gaussian and multinomial distributions.) Each latent point $\mathbf{x}_i$ is then mapped to a corresponding point $\mathbf{y}(\mathbf{x}_i;\mathbf{w})$ in data space, which forms the centre of a Gaussian density function, as illustrated in Figure 12. From (7) and (44) we see that the distribution function in data space then takes the form

$$p(\mathbf{t}|\mathbf{W},\sigma^2) = \frac{1}{K}\sum_{i=1}^{K}p(\mathbf{t}|\mathbf{x}_i,\mathbf{W},\sigma^2) \tag{45}$$

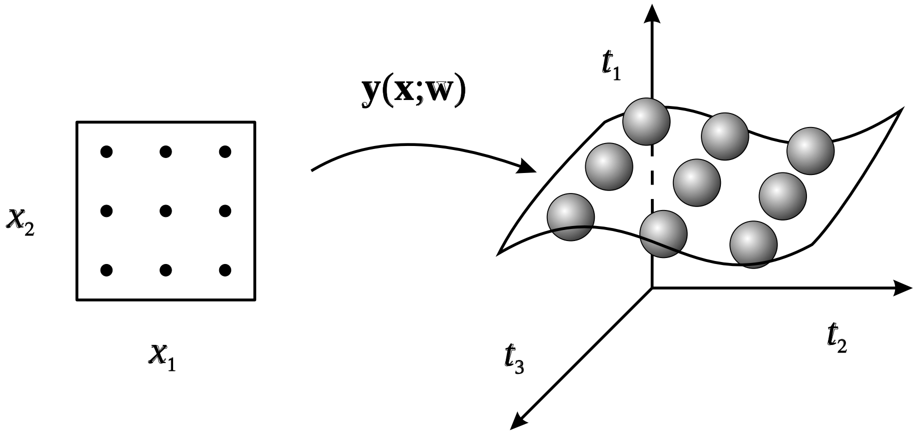

Figure 12. In order to formulate a tractable non-linear latent variable model, we consider a prior distribution $p(\mathbf{x})$ consisting of a superposition of delta functions, located at the nodes of a regular grid in latent space. Each node $\mathbf{x}_i$ is mapped to a corresponding point $\mathbf{y}(\mathbf{x}_i;\mathbf{w})$ in data space, and forms the centre of a corresponding Gaussian distribution.

which corresponds to a *constrained* Gaussian mixture model [13] since the centres of the Gaussians, given by $\mathbf{y}(\mathbf{x}_i;\mathbf{w})$, cannot move independently but are related through the function $\mathbf{y}(\mathbf{x};\mathbf{w})$. Note that, provided the mapping function $\mathbf{y}(\mathbf{x};\mathbf{w})$ is smooth and continuous, the projected points $\mathbf{y}(\mathbf{x}_i;\mathbf{w})$ will necessarily have a *topographic* ordering in the sense that any two points $\mathbf{x}_A$ and $\mathbf{x}_B$ which are close in latent space will map to points $\mathbf{y}(\mathbf{x}_A;\mathbf{w})$ and $\mathbf{y}(\mathbf{x}_B;\mathbf{w})$ which are close in data space.

5.1. AN EM ALGORITHM FOR GTM

Since GTM is a form of mixture model it is natural to seek an EM algorithm for maximizing the corresponding log likelihood. By choosing a particular form for the mapping $\mathbf{y}(\mathbf{x};\mathbf{w})$ we can obtain an EM algorithm in which the M-step has a simple form. In particular we shall choose $\mathbf{y}(\mathbf{x};\mathbf{w})$ to be given by a generalized linear regression model of the form

$$\mathbf{y}(\mathbf{x};\mathbf{w}) = \mathbf{W}\boldsymbol{\phi}(\mathbf{x}) \tag{46}$$

where the elements of $\boldsymbol{\phi}(\mathbf{x})$ consist of M fixed basis functions $\phi_j(\mathbf{x})$, and $\mathbf{W}$ is a $d \times M$ matrix. Generalized linear regression models possess the same universal approximation capabilities as multi-layer adaptive networks, provided the basis functions $\phi_j(\mathbf{x})$ are chosen appropriately. The usual limitation of such models, however, is that the number of basis functions must typically grow exponentially with the dimensionality q of the input space [5]. In the present context this is not a significant problem since the dimensionality is governed by the number of latent variables which will typically be small. In fact for data visualization applications we generally use $q = 2$.

In the E-step of the EM algorithm we evaluate the posterior probabilities for each of the latent points i for every data point $\mathbf{t}_n$ using

$$\begin{aligned} R_{in} &= p(\mathbf{x}_i|\mathbf{t}_n, \mathbf{W}, \sigma^2) && (47) \\ &= \frac{p(\mathbf{t}_n|\mathbf{x}_i, \mathbf{W}, \sigma^2)}{\sum_{i'=1}^{K} p(\mathbf{t}_n|\mathbf{x}_{i'}, \mathbf{W}, \sigma^2)}. && (48) \end{aligned}$$

Then in the M-step we obtain a revised value for $\mathbf{W}$ by solving a set of coupled *linear* equations of the form

$$\mathbf{\Phi}^{\mathrm{T}}\mathbf{G}\mathbf{\Phi}\mathbf{W}^{\mathrm{T}} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{T} \tag{49}$$

where $\mathbf{\Phi}$ 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

$$G_{ii} = \sum_{n=1}^{N} R_{in}(\mathbf{W}, \sigma^2). \tag{50}$$

We can now solve (49) for $\mathbf{W}$ using singular value decomposition to allow for possible ill-conditioning. Also in the M-step we update σ^2 using the following re-estimation formula

$$\sigma^2 = \frac{1}{Nd} \sum_{n=1}^{N} \sum_{i=1}^{K} R_{in}(\mathbf{W}, \sigma^2) \left\| \mathbf{W}\boldsymbol{\phi}(\mathbf{x}_i) - \mathbf{t}_n \right\|^2. \tag{51}$$

Note that the matrix $\mathbf{\Phi}$ is constant throughout the algorithm, and so need only be evaluated once at the start.

When using GTM for data visualization we can again plot each data point at the point on latent space corresponding to the mean of the posterior distribution, given by

$$\begin{aligned} \langle \mathbf{x}|\mathbf{t}_n, \mathbf{W}, \sigma^2 \rangle &= \int \mathbf{x} p(\mathbf{x}|\mathbf{t}_n, \mathbf{W}, \sigma^2)\, d\mathbf{x} && (52) \\ &= \sum_{i=1}^{K} R_{in}\mathbf{x}_i. && (53) \end{aligned}$$

It should be borne in mind, however, that as a consequence of the non-linear mapping from latent space to data space the posterior distribution can be multi-modal in which case the posterior mean can potentially give a very misleading summary of the true distribution. An alternative approach is therefore to evaluate the mode of the distribution, given by

$$i^{\max} = \arg\max_{\{i\}} R_{in}. \tag{54}$$

In practice it is often convenient to plot both the mean and the mode for each data point, as significant differences between them can be indicative of a multi-modal distribution.

One of the motivations for the development of the GTM algorithm was to provide a principled alternative to the widely used ‘self-organizing map’ (SOM) algorithm [17] in which a set of unlabelled data vectors $\mathbf{t}_n$ ($n = 1, \ldots, N$) in a d-dimensional data space is summarized in terms of a set of reference vectors having a spatial organization corresponding to a (generally) two-dimensional sheet. These reference vectors are analogous to the projections of the latent points into data space given by $\mathbf{y}(\mathbf{x}_i; \mathbf{w})$. While the SOM algorithm has achieved many successes in practical applications, it also suffers from some significant deficiencies, many of which are highlighted in [18]. These include: the absence of a cost function, the lack of any guarantee of topographic ordering, the absence of any general proofs of convergence, and the fact that the model does not define a probability density. These problems are all absent in GTM. The computational complexities of the GTM and SOM algorithms are similar, since the dominant cost in each case is the evaluation of the Euclidean distanced between each data point and each reference point in data space, and is the same for both algorithms.

Clearly, we can easily formulate a density model consisting of a mixture of GTM models, and obtain the corresponding EM algorithm, in a principled manner. The development of an analogous algorithm for the SOM would necessarily be somewhat ad-hoc.

5.2. GEOMETRY OF THE MANIFOLD

An additional advantage of the GTM algorithm (compared with the SOM) is that the non-linear manifold in data space is defined explicitly in terms of the analytic function $\mathbf{y}(\mathbf{x}; \mathbf{w})$. This allows a whole variety of geometrical properties of the manifold to be evaluated [8]. For example, local magnification factors can be expressed in terms of derivatives of the basis functions appearing in (46). Magnification factors specify the extent to which the area of a small patch of the latent space of a topographic mapping is magnified on projection to the data space, and are of considerable interest in both neuro-biological and data analysis contexts. Previous attempts to consider magnification factors for the SOM were been hindered because the manifold is only defined at discrete points (given by the reference vectors).

We can determine the properties of the manifold, including magnification factors, using techniques of differential geometry as follows [8]. Consider a standard set of Cartesian coordinates x^i in the latent space. Since each point P in latent space is mapped by a continuous function to a corresponding point P' in data space, the mapping defines a set of curvilinear coordinates ξ^i in the manifold in which each point P' is labelled with the coordinate values $\xi^i = x^i$ of P, as illustrated in Figure 13. Throughout this section we shall use the standard notation of differential geometry in which raised indices denote contravariant components and lowered indices denote covariant components, with an implicit summation over pairs of repeated covariant-contravariant indices.

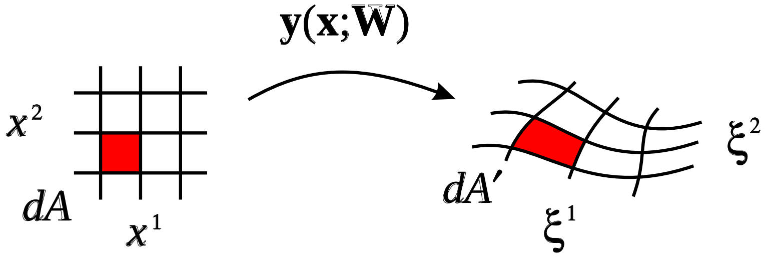

Figure 13. This diagram shows the mapping of the Cartesian coordinate system x^i in latent space onto a curvilinear coordinate system ξ^i in the q-dimensional manifold $\mathcal{S}$.

We first discuss the metric properties of the manifold $\mathcal{S}$. Consider a local transformation, at some point P' in $\mathcal{S}$, to a set of rectangular Cartesian coordinates $\zeta^i = \zeta^i(\boldsymbol{\xi})$. Then the squared length element in these coordinates is given by

$$ds^2 = \delta_{\mu\nu} d\zeta^\mu d\zeta^\nu = \delta_{\mu\nu} \frac{\partial \zeta^\mu}{\partial \xi^i} \frac{\partial \zeta^\nu}{\partial \xi^j} d\xi^i d\xi^j = g_{ij} d\xi^i d\xi^j \tag{55}$$

where g_{ij} is the metric tensor, which is therefore given by

$$g_{ij} = \delta_{\mu\nu} \frac{\partial \zeta^\mu}{\partial \xi^i} \frac{\partial \zeta^\nu}{\partial \xi^j}. \tag{56}$$

We now seek an expression for g_{ij} in terms of the non-linear mapping $\mathbf{y}(\mathbf{x})$. Consider again the squared length element ds^2 lying within the manifold $\mathcal{S}$. Since $\mathcal{S}$ is embedded within the Euclidean data space, this also corresponds to the squared length element of the form

$$ds^2 = \delta_{kl} dy^k dy^l = \delta_{kl} \frac{\partial y^k}{\partial x^i} \frac{\partial y^l}{\partial x^j} dx^i dx^j = g_{ij} dx^i dx^j \tag{57}$$

and so we have

$$g_{ij} = \delta_{kl} \frac{\partial y^k}{\partial x^i} \frac{\partial y^l}{\partial x^j}. \tag{58}$$

Using (46) the metric tensor can be expressed in terms of the derivatives of the basis functions $\phi_j(\mathbf{x})$ in the form

$$\mathbf{g} = \boldsymbol{\Omega}^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{W} \boldsymbol{\Omega} \tag{59}$$

where $\boldsymbol{\Omega}$ has elements $\Omega_{ji} = \partial \phi_j / \partial x^i$. It should be emphasized that, having obtained the metric tensor as a function of the latent space coordinates, many other geometrical properties are easily evaluated, such as the local curvatures of the manifold.

Our goal is to find an expression for the area dA' of the region of $\mathcal{S}$ corresponding to an infinitesimal rectangle in latent space with area $dA = \prod_i dx^i$ as shown in Figure 13. The area element in the manifold $\mathcal{S}$ can be related to the corresponding area element in the latent space by the Jacobian of the transformation $\xi \to \zeta$

$$dA' = \prod_{\mu} d\zeta^{\mu} = J \prod_{i} d\xi^{i} = J \prod_{i} dx^{i} = J dA \tag{60}$$

where the Jacobian J is given by

$$J = \det \left(\frac{\partial \zeta^{\mu}}{\partial \xi^{i}} \right) = \det \left(\frac{\partial \zeta^{\mu}}{\partial x^{i}} \right). \tag{61}$$

We now introduce the determinant g of the metric tensor which we can write in the form

$$g = \det(g_{ij}) = \det \left(\delta_{\mu\nu} \frac{\partial \zeta^{\mu}}{\partial x^{i}} \frac{\partial \zeta^{\nu}}{\partial x^{j}} \right) = \det \left(\frac{\partial \zeta^{\mu}}{\partial x^{i}} \right) \det \left(\frac{\partial \zeta^{\nu}}{\partial x^{j}} \right) = J^2 \tag{62}$$

and so, using (60), we obtain an expression for the local magnification factor in the form

$$\frac{dA'}{dA} = J = \det{}^{1/2} \mathbf{g}. \tag{63}$$

Although the magnification factor represents the extent to which areas are magnified on projection to the data space, it gives no information about which directions in latent space correspond to the stretching. We can recover this information by considering the decomposition of the metric tensor $\mathbf{g}$ in terms of its eigenvectors and eigenvalues. This information can be conveniently displayed by selecting a regular grid in latent space (which could correspond to the reference vector grid, but could also be much finer) and plotting at each grid point an ellipse with principal axes oriented according to the eigenvectors, with principal radii given by the square roots of the eigenvalues. The standard area magnification factor is given from (63) by the square root of the product of the eigenvalues, and so corresponds to the area of the ellipse.

As an illustration of the GTM algorithm and the evaluation of magnification factors we consider a data set of measurements taken from the genus *Leptograpsus* of rock crabs[1]. Measurements were taken from two species classified by their colour (orange or blue) with the aim of discovering morphological differences which would allow preserved specimens (which have lost their colour) to be distinguished. The data set contains 50 examples of each sex from each species, and the measurements correspond to length of frontal lip, rear width, length along mid-line, maximum width of carapace, and body length. Since all of the variables correspond to length measurements, the dominant feature of the crabs data is an overall scaling of the data vector in relation to the size of the crab. To remove this effect each data vector $\mathbf{t}_n = (t_{1n}, \ldots, t_{dn})^{\mathrm{T}}$ is normalized to unit mean, so that

$$\widetilde{t}_{kn} = t_{kn} \Bigg/ \sum_{k'=1}^{d} t_{k'n}. \tag{64}$$

The latent space visualization of the crabs data is shown in Figure 14 together with the local magnification factor. It can be seen that the two species form distinct clusters, with the manifold undergoing a relatively

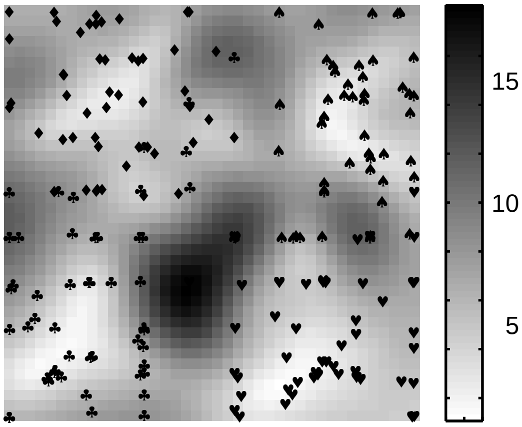

Figure 14. Plot of the latent-space distribution of the crabs data, in which ♣ denotes blue males, ♦ denotes blue females, ♥ denotes orange males, and ♠ denotes orange females. The grey-scale background shows the corresponding magnification factor as a function of the latent space coordinates, in which darker shades indicate larger values of the magnification factor.

[1] Available from Brian Ripley at: `http://markov.stats.ox.ac.uk/pub/PRNN`.

large stretching in the region between them. Within each cluster there is a partial separation of males from females.

The corresponding plot of the local eigenvector decomposition of the metric is given in Figure 15, and shows both the direction and magnitude of the stretching.

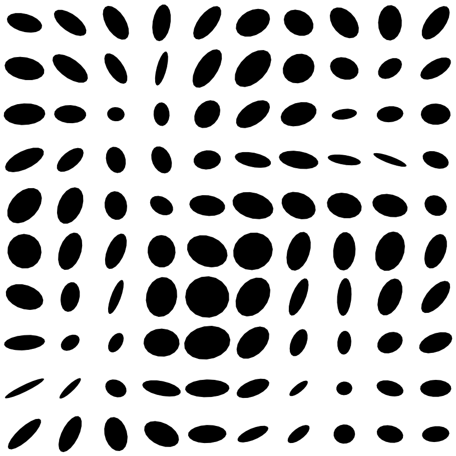

Figure 15. Plots of the local stretching of the latent space, corresponding to the example in Figure 14, using the ellipse representation discussed in the text.

6. Temporal Models: GTM Through Time

In all of the models we have considered so far, it has been assumed that the data vectors are independent and identically distributed. One common situation in which this assumption is generally violated in where the data vectors are successive samples in a time series, with neighbouring vectors having typically a high correlation. As our final example of the use of latent variables, we consider an extension of the GTM algorithm to deal with temporal data [6]. The key observation is that the hidden states of the GTM model are discrete, as a result of the choice of latent distribution $p(\mathbf{x})$, which allows the machinery of hidden Markov models to be combined with GTM to give a non-linear temporal latent variable model.

The structure of the model is illustrated in Figure 16, in which the hidden states of the model at each time step n are labelled by the index i_n corresponding to the latent points $\{\mathbf{x}_{i_n}\}$. We introduce a set of transition probabilities $p(i_{n+1}|i_n)$ corresponding to the probability of making a transition to state i_{n+1} given that the current state is i_n. The emission density for the hidden Markov model is then given by the GTM density model (45). It should be noted that both the transition probabilities $p(i_{n+1}|i_n)$

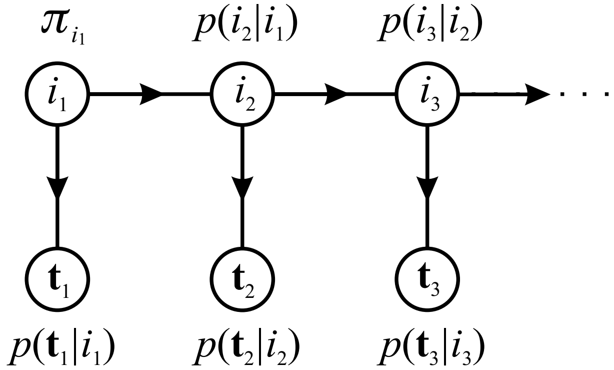

Figure 16. The temporal version of GTM consists of a hidden Markov model in which the hidden states are given by the latent points of the GTM model, and the emission probabilities are governed by the GTM mixture distribution. Note that the parameters of the GTM model, as well as the transition probabilities between states, are tied to common values across all time steps. For clarity we have simplified the graph and not made the factorization property of the conditional distribution $p(\mathbf{t}|i)$ explicit.

and the parameters $\mathbf{W}$ and σ^2 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 introduce separate prior probabilities π_{i_1} on each of the latent points at the first time step of the algorithm.

Again we can obtain an EM algorithm for maximizing the likelihood for the temporal GTM model. In the context of hidden Markov models, the EM algorithm is often called the Baum-Welch algorithm, and is reviewed in [21]. The E-step involves the evaluation of the posterior probabilities of the hidden states at each time step, and can be accomplished efficiently using a technique called the *forward-backward* algorithm since it involves two counter-directional propagations along the Markov chain. The M-step equations again take the form given in Section 5.1.

As an illustration of the temporal GTM algorithm we consider a data set obtained from a series of helicopter test flights. The motivation behind this application is to determine 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 with greater accuracy.

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 17.

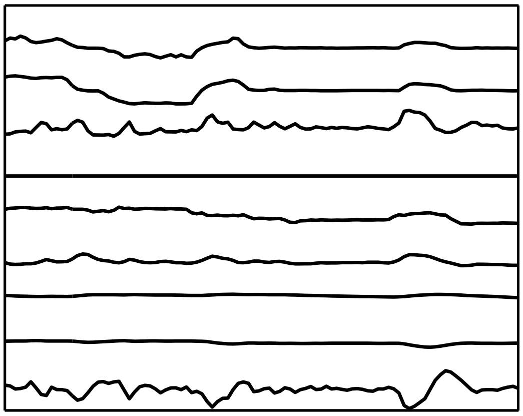

Figure 17. Sample of the helicopter data showing the evolution of 9 different dynamical quantities such as acceleration, speed, and engine torque as a function of time.

Figure 18 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.

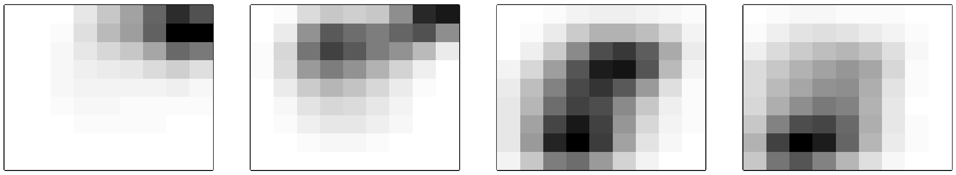

Figure 18. Plots of the posterior probability distribution in latent space at 4 time steps, corresponding to a transition from one flight regime to another.

7. Discussion

In this chapter we have surveyed a number of hidden variable models for representing the distributions of continuous variables. We considered examples involving discrete, continuous and mixed hidden variables, together with linear and non-linear models for the distribution of observed variables conditioned on the hidden variables.

These models can conveniently be expressed in terms of probabilistic graphical structures, which provides insight into the corresponding inference and learning algorithms. For example, we saw that two key operations are the inference of the posterior distribution of the hidden variables given the observed variables (corresponding to the E-step of the EM algorithm) and the evaluation of the likelihood function which involves summing or integrating over all possible configurations of the hidden variables. For the models considered in this chapter, the integration over continuous hidden variables was possible because of simple (linear and Gaussian) choices for the model structure. In the case of discrete hidden variables we considered models in which only one of the hidden states can be active at any one time, giving rise to the standard mixture distribution.

The graphical viewpoint, however, also helps to motivate the development of new classes of probabilistic model. For instance, more general representations for discrete hidden states can be considered in which there are multiple hidden variables. However, for many models this leads to intractable algorithms since the number of *configurations* of hidden states may grow exponentially with the number of hidden variables. The development of controlled approximations to deal with such models is currently the focus of extensive research within the graphical modelling and neural computing communities.

ACKNOWLEDGEMENTS

The author would like to thank the following for their contributions to the work reported in this chapter: Geoffrey Hinton, Iain Strachan, Markus Svensén, Michael Tipping and Chris Williams.

References

1. Anderson, T. W. (1958). *An Introduction to Multivariate Statistical Analysis.* New York: John Wiley.
2. Anderson, T. W. (1963). Asymptotic theory for principal component analysis. *Annals of Mathematical Statistics 34*, 122–148.
3. Bartholomew, D. J. (1987). *Latent Variable Models and Factor Analysis.* London: Charles Griffin & Co. Ltd.
4. Basilevsky, A. (1994). *Statistical Factor Analysis and Related Methods.* New York: Wiley.
5. Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press.
6. 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.
7. Bishop, C. M. and G. D. James (1993). Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. *Nuclear Instruments and Methods in Physics Research A327*, 580–593.
8. Bishop, C. M., M. Svensén, and C. K. I. Williams (1997). Magnification factors for the GTM algorithm. In *Proceedings IEE Fifth International Conference on Artificial Neural Networks, Cambridge, U.K.*, pp. 64–69.
9. Bishop, C. M., M. Svensén, and C. K. I. Williams (1998). GTM: the Generative

Topographic Mapping. *Neural Computation 10*(1), 215–234.

10. 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.
11. 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.
12. Hinton, G. E., P. Dayan, and M. Revow (1997). Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks 8*(1), 65–74.
13. Hinton, G. E., C. K. I. Williams, and M. D. Revow (1992). Adaptive elastic models for hand-printed character recognition. In J. E. Moody, S. J. Hanson, and R. P. Lippmann (Eds.), *Advances in Neural Information Processing Systems*, Volume 4, pp. 512–519. Morgan Kauffmann.
14. Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology 24*, 417–441.
15. Hull, J. J. (1994). A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence 16*, 550–554.
16. Jordan, M. I. and R. A. Jacobs (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation 6*(2), 181–214.
17. Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics 43*, 59–69.
18. Kohonen, T. (1995). *Self-Organizing Maps*. Berlin: Springer-Verlag.
19. Krzanowski, W. J. and F. H. C. Marriott (1994). *Multivariate Analysis Part I: Distributions, Ordination and Inference*. London: Edward Arnold.
20. Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, Sixth Series 2*, 559–572.
21. Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*(2), 257–285.
22. Rao, C. R. (1955). Estimation and tests of significance in factor analysis. *Psychometrika 20*, 93–111.
23. Rubin, D. B. and D. T. Thayer (1982). EM algorithms for ML factor analysis. *Psychometrika 47*(1), 69–76.
24. Tipping, M. E. and C. M. Bishop (1999a). Mixtures of probabilistic principal component analyzers. *Neural Computation 11*(2), 443–482.
25. Tipping, M. E. and C. M. Bishop (1999b). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B 21*(3), 611–622.