A Hierarchical Latent Variable Model for Data Visualization

Christopher M. Bishop and Michael E. Tipping

Abstract—Visualization has proven to be a powerful and widely-applicable tool for the analysis and interpretation of multivariate data. Most visualization algorithms aim to find a projection from the data space down to a two-dimensional visualization space. However, for complex data sets living in a high-dimensional space, it is unlikely that a single two-dimensional projection can reveal all of the interesting structure. We therefore introduce a hierarchical visualization algorithm which allows the complete data set to be visualized at the top level, with clusters and subclusters of data points visualized at deeper levels. The algorithm is based on a hierarchical mixture of latent variable models, whose parameters are estimated using the expectation-maximization algorithm. We demonstrate the principle of the approach on a toy data set, and we then apply the algorithm to the visualization of a synthetic data set in 12 dimensions obtained from a simulation of multiphase flows in oil pipelines, and to data in 36 dimensions derived from satellite images. A Matlab software implementation of the algorithm is publicly available from the World Wide Web.

Index Terms—Latent variables, data visualization, EM algorithm, hierarchical mixture model, density estimation, principal component analysis, factor analysis, maximum likelihood, clustering, statistics.

1 Introduction

Many algorithms for data visualization have been proposed by both the neural computing and statistics communities, most of which are based on a projection of the data onto a two-dimensional visualization space. While such algorithms can usefully display the structure of simple data sets, they often prove inadequate in the face of data sets which are more complex. A single two-dimensional projection, even if it is nonlinear, may be insufficient to capture all of the interesting aspects of the data set. For example, the projection which best separates two clusters may not be the best for revealing internal structure within one of the clusters. This motivates the consideration of a hierarchical model involving multiple two-dimensional visualization spaces. The goal is that the top-level projection should display the entire data set, perhaps revealing the presence of clusters, while lower-level projections display internal structure within individual clusters, such as the presence of subclusters, which might not be apparent in the higher-level projections.

Once we allow the possibility of many complementary visualization projections, we can consider each projection model to be relatively simple, for example, based on a linear projection, and compensate for the lack of flexibility of individual models by the overall flexibility of the complete hierarchy. The use of a hierarchy of relatively simple models offers greater ease of interpretation as well as the benefits of analytical and computational simplification. This philosophy for modeling complexity is similar in spirit to the "mixture of experts" approach for solving regression problems [1].

The algorithm discussed in this paper is based on a form of latent variable model which is closely related to both principal component analysis (PCA) and factor analysis. At the top level of the hierarchy we have a single visualization plot corresponding to one such model. By considering a probabilistic mixture of latent variable models we obtain a soft partitioning of the data set into "clusters," corresponding to the second level of the hierarchy. Subsequent levels, obtained using nested mixture representations, provide successively refined models of the data set. The construction of the hierarchical tree proceeds top down, and can be driven interactively by the user. At each stage of the algorithm the relevant model parameters are determined using the expectation-maximization (EM) algorithm.

In the next section, we review the latent-variable model, and, in Section 3, we discuss the extension to mixtures of such models. This is further extended to hierarchical mixtures in Section 4, and is then used to formulate an interactive visualization algorithm in Section 5. We illustrate the operation of the algorithm in Section 6 using a simple toy data set. Then we apply the algorithm to a problem involving the monitoring of multiphase flows along oil pipes in Section 7 and to the interpretation of satellite image data in Section 8. Finally, extensions to the algorithm, and the relationships to other approaches, are discussed in Section 9.

2 Latent Variables

We begin by introducing a simple form of linear latent variable model and discussing its application to data analysis. Here we give an overview of the key concepts, and leave

- *C.M. Bishop is with Microsoft Research, St. George House, 1 Guildhall Street, Cambridge CB2 3NH, U.K. E-mail: cmbishop@microsoft.com.*
- *M.E. Tipping is with the Neural Computing Research Group, Aston University, Birmingham B4 7ET, U.K. E-mail: m.e.tipping@aston.ac.uk.*

Manuscript received 3 Apr. 1997; revised 23 Jan. 1998. Recommended for acceptance by R.W. Picard.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 106323.

the detailed mathematical discussion to Appendix A. The aim is to find a representation of a multidimensional data set in terms of two latent (or "hidden") variables. Suppose the data space is d-dimensional with coordinates $y_1, \ldots, y_d$ and that the data set consists of a set of d-dimensional vectors $\{\mathbf{t}_n\}$ where $n = 1, \ldots, N$. Now consider a two-dimensional latent space $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$ together with a linear function which maps the latent space into the data space

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} \tag{1}$$

where $\mathbf{W}$ is a $d \times 2$ matrix and $\boldsymbol{\mu}$ is a d-dimensional vector. The mapping (1) defines a two-dimensional planar surface in the data space. If we introduce a prior probability distribution $p(\mathbf{x})$ over the latent space given by a zero-mean Gaussian with a unit covariance matrix, then (1) defines a singular Gaussian distribution in data space with mean $\boldsymbol{\mu}$ and covariance matrix $\langle(\mathbf{y} - \boldsymbol{\mu})(\mathbf{y} - \boldsymbol{\mu})^{\mathrm{T}}\rangle = \mathbf{W}\mathbf{W}^{\mathrm{T}}$. Finally, since we do not expect the data to be confined exactly to a two-dimensional sheet, we convolve this distribution with an isotropic Gaussian distribution $p(\mathbf{t} \mid \mathbf{x}, \sigma^2)$ in data space, having a mean of zero and covariance $\sigma^2\mathbf{I}$, where $\mathbf{I}$ is the unit matrix. Using the rules of probability, the final density model is obtained from the convolution of the noise model with the prior distribution over latent space in the form

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

Since this represents the convolution of two Gaussians, the integral can be evaluated analytically, resulting in a distribution $p(\mathbf{t})$ which corresponds to a d-dimensional Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$.

If we had considered a more general model in which the conditional distribution $p(\mathbf{t} \mid \mathbf{x})$ is given by a Gaussian with a general diagonal covariance matrix (having d independent parameters), then we would obtain standard linear factor analysis [2], [3]. In fact, our model is more closely related to principal component analysis, as we now discuss.

The log likelihood function for this model is given by $L = \sum_n \ln p(\mathbf{t}_n)$, and maximum likelihood can be used to fit the model to the data and hence determine values for the parameters $\boldsymbol{\mu}$, $\mathbf{W}$, and σ^2. The solution for $\boldsymbol{\mu}$ is just given by the sample mean. In the case of the factor analysis model, the determination of $\mathbf{W}$ and σ^2 corresponds to a nonlinear optimization which must be performed iteratively. For the isotropic noise covariance matrix, however, it was shown by Tipping and Bishop [4], [5] that there is an exact closed form solution as follows. If we introduce the sample covariance matrix given by

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}\left(\mathbf{t}_n - \boldsymbol{\mu}\right)\left(\mathbf{t}_n - \boldsymbol{\mu}\right)^{\mathrm{T}}, \tag{3}$$

then the only nonzero stationary points of the likelihood occur for:

$$\mathbf{W} = \mathbf{U}\,(\boldsymbol{\Lambda} - \sigma^2\mathbf{I})^{1/2}\mathbf{R}, \tag{4}$$

where the two columns of the matrix $\mathbf{U}$ are eigenvectors of $\mathbf{S}$, with corresponding eigenvalues in the diagonal matrix $\boldsymbol{\Lambda}$, and $\mathbf{R}$ is an arbitrary 2×2 orthogonal rotation matrix. Furthermore, it was shown that the stationary point corresponding to the *global maximum* of the likelihood occurs when the columns of $\mathbf{U}$ comprise the two *principal eigenvectors* of $\mathbf{S}$ (i.e., the eigenvectors corresponding to the two largest eigenvalues) and that all other combinations of eigenvectors represent saddle-points of the likelihood surface. It was also shown that the maximum-likelihood estimator of σ^2 is given by

$$\sigma_{\mathrm{ML}}^2 = \frac{1}{d-2}\sum_{j=3}^{d}\lambda_j, \tag{5}$$

which has a clear interpretation as the variance "lost" in the projection, averaged over the lost dimensions.

Unlike conventional PCA, however, our model defines a probability density in data space, and this is important for the subsequent hierarchical development of the model. The choice of a radially symmetric rather than a more general diagonal covariance matrix for $p(\mathbf{t} \mid \mathbf{x})$ is motivated by the desire for greater ease of interpretability of the visualization results, since the projections of the data points onto the latent plane in data space correspond (for small values of σ^2) to an orthogonal projection as discussed in Appendix A.

Although we have an explicit solution for the maximum-likelihood parameter values, it was shown by Tipping and Bishop [4], [5] that significant computational savings can sometimes be achieved by using the following EM (expectation-maximization) algorithm [6], [7], [8]. Using (2), we can write the log likelihood function in the form

$$L = \sum_{n=1}^{N}\ln\int p\left(\mathbf{t}_n \middle| \mathbf{x}_n\right)p\left(\mathbf{x}_n\right)d\mathbf{x}_n, \tag{6}$$

in which we can regard the quantities $\mathbf{x}_n$ as missing variables. The posterior distribution of the $\mathbf{x}_n$, given the observed $\mathbf{t}_n$ and the model parameters, is obtained using Bayes' theorem and again consists of a Gaussian distribution. The E-step then involves the use of "old" parameter values to evaluate the sufficient statistics of this distribution in the form

$$\langle\mathbf{x}_n\rangle = \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{t}_n - \boldsymbol{\mu}) \tag{7}$$

$$\left\langle\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\right\rangle = \sigma^2\mathbf{M}^{-1} + \left\langle\mathbf{x}_n\right\rangle\left\langle\mathbf{x}_n^{\mathrm{T}}\right\rangle, \tag{8}$$

where $\mathbf{M} = \mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2\mathbf{I}$ is a 2×2 matrix, and $\langle\ \rangle$ denotes the expectation computed with respect to the posterior distribution of $\mathbf{x}$. The M-step then maximizes the expectation of the complete-data log likelihood to give

$$\widetilde{\mathbf{W}} = \left[\sum_{n=1}^{N}\left(\mathbf{t}_n - \boldsymbol{\mu}\right)\left\langle\mathbf{x}_n^{\mathrm{T}}\right\rangle\right]\left[\sum_{n=1}^{N}\left\langle\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\right\rangle\right]^{-1} \tag{9}$$

$$\tilde{\sigma}^2 = \frac{1}{Nd}\sum_{n=1}^{N}\left\{\left\|\mathbf{t}_n - \boldsymbol{\mu}\right\|^2 + \mathrm{Tr}\left(\widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{W}}\left\langle\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\right\rangle\right) - 2\left\langle\mathbf{x}_n^{\mathrm{T}}\right\rangle\widetilde{\mathbf{W}}^{\mathrm{T}}\left(\mathbf{t}_n - \boldsymbol{\mu}\right)\right\} \tag{10}$$

in which ~ denotes "new" quantities. Note that the new value for $\widetilde{\mathbf{W}}$ obtained from (9) is used in the evaluation of σ^2 in (10). The model is trained by alternately evaluating the sufficient statistics of the latent-space posterior distribution using (7) and (8) for given σ^2 and $\mathbf{W}$ (the E-step), and re-evaluating σ^2 and $\mathbf{W}$ using (9) and (10) for given $\langle \mathbf{x}_n \rangle$ and $\left\langle \mathbf{x}_n \mathbf{x}_n^T \right\rangle$ (the M-step). It can be shown that, at each stage of the EM algorithm, the likelihood is increased unless it is already at a local maximum, as demonstrated in Appendix E.

For N data points in d dimensions, evaluation of the sample covariance matrix requires $O(Nd^2)$ operations, and so any approach to finding the principal eigenvectors based on an explicit evaluation of the covariance matrix must have at least this order of computational complexity. By contrast, the EM algorithm involves steps which are only $O(Nd)$. This saving of computational cost is a consequence of having a latent space whose dimensionality (which, for the purposes of our visualization algorithm, is fixed at two) does not scale with d.

If we substitute the expressions for the expectations given by the E-step equations (7) and (8) into the M-step equations we obtain the following re-estimation formulas

$$\widetilde{\mathbf{W}} = \mathbf{SW}(\sigma^2\mathbf{I} + \mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{SW})^{-1} \qquad (11)$$

$$\widetilde{\sigma}^2 = \frac{1}{d}\mathrm{Tr}\left\{\mathbf{S} - \mathbf{SWM}^{-1}\widetilde{\mathbf{W}}^{\mathrm{T}}\right\} \qquad (12)$$

which shows that all of the dependence on the data occurs through the sample covariance matrix $\mathbf{S}$. Thus the EM algorithm can be expressed as alternate evaluations of (11) and (12). (Note that (12) involves a combination of "old" and "new" quantities.) This form of the EM algorithm has been introduced for illustrative purposes only, and would involve $O(Nd^2)$ computational cost due to the evaluation of the covariance matrix.

We have seen that each data point $\mathbf{t}_n$ induces a Gaussian posterior distribution $p(\mathbf{x}_n | \mathbf{t}_n)$ in the latent space. For the purposes of visualization, however, it is convenient to summarize each such distribution by its mean, given by $\langle \mathbf{x}_n \rangle$, as illustrated in Fig. 1. Note that these quantities are obtained directly from the output of the E-step (7). Thus, a set of data points $\{\mathbf{t}_n\}$ where $n = 1, \ldots, N$ is projected onto a corresponding set of points $\{\langle \mathbf{x}_n \rangle\}$ in the two-dimensional latent space.

3 Mixtures of Latent Variable Models

We can perform an automatic soft clustering of the data set, and at the same time obtain multiple visualization plots corresponding to the clusters, by modeling the data with a *mixture* of latent variable models of the kind described in Section 2. The corresponding density model takes the form

$$p(\mathbf{t}) = \sum_{i=1}^{M_0} \pi_i p(\mathbf{t}|i) \qquad (13)$$

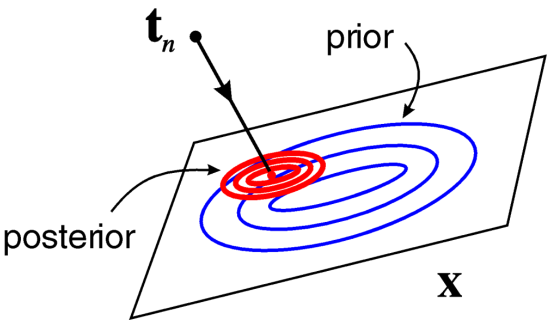


Fig. 1. Illustration of the projection of a data point onto the mean of the posterior distribution in latent space.

where M_0 is the number of components in the mixture, and the parameters π_i are the mixing coefficients, or prior probabilities, corresponding to the mixture components $p(\mathbf{t} | i)$. Each component is an independent latent variable model with parameters $\boldsymbol{\mu}_i$, $\mathbf{W}_i$, and σ_i^2. This mixture distribution will form the second level in our hierarchical model.

The EM algorithm can be extended to allow a mixture of the form (13) to be fitted to the data (see Appendix B for details). To derive the EM algorithm we note that, in addition to the $\{\mathbf{x}_n\}$, the missing data now also includes labels which specify which component is responsible for each data point. It is convenient to denote this missing data by a set of variables z_{ni} where $z_{ni} = 1$ if $\mathbf{t}_n$ was generated by model i (and zero otherwise). The prior expectations for these variables are given by the π_i and the corresponding posterior probabilities, or responsibilities, are evaluated in the extended E-step using Bayes' theorem in the form

$$R_{ni} = P(i|\mathbf{t}_n) = \frac{\pi_i p(\mathbf{t}_n|i)}{\sum_{i'} \pi_{i'} p(\mathbf{t}_n|i')}. \qquad (14)$$

Although a standard EM algorithm can be derived by treating the $\{\mathbf{x}_n\}$ and the z_{ni} jointly as missing data, a more efficient algorithm can be obtained by considering a two-stage form of EM. At each complete cycle of the algorithm we commence with an "old" set of parameter values π_i, $\boldsymbol{\mu}_i$, $\mathbf{W}_i$, and σ_i^2. We first use these parameters to evaluate the posterior probabilities R_{ni} using (14). These posterior probabilities are then used to obtain "new" values $\widetilde{\pi}_i$ and $\widetilde{\boldsymbol{\mu}}_i$ using the following re-estimation formulas

$$\widetilde{\pi}_i = \frac{1}{N}\sum_n R_{ni} \qquad (15)$$

$$\widetilde{\boldsymbol{\mu}}_i = \frac{\sum_n R_{ni}\mathbf{t}_n}{\sum_n R_{ni}}. \qquad (16)$$

The new values $\widetilde{\boldsymbol{\mu}}_i$ are then used in evaluation of the sufficient statistics for the posterior distribution for $\mathbf{x}_{ni}$

$$\left\langle \mathbf{x}_{ni} \right\rangle = \mathbf{M}_i^{-1}\mathbf{W}_i^{\mathrm{T}}\left(\mathbf{t}_n - \widetilde{\boldsymbol{\mu}}_i\right) \qquad (17)$$

$$\left\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}} \right\rangle = \sigma_i^2\mathbf{M}_i^{-1} + \left\langle \mathbf{x}_{ni} \right\rangle \left\langle \mathbf{x}_{ni}^{\mathrm{T}} \right\rangle \qquad (18)$$

where $\mathbf{M}_i = \mathbf{W}_i^{\mathrm{T}}\mathbf{W}_i + \sigma_i^2\mathbf{I}$. Finally, these statistics are used to evaluate "new" values $\widetilde{\mathbf{W}}_i$ and $\widetilde{\sigma}_i^2$ using

$$\widetilde{\mathbf{W}}_i = \left[\sum_n R_{ni}\left(\mathbf{t}_n - \widetilde{\boldsymbol{\mu}}_i\right)\left\langle \mathbf{x}_{ni}^{\mathrm{T}}\right\rangle\right]\left[\sum_n R_{ni}\left\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}}\right\rangle\right]^{-1} \tag{19}$$

$$\widetilde{\sigma}_i^2 = \frac{1}{d\sum_n R_{ni}}\left\{\sum_n R_{ni}\left\|\mathbf{t}_n - \widetilde{\boldsymbol{\mu}}_i\right\|^2 - 2\sum_n R_{ni}\left\langle \mathbf{x}_{ni}^{\mathrm{T}}\right\rangle \widetilde{\mathbf{W}}_i^{\mathrm{T}}\left(\mathbf{t}_n - \widetilde{\boldsymbol{\mu}}_i\right) + \sum_n R_{ni}\mathrm{Tr}\left[\left\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}}\right\rangle \widetilde{\mathbf{W}}_i^{\mathrm{T}}\widetilde{\mathbf{W}}_i\right]\right\} \tag{20}$$

which are derived in Appendix B.

As for the single latent variable model, we can substitute the expressions for $\langle \mathbf{x}_{ni}\rangle$ and $\left\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}}\right\rangle$, given by (17) and (18), respectively, into (19) and (20). We then see that the re-estimation formulae for $\widetilde{\mathbf{W}}_i$ and $\widetilde{\sigma}_i^2$ take the form

$$\widetilde{\mathbf{W}}_i = \mathbf{S}_i\mathbf{W}_i\left(\sigma_i^2\mathbf{I} + \mathbf{M}_i^{-1}\mathbf{W}_i^{\mathrm{T}}\mathbf{S}_i\mathbf{W}_i\right)^{-1} \tag{21}$$

$$\widetilde{\sigma}_i^2 = \frac{1}{d}\mathrm{Tr}\left(\mathbf{S}_i - \mathbf{S}_i\mathbf{W}_i\mathbf{M}_i^{-1}\widetilde{\mathbf{W}}_i^{\mathrm{T}}\right), \tag{22}$$

where all of the data dependence been expressed in terms of the quantities

$$\mathbf{S}_i = \frac{1}{N_i}\sum_n R_{ni}\left(\mathbf{t}_n - \widetilde{\boldsymbol{\mu}}_i\right)\left(\mathbf{t}_n - \widetilde{\boldsymbol{\mu}}_i\right)^{\mathrm{T}}, \tag{23}$$

and we have defined $N_i = \sum_n R_{ni}$. The matrix $\mathbf{S}_i$ can clearly be interpreted as a responsibility weighted covariance matrix. Again, for reasons of computational efficiency, the form of EM algorithm given by (17) to (20) is to be preferred if d is large.

4 Hierarchical Mixture Models

We now extend the mixture representation of Section 3 to give a hierarchical mixture model. Our formulation will be quite general and can be applied to mixtures of any parametric density model.

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

$$p(\mathbf{t}) = \sum_{i=1}^{M_0}\pi_i\sum_{j\in\mathcal{G}_i}\pi_{j|i}\,p(\mathbf{t}|i,j), \tag{24}$$

where $p(\mathbf{t}|i, j)$ again represent independent latent variable models, and $\pi_{j|i}$ correspond to sets of mixing coefficients, one for each i, which satisfy $\sum_j\pi_{j|i} = 1$. 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 Fig. 2.

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 (24) would take the form

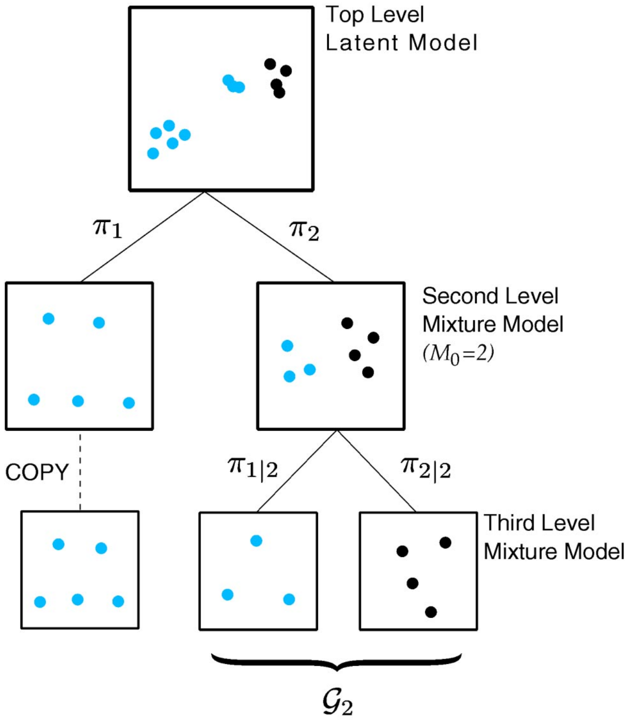


Fig. 2. The structure of the hierarchical model.

$$L = \sum_{n=1}^{N}\ln\left\{\sum_{i=1}^{M_0}\pi_i\sum_{j\in\mathcal{G}_i}\pi_{j|i}\,p(\mathbf{t}|i,j)\right\}. \tag{25}$$

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

$$L = \sum_{n=1}^{N}\sum_{i=1}^{M_0}z_{ni}\ln\left\{\pi_i\sum_{j\in\mathcal{G}_i}\pi_{j|i}\,p(\mathbf{t}|i,j)\right\}. \tag{26}$$

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. Taking the expectation of (26), we then obtain the log likelihood for the third level of the hierarchy in the form

$$L = \sum_{n=1}^{N}\sum_{i=1}^{M_0}R_{ni}\ln\left\{\pi_i\sum_{j\in\mathcal{G}_i}\pi_{j|i}\,p(\mathbf{t}|i,j)\right\}, \tag{27}$$

in which the R_{ni} are 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 (27) reduces to the form (26).

Maximization of (27) can again be performed using the EM algorithm, as discussed in Appendix C. This has the same form as the EM algorithm for a simple mixture, discussed in Section 3, except that in the E-step, the posterior probability that model (i, j) generated data point $\mathbf{t}_n$ is given by

$$R_{ni,j} = R_{ni} R_{nj\,|\,i}, \tag{28}$$

in which

$$R_{nj|i} = \frac{\pi_{j|i} p(\mathbf{t}_n | i, j)}{\sum_{j'} \pi_{j'|i} p(\mathbf{t}_n | i, j')}. \tag{29}$$

Note that R_{ni} are constants determined from the second level of the hierarchy, and $R_{nj\,|\,i}$ are functions of the "old" parameter values in the EM algorithm. The expression (29) automatically satisfies the relation

$$\sum_{j \in \mathcal{G}_i} R_{ni,j} = R_{ni} \tag{30}$$

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.

The corresponding EM algorithm can be derived by a straightforward extension of the discussion given in Section 3 and Appendix B, and is outlined in Appendix C. This shows that the M-step equations for the mixing coefficients and the means are given by

$$\tilde{\pi}_{j|i} = \frac{\sum_n R_{ni,j}}{\sum_n R_{ni}}, \tag{31}$$

$$\tilde{\boldsymbol{\mu}}_{i,j} = \frac{\sum_n R_{ni,j} \mathbf{t}_n}{\sum_n R_{ni,j}}. \tag{32}$$

The posterior expectations for the missing variables $z_{ni,j}$ are then given by

$$\langle \mathbf{x}_{ni,j} \rangle = \mathbf{M}_{i,j}^{-1} \mathbf{W}_{i,j}^{\mathrm{T}} (\mathbf{t}_n - \tilde{\boldsymbol{\mu}}_{i,j}) \tag{33}$$

$$\langle \mathbf{x}_{ni,j} \mathbf{x}_{ni,j}^{\mathrm{T}} \rangle = \sigma_{i,j}^2 \mathbf{M}_{i,j}^{-1} + \langle \mathbf{x}_{ni,j} \rangle \langle \mathbf{x}_{ni,j}^{\mathrm{T}} \rangle \tag{34}$$

Finally, the $\mathbf{W}_{i,j}$ and $\sigma_{i,j}^2$ are updated using the M-step equations

$$\tilde{\mathbf{W}}_{i,j} = \left[\sum_n R_{ni,j} (\mathbf{t}_n - \tilde{\boldsymbol{\mu}}_{i,j}) \langle \mathbf{x}_{ni,j}^{\mathrm{T}} \rangle \right] \left[\sum_n R_{ni,j} \langle \mathbf{x}_{ni,j} \mathbf{x}_{ni,j}^{\mathrm{T}} \rangle \right]^{-1} \tag{35}$$

$$\begin{aligned} \tilde{\sigma}_{i,j}^2 = \frac{1}{d \sum_n R_{ni,j}} \Bigg\{ & \sum_n R_{ni,j} \left\| \mathbf{t}_n - \tilde{\boldsymbol{\mu}}_{i,j} \right\|^2 \\ & - 2 \sum_n R_{ni,j} \langle \mathbf{x}_{ni,j}^{\mathrm{T}} \rangle \tilde{\mathbf{W}}_{i,j}^{\mathrm{T}} (\mathbf{t}_n - \tilde{\boldsymbol{\mu}}_{i,j}) \\ & + \sum_n R_{ni,j} \mathrm{Tr} \left[\langle \mathbf{x}_{ni,j} \mathbf{x}_{ni,j}^{\mathrm{T}} \rangle \tilde{\mathbf{W}}_{i,j}^{\mathrm{T}} \tilde{\mathbf{W}}_{i,j} \right] \Bigg\}. \end{aligned} \tag{36}$$

Again, we can substitute the E-step equations into the M-step equations to obtain a set of update formulas of the form

$$\tilde{\mathbf{W}}_{i,j} = \mathbf{S}_{i,j} \mathbf{W}_{i,j} \left(\sigma_{i,j}^2 \mathbf{I} + \mathbf{M}_{i,j}^{-1} \mathbf{W}_{i,j}^{\mathrm{T}} \mathbf{S}_{i,j} \mathbf{W}_{i,j} \right)^{-1} \tag{37}$$

$$\tilde{\sigma}_{i,j}^2 = \frac{1}{d} \mathrm{Tr} \left(\mathbf{S}_{i,j} - \mathbf{S}_{i,j} \mathbf{W}_{i,j} \mathbf{M}_{i,j}^{-1} \tilde{\mathbf{W}}_{i,j}^{\mathrm{T}} \right) \tag{38}$$

where all of the summations over n have been expressed in terms of the quantities

$$\mathbf{S}_{i,j} = \frac{1}{N_{i,j}} \sum_n R_{ni,j} (\mathbf{t}_n - \tilde{\boldsymbol{\mu}}_{i,j})(\mathbf{t}_n - \tilde{\boldsymbol{\mu}}_{i,j})^{\mathrm{T}} \tag{39}$$

in which we have defined $N_{i,j} = \sum_n R_{ni,j}$. The $\mathbf{S}_{i,j}$ can again be interpreted as responsibility-weighted covariance matrices.

It is straightforward to extend this hierarchical modeling technique to any desired number of levels, for any parametric family of component distributions.

5 The Visualization Algorithm

So far, we have described the theory behind hierarchical mixtures of latent variable models, and have illustrated the overall form of the visualization hierarchy in Fig. 2. We now complete the description of our algorithm by considering the construction of the hierarchy, and its application to data visualization.

Although the tree structure of the hierarchy can be predefined, a more interesting possibility, with greater practical applicability, is to build the tree *interactively*. Our multilevel visualization algorithm begins by fitting a single latent variable model to the data set, in which the value of $\boldsymbol{\mu}$ is given by the sample mean. For low values of the data space dimensionality d, we can find $\mathbf{W}$ and σ^2 directly by evaluating the covariance matrix and applying (4) and (5). However, for larger values of d, it may be computationally more efficient to apply the EM algorithm, and a scheme for initializing $\mathbf{W}$ and σ^2 is given in Appendix D. Once the EM algorithm has converged, the visualization plot is generated by plotting each data point $\mathbf{t}_n$ at the corresponding posterior mean $\langle \mathbf{x}_n \rangle$ in latent space.

On the basis of this plot, the user then decides on a suitable number of models to fit at the next level down, and selects points $\mathbf{x}^{(i)}$ on the plot corresponding, for example, to the centers of apparent clusters. The resulting points $\mathbf{y}^{(i)}$ in data space, obtained from (1), are then used to initialize the means $\boldsymbol{\mu}_i$ of the respective submodels. To initialize the remaining parameters of the mixture model, we first assign the data points to their nearest mean vector $\boldsymbol{\mu}_i$, and then either compute the corresponding sample covariance matrices and apply a direct eigenvector decomposition, or use the initialization scheme of Appendix D followed by the EM algorithm.

Having determined the parameters of the mixture model at the second level we can then obtain the corresponding set of visualization plots, in which the posterior means $\langle \mathbf{x}_{ni} \rangle$ are again used to plot the data points. For these, it is useful to plot all of the data points on every plot, but to modify the density of "ink" in proportion to the responsibility which each plot has for that particular data point. Thus, if one particular component takes most of the responsibility for a particular point, then that point will effectively be visible only on the corresponding plot. The projection of a data point onto the latent spaces for a mixture of two latent variable models is illustrated schematically in Fig. 3.

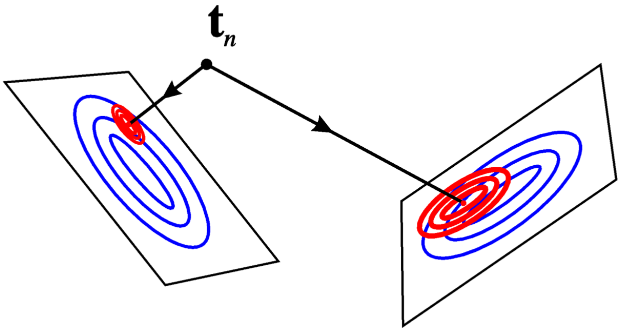


Fig. 3. Illustration of the projection of a data point onto the latent spaces of a mixture of two latent variable models.

The resulting visualization plots are then used to select further submodels, if desired, with the responsibility weighting of (28) being incorporated at this stage. If it is decided not to partition a particular model at some level, then it is easily seen from (30) that the result of training is equivalent to copying the model down unchanged to the next level. Equation (30) further ensures that the combination of such copied models with those generated through further submodeling defines a consistent probability model, such as that represented by the lower three models in Fig. 2. The initialization of the model parameters is by direct analogy with the second-level scheme, with the covariance matrices now also involving the responsibilities R_{ni} as weighting coefficients, as in (23). Again, each data point is in principle plotted on every model at a given level, with a density of "ink" proportional to the corresponding posterior probability, given, for example, by (28) in the case of the third level of the hierarchy.

Deeper levels of the hierarchy involve greater numbers of parameters, and it is therefore important to avoid overfitting and to ensure that the parameter values are well-determined by the data. If we consider principal component analysis, then we see that three (noncolinear) data points are sufficient to ensure that the covariance matrix has rank two and hence that the first two principal components are defined, irrespective of the dimensionality of the data set. In the case of our latent variable model, four data points are sufficient to determine both $\mathbf{W}$ and σ^2. From this, we see that we do not need excessive numbers of data points in each leaf of the tree, and that the dimensionality of the space is largely irrelevant.

Finally, it is often also useful to be able to visualize the spatial relationship between a group of models at one level and their parent at the previous level. This can be done by considering the orthogonal projection of the latent plane in data space onto the corresponding plane of the parent model, as illustrated in Fig. 4. For each model in the hierarchy (except those at the lowest level), we can plot the projections of the associated models from the level below.

In the next section, we illustrate the operation of this algorithm when applied to a simple toy data set, before presenting results from the study of more realistic data in Sections 7 and 8.

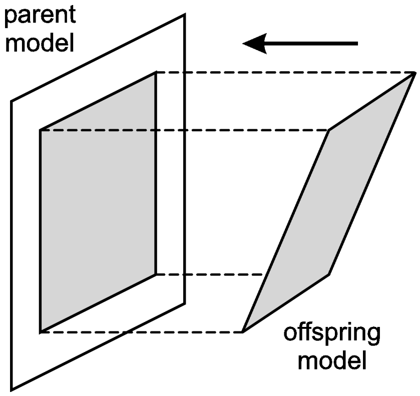


Fig. 4. Illustration of the projection of one of the latent planes onto its parent plane.

6 Illustration Using Toy Data

We first consider a toy data set consisting of 450 data points generated from a mixture of three Gaussians in a three-dimensional space. Each Gaussian is relatively flat (has small variance) in one dimension, and all have the same covariance but differ in their means. Two of these pancake-like clusters are closely spaced, while the third is well separated from the first two. The structure of this data set has been chosen order to illustrate the interactive construction of the hierarchical model.

To visualize the data, we first generate a single top-level latent variable model, and plot the posterior mean of each data point in the latent space. This plot is shown at the top of Fig. 5, and clearly suggests the presence of two distinct clusters within the data. The user then selects two initial cluster centers within the plot, which initialize the second-level. This leads to a mixture of two latent variable models, the latent spaces of which are plotted at the second level in Fig. 5. Of these two plots, that on the right shows evidence of further structure, and so a submodel is generated, again based on a mixture of two latent variable models, which illustrates that there are indeed two further distinct clusters.

At this third step of the data exploration, the hierarchical nature of the approach is evident as the latter two models only attempt to account for the data points which have already been modeled by their immediate ancestor. Indeed, a group of offspring models may be combined with the siblings of the parent and still define a consistent density model. This is illustrated in Fig. 5, in which one of the second level plots has been "copied down" (shown by the dotted line) and combined with the other third-level models. When offspring plots are generated from a parent, the extent of each offspring latent space (i.e., the axis limits shown on the plot) is indicated by a projected rectangle within the parent space, using the approach illustrated in Fig. 4, and these rectangles are numbered sequentially such that the leftmost submodel is "1." In order to display the relative orientations of the latent planes, this number is plotted on the side of the rectangle which corresponds to the *top* of the corresponding offspring plot. The original three clusters have been individually colored, and it can be seen that the red, yellow, and blue data

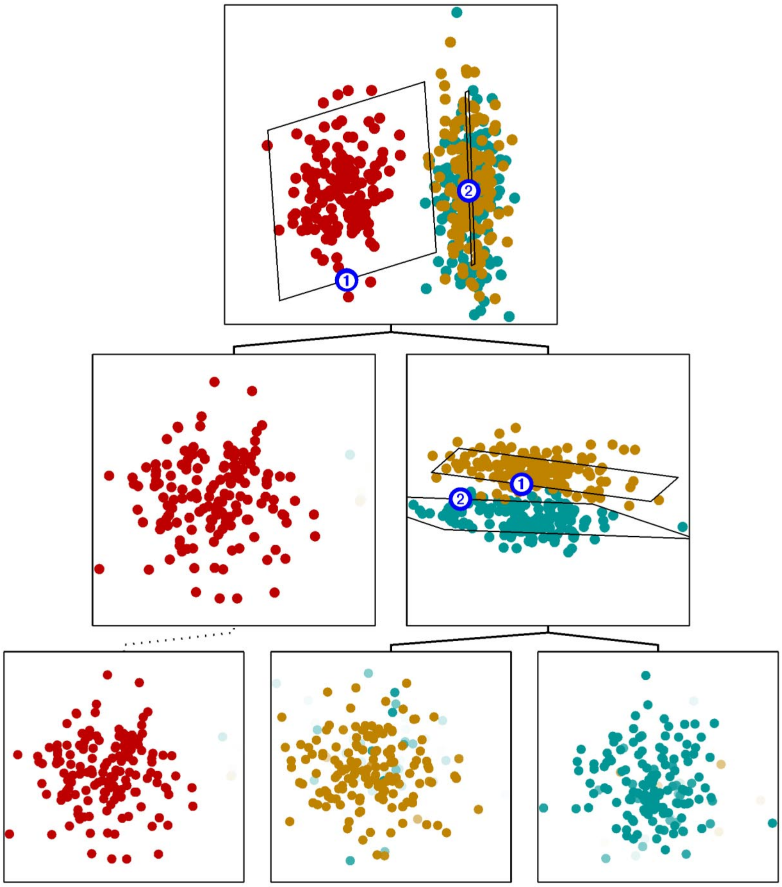


Fig. 5. A summary of the final results from the toy data set. Each data point is plotted on every model at a given level, but with a density of ink which is proportional to the posterior probability of that model for the given data point.

points have been almost perfectly separated in the third level.

7 Oil Flow Data

As an example of a more complex problem, we consider a data set arising from a noninvasive monitoring system used to determine the quantity of oil in a multiphase pipeline containing a mixture of oil, water, and gas [9]. 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 multiphase 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 1,000 points is obtained synthetically by simulating the physical processes in the pipe, including the presence of noise dominated by photon statistics. Locally, the data is expected to have an intrinsic dimensionality of two corresponding to the two 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 flow 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 three-level hierarchical model are shown in Fig. 6.

In the case of the toy data discussed in Section 6, 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 Fig. 6 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, and how 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.

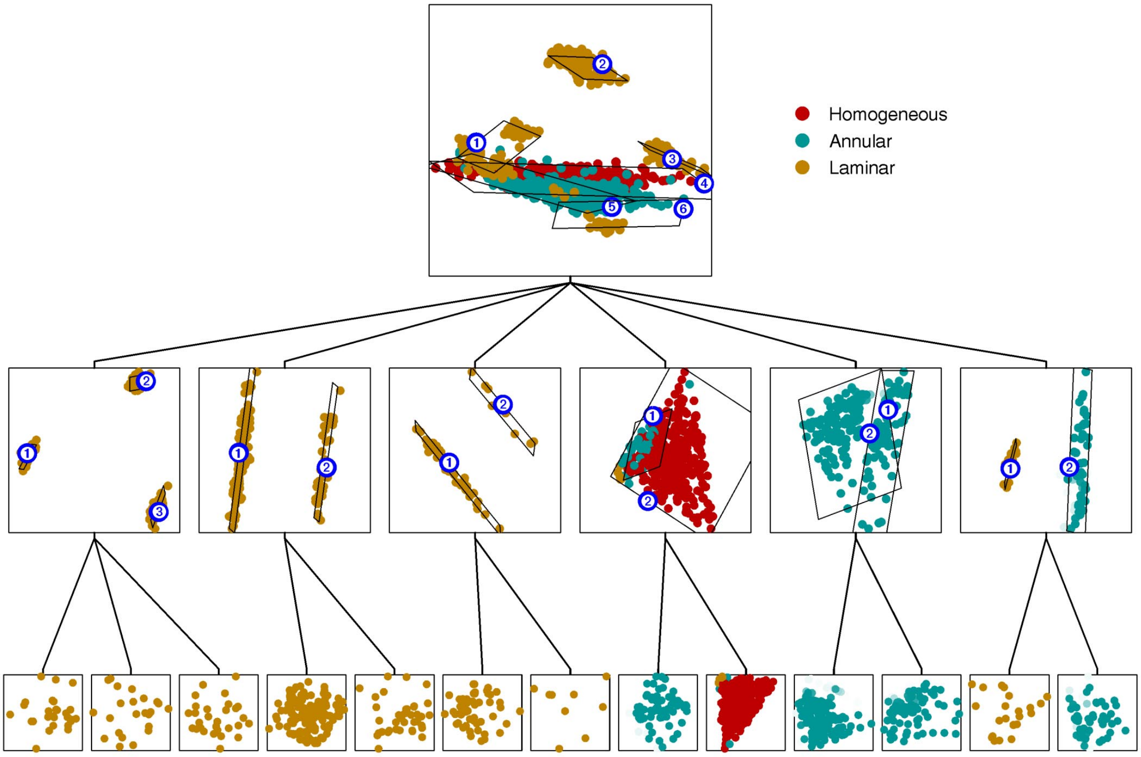


Fig. 6. Results of fitting the oil data. Colors denote different multiphase flow configurations corresponding to homogeneous (red), annular (blue), and laminar (yellow).

8 Satellite Image Data

As a final example, we consider the visualization of a data set obtained from remote-sensing satellite images. Each data point represents a 3 × 3 pixel region of a satellite land image, and, for each pixel, there are four measurements of intensity taken at different wavelengths (approximately red and green in the visible spectrum, and two in the near infrared). This gives a total of 36 variables for each data point. There is also a label indicating the type of land represented by the central pixel. This data set has previously been the subject of a classification study within the STATLOG project [10].

We applied the hierarchical visualization algorithm to 600 data points, with 100 drawn at random of each of six classes in the 4,435-point data set. The result of fitting a three-level hierarchy is shown in Fig. 7. Note that the class labels are used only to color the data points and play no role in the maximum likelihood determination of the model parameters. Fig. 7 illustrates that the data can be approximately separated into classes, and the "gray soil" → "damp gray soil" → "very damp gray soil" continuum is clearly evident in component 3 at the second level. One particularly interesting additional feature is that there appear to be two distinct and separated clusters of "cotton crop" pixels, in mixtures 1 and 2 at the second level, which are not evident in the single top-level projection. Study of the original image [10] indeed indicates that there are *two* separate areas of "cotton crop."

9 Discussion

We have presented a novel approach to data visualization which is both statistically principled and which, as illustrated by real examples, can be very effective at revealing structure within data. The hierarchical summaries of Figs. 5, 6, and 7 are relatively simple to interpret, yet still convey considerable structural information.

It is important to emphasize that in data visualization there is no objective measure of quality, and so it is difficult to quantify the merit of a particular data visualization technique. This is one reason, no doubt, why there is a multitude of visualization algorithms and associated software available. While the effectiveness of many of these techniques is often highly data-dependent, we would expect the hierarchical visualization model to be a very useful tool for the visualization and exploratory analysis of data in many applications.

In relation to previous work, the concept of subsetting, or isolating, data points for further investigation can be traced back to Maltson and Dammann [11], and was further developed by Friedman and Tukey [12] for exploratory data analysis in conjunction with projection pursuit. Such subsetting operations are also possible in current dynamic visualization software, such as "XGobi" [13]. However, in these approaches there are two limitations. First, the partitioning of the data is performed in a *hard* fashion, while the mixture of latent variable models approach discussed in this paper permits a *soft* partitioning in which data points can effectively belong to more than one cluster at any given level. Second, the mechanism for the partitioning of the data is prone to suboptimality as the clusters must be fixed

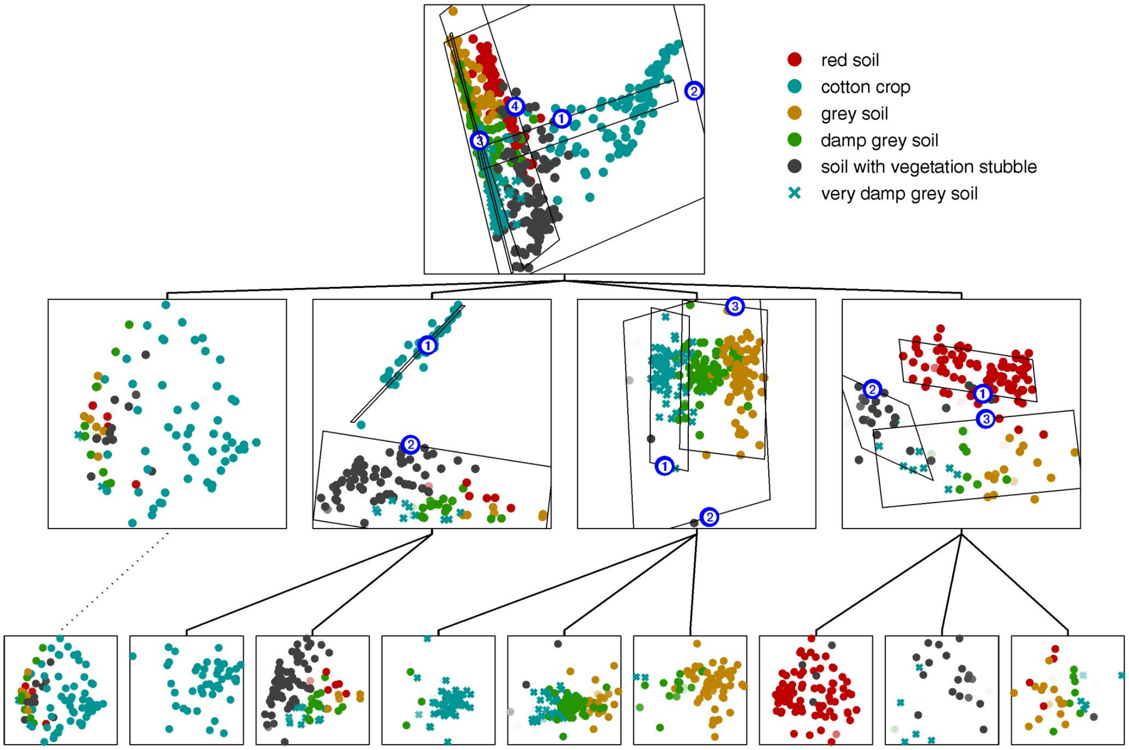


Fig. 7. Results of fitting the satellite image data.

by the user based on a single two-dimensional projection. In the hierarchical approach advocated in this paper, the user selects only a "first guess" for the cluster centers in the mixture model. The EM algorithm is then utilized to determine the parameters which maximize the likelihood of the model, thus allowing both the centers and the widths of the clusters to adapt to the data in the full multidimensional data space. There is also some similarity between our method and earlier hierarchical methods in script recognition [14] and motion planning [15] which incorporate the Kohonen Self-Organizing Feature Map [16] and so offer the potential for visualization. As well as again performing a hard clustering, a key distinction in both of these approaches is that different levels in the hierarchies operate on different subsets of input variables and their operation is thus quite different from the hierarchical algorithm described in this paper.

Our model is based on a hierarchical combination of linear latent variable models. A related latent variable technique called the *generative topographic mapping* (GTM) [17] uses a nonlinear transformation from latent space to data space and is again optimized using an EM algorithm. It is straightforward to incorporate GTM in place of the linear latent variable models in the current hierarchical framework.

As described, our model applies to continuous data variables. We can easily extend the model to handle discrete data as well as combinations of discrete and continuous variables. In case of a set of binary data variables $y_k \in \{0, 1\}$, we can express the conditional distribution of a binary variable, given $\mathbf{x}$, using a binomial distribution of the form $p(\mathbf{t}|\mathbf{x}) = \prod_k \sigma\left(\mathbf{w}_k^{\mathrm{T}}\mathbf{x} + \mu_k\right)^{t_k}\left[1 - \sigma\left(\mathbf{w}_k^{\mathrm{T}}\mathbf{x} + \mu_k\right)\right]^{1-t_k}$, where $\sigma(a) = (1 + \exp(-a))^{-1}$ is the logistic sigmoid function, and $\mathbf{w}_k$ is the kth column of $\mathbf{W}$. For data having a 1-of-D coding scheme we can represent the distribution of data variables using a multinomial distribution of the form $p(\mathbf{t}|\mathbf{x}) = \prod_{k=1}^{D} m_k^{x_k}$ where m_k are defined by a softmax, or normalized exponential, transformation of the form

$$m_k = \frac{\exp\left(\mathbf{w}_k^{\mathrm{T}}\mathbf{x} + \mu_k\right)}{\sum_j \exp\left(\mathbf{w}_j^{\mathrm{T}}\mathbf{x} + \mu_j\right)}. \tag{40}$$

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 $p(\mathbf{t}\,|\,\mathbf{x})$ as a product of Gaussian, binomial and multinomial distributions as appropriate. The E-step of the EM algorithm now becomes more complex since the marginalization over the latent variables, needed to normalize the posterior distribution in latent space, will in general be analytically intractable. One approach is to approximate the integration using a finite sample of points drawn from the prior [17]. Similarly, the M-step is more complex, although it can be tackled efficiently using the iterative reweighted least squares (IRLS) algorithm [18].

One important consideration with the present model is that the parameters are determined by maximum likelihood, and this criterion need not always lead to the most interesting visualization plots. We are currently investigating alternative models which optimize other criteria such as the separation of clusters. Other possible refinements include algorithms which allow a self-consistent fitting of the whole tree, so that lower levels have the opportunity to influence the parameters at higher levels. While the user-driven nature of the current algorithm is highly appropriate for the visualization context, the development of an automated procedure for generating the hierarchy would clearly also be of interest.

A software implementation of the probabilistic hierarchical visualization algorithm in MATLAB is available from:
http://www.ncrg.aston.ac.uk/PhiVis

APPENDIX A
PROBABILISTIC PRINCIPAL COMPONENT ANALYSIS AND EM

The algorithm discussed in this paper is based on a latent variable model corresponding to a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$, in which the parameters of the model, given by $\boldsymbol{\mu}$, $\mathbf{W}$, and σ^2 are determined by maximizing the likelihood function given by (6). For a single such model, the solution for the mean $\boldsymbol{\mu}$ is given by the sample mean of the data set. We can express the solutions for $\mathbf{W}$ and σ^2 in closed form in terms of the eigenvectors and eigenvalues of the sample covariance matrix, as discussed in Section 2. Here we derive an alternative approach based on the EM (expectation-maximization) algorithm. We first regard the variables $\mathbf{x}_n$ appearing in (6) as "missing data." If these quantities were known, then the corresponding "complete data" log likelihood function would be given by

$$L_{\mathrm{C}} = \sum_{n=1}^{N} \ln p(\mathbf{t}_n, \mathbf{x}_n) = \sum_{n=1}^{N} \ln p(\mathbf{t}_n|\mathbf{x}_n)p(\mathbf{x}_n). \qquad (41)$$

We do not, of course, know the values of the $\mathbf{x}_n$, but we can find their posterior distribution using Bayes' theorem in the form

$$p(\mathbf{x}|\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{t})}. \qquad (42)$$

Since $p(\mathbf{t} \mid \mathbf{x})$ is Gaussian with mean $\mathbf{W}\mathbf{x} + \boldsymbol{\mu}$ and covariance $\sigma^2\mathbf{I}$, and $p(\mathbf{x})$ is Gaussian with zero mean and unit variance, it follows by completing the square that $p(\mathbf{x} \mid \mathbf{t})$ is also Gaussian with mean given by $\mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{t}_n - \boldsymbol{\mu})$, and covariance given by $\sigma^2\mathbf{M}^{-1}$, where we have defined $\mathbf{M} = \mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2\mathbf{I}$.

We can then compute the expectation of L_{C} with respect to this posterior distribution to give

$$\langle L_{\mathrm{C}} \rangle = \sum_{n=1}^{N} \left\{ -\frac{d}{2}\ln\sigma^2 - \frac{1}{2}\mathrm{Tr}\left(\left\langle \mathbf{x}_n\mathbf{x}_n^{\mathrm{T}} \right\rangle\right) - \frac{1}{2\sigma^2}\left\|\mathbf{t}_n - \boldsymbol{\mu}\right\|^2 \right.$$
$$\left. + \frac{1}{\sigma^2}\left\langle \mathbf{x}_n^{\mathrm{T}} \right\rangle \mathbf{W}^{\mathrm{T}}(\mathbf{t}_n - \boldsymbol{\mu}) - \frac{1}{2\sigma^2}\mathrm{Tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\left\langle \mathbf{x}_n\mathbf{x}_n^{\mathrm{T}} \right\rangle\right) \right\} \qquad (43)$$

which corresponds to the E-step of the EM algorithm.

The M-step corresponds to the maximization of $\langle L_{\mathrm{C}} \rangle$ with respect to $\mathbf{W}$ and σ^2, for fixed $\langle \mathbf{x}_n \rangle$ and $\left\langle \mathbf{x}_n\mathbf{x}_n^{\mathrm{T}} \right\rangle$. This is straightforward, and gives the results (9) and (10). A simple proof of convergence for the EM algorithm is given in Appendix E.

An important aspect of our algorithm is the choice of an isotropic covariance matrix for the noise model of the form $\sigma^2\mathbf{I}$. The maximum likelihood solution for $\mathbf{W}$ is given by the scaled principal component eigenvectors of the data set, in the form

$$\mathbf{W} = \mathbf{U}_q\,(\boldsymbol{\Lambda}_q - \sigma^2\mathbf{I})^{1/2}\mathbf{R} \qquad (44)$$

where $\mathbf{U}_q$ is a $d \times q$ matrix whose columns are the eigenvectors of the data covariance matrix corresponding to the q largest eigenvalues (where q is the dimensionality of the latent space, so that $q = 2$ in our model), and $\boldsymbol{\Lambda}_q$ is a $q \times q$ diagonal matrix whose elements are given by the eigenvalues. The matrix $\mathbf{R}$ is an arbitrary orthogonal matrix corresponding to a rotation of the axes in latent space. This result is derived and discussed in [5], and shows that the image of the latent plane in data space coincides with the principal components plane.

Also, for $\sigma^2 \to 0$, the projection of data points onto the latent plane, defined by the posterior means $\langle \mathbf{x}_n \rangle$, coincides with the principal components projection. To see this we note that when a point $\mathbf{x}_n$ in latent space is projected onto a point $\mathbf{W}\mathbf{x}_n + \boldsymbol{\mu}$ in data space, the squared distance between the projected point and a data point $\mathbf{t}_n$ is given by

$$\|\mathbf{W}\mathbf{x}_n + \boldsymbol{\mu} - \mathbf{t}_n\|^2. \qquad (45)$$

If we minimize this distance with respect to $\mathbf{x}_n$ we obtain a solution for the orthogonal projection of $\mathbf{t}_n$ onto the plane defined by $\mathbf{W}$ and $\boldsymbol{\mu}$, given by $\mathbf{W}\tilde{\mathbf{x}}_n + \boldsymbol{\mu}$ where

$$\tilde{\mathbf{x}}_n = \left(\mathbf{W}^{\mathrm{T}}\mathbf{W}\right)^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{t}_n - \boldsymbol{\mu}). \qquad (46)$$

We see from (7) that, in the limit $\sigma^2 \to 0$, the posterior mean for a data point $\mathbf{t}_n$ reduces to (46) and hence the corresponding point $\mathbf{W}\langle \mathbf{x}_n \rangle + \boldsymbol{\mu}$ is given by the orthogonal projection of $\mathbf{t}_n$ onto the plane defined by (1). For $\sigma^2 \neq 0$, the posterior mean is skewed towards the origin by the prior, and hence the projection $\mathbf{W}\tilde{\mathbf{x}}_n + \mu$ is shifted toward $\boldsymbol{\mu}$.

The crucial difference between our latent variable model and principal component analysis is that, unlike PCA, our model defines a probability density, and hence allows us to consider mixtures, and indeed hierarchical mixtures, of models in a probabilistically principled manner.

APPENDIX B
EM FOR MIXTURES OF PRINCIPAL COMPONENT ANALYZERS

At the second level of the hierarchy we must fit a mixture of latent variable models, in which the overall model distribution takes the form

$$p(\mathbf{t}) = \sum_{i=1}^{M_0} \pi_i p(\mathbf{t}|i), \qquad (47)$$

where $p(\mathbf{t} \mid i)$ is a single latent variable model of the form discussed in Appendix A and π_i is the corresponding mixing proportion. The parameters for this mixture model can be determined by an extension of the EM algorithm. We begin by considering the standard form which the EM algorithm would take for this model and highlight a number of limitations. We then show that a two-stage form of EM leads to a much more efficient algorithm.

We first note that in addition to a set of $\mathbf{x}_{ni}$ for each model i, the missing data includes variables z_{ni} labeling which model is responsible for generating each data point $\mathbf{t}_n$. At this point, we can derive a standard EM algorithm by considering the corresponding complete-data log likelihood which takes the form

$$L_{\mathrm{C}} = \sum_{n=1}^{N} \sum_{i=1}^{M_0} z_{ni} \ln\left\{\pi_i p\left(\mathbf{t}_n, \mathbf{x}_{ni}\right)\right\}. \tag{48}$$

Starting with "old" values for the parameters π_i, $\boldsymbol{\mu}_i$, $\mathbf{W}_i$, and σ_i^2 we first evaluate the posterior probabilities R_{ni} using (14) and similarly evaluate the expectations $\langle \mathbf{x}_{ni} \rangle$ and $\left\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}} \right\rangle$ using (17) and (18) which are easily obtained by inspection of (7) and (8). Then we take the expectation of L_{C} with respect to this posterior distribution to obtain

$$\begin{aligned}\left\langle\left\langle L_{\mathrm{C}}\right\rangle\right\rangle = \sum_{n=1}^{N} \sum_{i=1}^{M_0} R_{ni} \Bigg\{ & \ln \pi_i - \frac{d}{2}\ln\sigma_i^2 - \frac{1}{2}\mathrm{Tr}\left(\left\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}} \right\rangle\right) \\ & - \frac{1}{2\sigma_i^2}\left\|\mathbf{t}_{ni} - \boldsymbol{\mu}_i\right\|^2 + \frac{1}{\sigma_i^2}\left\langle \mathbf{x}_{ni}^{\mathrm{T}} \right\rangle \mathbf{W}_i^{\mathrm{T}}\left(\mathbf{t}_n - \boldsymbol{\mu}_i\right) \\ & - \frac{1}{2\sigma_i^2}\mathrm{Tr}\left(\mathbf{W}_i^{\mathrm{T}}\mathbf{W}_i\left\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}} \right\rangle\right)\Bigg\} + \text{const.}\end{aligned} \tag{49}$$

where $\langle\langle \cdot \rangle\rangle$ denotes the expectation with respect to the posterior distributions of both $\mathbf{x}_{ni}$ and z_{ni}. The M-step then involves maximizing (49) with respect to π_i, $\boldsymbol{\mu}_i$, σ_i^2, and $\mathbf{W}_i$ to obtain "new" values for these parameters. The maximization with respect to π_i must take account of the constraint that $\sum_i \pi_i = 1$. This can be achieved with the use of a Lagrange multiplier λ [8] by maximizing

$$\left\langle\left\langle L_{\mathrm{C}}\right\rangle\right\rangle + \lambda\left(\sum_{i=1}^{M_0}\pi_i - 1\right). \tag{50}$$

Together with the results of maximizing (48) with respect to the remaining parameters, this gives the following M-step equations

$$\tilde{\pi}_i = \frac{1}{N}\sum_n R_{ni} \tag{51}$$

$$\tilde{\boldsymbol{\mu}}_i = \frac{\sum_n R_{ni}\left(\mathbf{t}_{ni} - \tilde{\mathbf{W}}_i\left\langle \mathbf{x}_{ni} \right\rangle\right)}{\sum_n R_{ni}} \tag{52}$$

$$\tilde{\mathbf{W}}_i = \left[\sum_n R_{ni}\left(\mathbf{t}_n - \tilde{\boldsymbol{\mu}}_i\right)\left\langle \mathbf{x}_{ni}^{\mathrm{T}} \right\rangle\right]\left[\sum_n R_{ni}\left\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}} \right\rangle\right]^{-1} \tag{53}$$

$$\begin{aligned}\tilde{\sigma}_i^2 = \frac{1}{d\sum_n R_{ni}}\Bigg\{ & \sum_n R_{ni}\left\|\mathbf{t}_n - \tilde{\boldsymbol{\mu}}_i\right\|^2 \\ & - 2\sum_n R_{ni}\left\langle \mathbf{x}_{ni}^{\mathrm{T}} \right\rangle \tilde{\mathbf{W}}_i^{\mathrm{T}}\left(\mathbf{t}_n - \tilde{\mu}_i\right) + \sum_n R_{ni}\mathrm{Tr}\left(\left\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}} \right\rangle \tilde{\mathbf{W}}_i^{\mathrm{T}}\tilde{\mathbf{W}}_i\right)\Bigg\}.\end{aligned} \tag{54}$$

Note that the M-step equations for $\tilde{\boldsymbol{\mu}}_i$ and $\tilde{\mathbf{W}}_i$, given by (52) and (53), are coupled, and so further manipulation is required to obtain explicit solutions. In fact, a simplification of the M-step equations, along with improved speed of convergence, is possible if we adopt a two-stage EM procedure as follows.

The likelihood function we wish to maximize is given by

$$L = \sum_{n=1}^{N}\ln\left\{\sum_{i=1}^{M_0}\pi_i p\left(\mathbf{t}_n \middle| i\right)\right\}. \tag{55}$$

Regarding the component labels z_{ni} as missing data, we can consider the corresponding expected complete-data log likelihood given by

$$\left\langle \hat{L}_{\mathrm{C}} \right\rangle = \sum_{n=1}^{N}\sum_{i=1}^{M_0} R_{ni}\ln\left\{\pi_i p\left(\mathbf{t}_n \middle| i\right)\right\}, \tag{56}$$

where R_{ni} represent the posterior probabilities (corresponding to the expected values of z_{ni}) and are given by (14). Maximization of (56) with respect to π_i, again using a Lagrange multiplier, gives the M-step equation (15). Similarly, maximization of (56) with respect to $\boldsymbol{\mu}_i$ gives (16).

In order to update $\mathbf{W}_i$ and σ_i^2, we seek only to increase the value of $\left\langle \hat{L}_{\mathrm{C}} \right\rangle$ and not actually to maximize it. This corresponds to the generalized EM (or GEM) algorithm. We do this by treating the labels z_{ni} as missing data and performing one cycle of the EM algorithm. This involves using the new values $\tilde{\boldsymbol{\mu}}_i$ to compute the sufficient statistics of the posterior distribution of $\mathbf{x}_{ni}$ using (17) and (18). The advantage of this strategy is that we are using the new rather than old values of $\boldsymbol{\mu}_i$ in computing these statistics, and overall this leads to simplifications to the algorithm as well as improved convergence speed. By inspection of (49) we see that the expected complete-data log likelihood takes the form

$$\begin{aligned}\left\langle\left\langle L_{\mathrm{C}}\right\rangle\right\rangle = \sum_{n=1}^{N}\sum_{i=1}^{M_0} R_{ni}\Bigg\{ & \ln \pi_i - \frac{d}{2}\ln\sigma_i^2 - \frac{1}{2}\mathrm{Tr}\left(\left\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}} \right\rangle\right) \\ & - \frac{1}{2\sigma_i^2}\left\|\mathbf{t}_{ni} - \tilde{\boldsymbol{\mu}}_i\right\|^2 + \frac{1}{\sigma_i^2}\left\langle \mathbf{x}_{ni}^{\mathrm{T}} \right\rangle \mathbf{W}_i^{\mathrm{T}}\left(\mathbf{t}_n - \tilde{\boldsymbol{\mu}}_i\right) \\ & - \frac{1}{2\sigma_i^2}\mathrm{Tr}\left(\mathbf{W}_i^{\mathrm{T}}\mathbf{W}_i\left\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}} \right\rangle\right)\Bigg\}.\end{aligned} \tag{57}$$

We then maximize (57) with respect to $\mathbf{W}_i$ and σ_i^2 (keeping $\tilde{\boldsymbol{\mu}}_i$ fixed). This gives the M-step equations (19) and (20).

APPENDIX C
EM FOR HIERARCHICAL MIXTURE MODELS

In the case of the third and subsequent levels of the hierarchy we have to maximize a likelihood function of the form (27) in which the R_{ni} and the π_i are treated as constants. To obtain an EM algorithm we note that the likelihood function can be written as

$$L = \sum_{i=1}^{M_0} L_i \quad \text{where} \quad L_i = \sum_{n=1}^{N} R_{ni} \ln\left\{\pi_i \sum_{j \in \mathcal{G}_i} \pi_{j|i} p(\mathbf{t}|i,j)\right\}. \quad (58)$$

Since the parameters for different values of i are independent this represents M_0 independent models each of which can be fitted separately, and each of which corresponds to a mixture model but with weighting coefficients R_{ni}. We can then derive the EM algorithm by introducing, for each i, the expected complete-data likelihood in the form

$$\langle L_{iC} \rangle = \sum_{n=1}^{N} R_{ni} \sum_{j \in \mathcal{G}_i} R_{nj|i} \ln\left\{\pi_{j|i} p(\mathbf{t}|i,j)\right\} \quad (59)$$

where $R_{nj\,|\,i}$ is defined by (29) and we have omitted the constant term involving π_i. Thus, the responsibility of the jth submodel in group $\mathcal{G}_i$ for generating data point $\mathbf{t}_n$ is effectively weighted by the responsibility of its parent model. Maximization of (59) gives rise to weighted M-step equations for the $\mathbf{W}_{i,j}$, $\boldsymbol{\mu}_{i,j}$, and $\sigma^2_{i,j}$ parameters with weighting factors $R_{ni,j}$ given by (28), as discussed in the text. For the mixing coefficients $\pi_{j\,|\,i}$, we can introduce a Lagrange multiplier λ_i, and hence maximize the function

$$\sum_{n=1}^{N} R_{ni} \sum_{j \in \mathcal{G}_i} R_{nj|i} \ln \pi_{j|i} + \lambda_i \left(\sum_j \pi_{j|i} - 1 \right) \quad (60)$$

to obtain the M-step result (31).

A final consideration is that while each offspring mixture within the hierarchy is fitted to the *entire* data set, the responsibilities of its parent model for many of the data points will approach zero. This implies that the weighted responsibilities for the component models of the mixture will likewise be at least as small. Thus, in a practical implementation, we need only fit offspring mixture models to a *reduced* data set, where data points for which the parental responsibility is less than some threshold are discarded. For reasons of numerical accuracy, this threshold should be no smaller than the machine precision (which is 2.22×10^{-16} for double-precision arithmetic). We adopted such a threshold for the experiments within this paper, and observed a considerable computational advantage, particularly at deeper levels in the hierarchy.

APPENDIX D
INITIALIZATION OF THE EM ALGORITHM

Here we outline a simple procedure for initializing $\mathbf{W}$ and σ^2 before applying the EM algorithm. Consider a covariance matrix $\mathbf{S}$ with eigenvalues $\mathbf{u}_j$ and eigenvalues λ_j. An arbitrary vector $\mathbf{v}$ will have an expansion in the eigenbasis of the form $\mathbf{v} = \sum_j v_j \mathbf{u}_j$, where $v_j = \mathbf{v}^{\mathrm{T}}\mathbf{u}_j$. If we multiply $\mathbf{v}$ by $\mathbf{S}$, we obtain a vector $\sum_j \lambda_j v_j \mathbf{u}_j$ which will tend to be dominated by the eigenvector $\mathbf{u}_1$ with the largest eigenvalue λ_1. Repeated multiplication and normalization will give an increasingly improved estimate of the normalized eigenvector and of the corresponding eigenvalue. In order to find the first two eigenvectors and eigenvalues, we start with a random $d \times 2$ matrix $\mathbf{V}$ and after each multiplication we orthonormalize the columns of $\mathbf{V}$. We choose two data points at random and, after subtraction of $\boldsymbol{\mu}$, use these as the columns of $\mathbf{V}$ to provide a starting point for this procedure. Degenerate eigenvalues do not present a problem since any two orthogonal vectors in the principal subspace will suffice. In practice only a few matrix multiplications are required to obtain a suitable initial estimate. We now initialize $\mathbf{W}$ using the result (4), and initialize σ^2 using (5). In the case of mixtures we simply apply this procedure for each weighted covariance matrix $\mathbf{S}_i$ in turn.

As stated this procedure appears to require the evaluation of $\mathbf{S}$, which would take $O(Nd^2)$ computational steps and would therefore defeat the purpose of using the EM algorithm. However, we only ever need to evaluate the product of $\mathbf{S}$ with some vector, which can be performed in $O(Nd)$ steps by rewriting the product as

$$\mathbf{S}\mathbf{v} = \sum_{n=1}^{N} (\mathbf{t}_n - \boldsymbol{\mu})\left[(\mathbf{t}_n - \boldsymbol{\mu})^{\mathrm{T}} \mathbf{v}\right] \quad (61)$$

and evaluating the inner products before performing the summation over n. Similarly the trace of $\mathbf{S}$, required to initialize σ^2, can also be obtained in $O(Nd)$ steps.

APPENDIX E
CONVERGENCE OF THE EM ALGORITHM

Here we give a very simple demonstration that the EM algorithms of the kind discussed in this paper have the desired property of guaranteeing that the likelihood will be increased at each cycle of the algorithm unless the parameters correspond to a (local) maximum of the likelihood. If we denote the set of observed data by D, then the log likelihood which we wish to maximize is given by

$$L = p(D \mid \boldsymbol{\theta}) \quad (62)$$

where $\boldsymbol{\theta}$ denotes the set of parameters of the model. If we denote the missing data by M, then the complete-data log likelihood function, i.e., the likelihood function which would be applicable if M were actually observed, is given by

$$L_{\mathrm{C}} = \ln p(D, M \mid \boldsymbol{\theta}). \quad (63)$$

In the E-step of the EM algorithm, we evaluate the posterior distribution of M given the observed data D and some current values $\boldsymbol{\theta}_{\mathrm{old}}$ for the parameters. We then use this distribution to take the expectation of L_{C}, so that

$$\langle L_{\mathrm{C}}(\boldsymbol{\theta}) \rangle = \int \ln\{p(D, M \mid \boldsymbol{\theta})\} p(M \mid D, \boldsymbol{\theta}_{\mathrm{old}}) dM. \quad (64)$$

In the M-step, the quantity $\langle L_{\mathrm{C}}(\boldsymbol{\theta}) \rangle$ is maximized with respect to $\boldsymbol{\theta}$ to give $\boldsymbol{\theta}_{\mathrm{new}}$. From the rules of probability we have

$$p(D, M \mid \boldsymbol{\theta}) = p(M \mid D, \boldsymbol{\theta})\, p(D \mid \boldsymbol{\theta}) \tag{65}$$

and substituting this into (64) gives

$$\langle L_{\mathrm{C}}(\boldsymbol{\theta})\rangle = \ln p(D \mid \boldsymbol{\theta}) + \int \ln \{p(M \mid D, \boldsymbol{\theta})\}\, p(M \mid D, \boldsymbol{\theta}_{\mathrm{old}})dM. \tag{66}$$

The change in the likelihood function in going from old to new parameter values is therefore given by

$$\ln p(D \mid \boldsymbol{\theta}_{\mathrm{new}}) - \ln p(D \mid \boldsymbol{\theta}_{\mathrm{old}}) = \langle L_{\mathrm{C}}(\boldsymbol{\theta}_{\mathrm{new}})\rangle - \langle L_{\mathrm{C}}(\boldsymbol{\theta}_{\mathrm{old}})\rangle$$

$$-\int \ln\left\{\frac{p\left(M \middle| D, \boldsymbol{\theta}_{\mathrm{new}}\right)}{p\left(M \middle| D, \boldsymbol{\theta}_{\mathrm{old}}\right)}\right\} p\left(M \middle| D, \boldsymbol{\theta}_{\mathrm{old}}\right) dM\,. \tag{67}$$

The final term on the right-hand side of (67) is the Kullback-Leibler divergence between the old and new posterior distributions. Using Jensen's inequality it is easily shown that $KL(\boldsymbol{\theta} \,\|\boldsymbol{\theta}_{\mathrm{old}}) \geq 0$ [8]. Since we have maximized $\langle L_{\mathrm{C}}\rangle$ (or more generally just increased its value in the case of the GEM algorithm) in going from $\boldsymbol{\theta}_{\mathrm{old}}$ to $\boldsymbol{\theta}_{\mathrm{new}}$, we see that $p(D \mid \boldsymbol{\theta}_{\mathrm{new}}) > p(D \mid \boldsymbol{\theta}_{\mathrm{old}})$ as required.

Acknowledgments

This work was supported by EPSRC grant GR/K51808: *Neural Networks for Visualization of High-Dimensional Data.* We are grateful to Michael Jordan for useful discussions, and we would like to thank the Isaac Newton Institute in Cambridge for their hospitality.

References

[1] M.I. Jordan and R.A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181-214, 1994.

[2] B.S. Everitt, *An Introduction to Latent Variable Models*. London: Chapman and Hall, 1984.

[3] W.J. Krzanowski and F.H.C. Marriott, *Multivariate Analysis Part 2: Classification, Covariance Structures and Repeated Measurements*. London: Edward Arnold, 1994.

[4] M.E. Tipping and C.M. Bishop, "Mixtures of Principal Component Analysers," *Proc. IEE Fifth Int'l Conf. Artificial Neural Networks,* pp. 13-18, Cambridge, U.K., July 1997.

[5] M.E. Tipping and C.M. Bishop, "Mixtures of Probabilistic Principal Component Analysers," Tech. Rep. NCRG/97/003, Neural Computing Research Group, Aston University, Birmingham, U.K., 1997.

[6] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood From Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc., B*, vol. 39, no. 1, pp. 1-38, 1977.

[7] D.B. Rubin and D.T. Thayer, "EM Algorithms for ML Factor Analysis," *Psychometrika*, vol. 47, no. 1, pp. 69-76, 1982.

[8] C.M. Bishop, *Neural Networks for Pattern Recognition*. Oxford Univ. Press, 1995.

[9] C.M. Bishop and G.D. James, "Analysis of Multiphase Flows Using Dual-Energy Gamma Densitometry and Neural Networks," *Nuclear Instruments and Methods in Physics Research*, vol. A327, pp. 580-593, 1993.

[10] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, *Machine Learning, Neural and Statistical Classification*. New York: Ellis Horwood, 1994.

[11] R.L. Maltson and J.E. Dammann, "A Technique for Determining and Coding Subclasses in Pattern Recognition Problems," *IBM J.*, vol. 9, pp. 294-302, 1965.

[12] J.H. Friedman and J.W. Tukey, "A Projection Pursuit Algorithm for Exploratory Data Analysis," *IEEE Trans. Computers*, vol. 23, pp. 881-889, 1974.

[13] A. Buja, D. Cook, and D.F. Swayne, "Interactive High-Dimensional Data Visualization," *J. Computational and Graphical Statistics*, vol. 5, no. 1, pp. 78-99, 1996.

[14] R. Miikkulainen, "Script Recognition With Hierarchical Feature Maps," *Connection Science*, vol. 2, pp. 83-101, 1990.

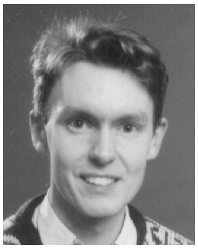
[15] C. Versino and L.M. Gambardella, "Learning Fine Motion by Using the Hierarchical Extended Kohonen Map," *Artificial Neural Networks—ICANN* 96, C. von der Malsburg, W. von Seelen, J.C. Vorbrüggen, and B. Sendhoff, eds., *Lecture Notes in Computer Science*, vol. 1,112, pp. 221-226. Berlin: Springer-Verlag, 1996.

[16] T. Kohonen, *Self-Organizing Maps*. Berlin: Springer-Verlag, 1995.

[17] C.M. Bishop, M. Svensén, and C.K.I. Williams, "GTM: The Generative Topographic Mapping," *Neural Computation*, vol. 10, no. 1, pp. 215-234, 1998.

[18] P. McCullagh and J.A. Nelder, *Generalized Linear Models,* 2nd ed. Chapman and Hall, 1989.

Christopher M. Bishop graduated from the University of Oxford in 1980 with First Class Honors in Physics and obtained a PhD from the University of Edinburgh in quantum field theory in 1983. After a period at Culham Laboratory researching the theory of magnetically confined plasmas for the fusion program, he developed an interest in statistical pattern recognition, and became head of the Applied Neurocomputing Center at Harwell Laboratory. In 1993, he was appointed to a chair in the Department of Computer Science and Applied Mathematics at Aston University, and he was the principal organizer of the six-month program on Neural Networks and Machine Learning at the Isaac Newton Institute for Mathematical Sciences in Cambridge in 1997. Recently, he moved to the Microsoft Research Laboratory in Cambridge and has also been elected to a chair of computer science at the University of Edinburgh. His current research interests include probabilistic inference, graphical models, and pattern recognition.



Michael E. Tipping received the BEng degree in electronic engineering from Bristol University in 1990 and the MSc degree in artificial intelligence from the University of Edinburgh in 1992. He received the PhD degree in neural computing from Aston University in 1996.

He has been a research fellow in the Neural Computing Research Group at Aston University since March 1996, and his research interests include neural networks, data visualization, probabilistic modeling, statistical pattern recognition, and topographic mapping.