MIXTURES OF PRINCIPAL COMPONENT ANALYZERS

Michael E. Tipping and Christopher M. Bishop

Neural Computing Research Group, Dept of Computer Science & Applied Mathematics, Aston University, Birmingham, UK

ABSTRACT

Principal component analysis (PCA) is a ubiquitous technique for data analysis but one whose effective application is restricted by its global linear character. While global nonlinear variants of PCA have been proposed, an alternative paradigm is to capture data nonlinearity by a mixture of local PCA models. However, existing techniques are limited by the absence of a probabilistic formalism with an appropriate likelihood measure and so require an arbitrary choice of implementation strategy. This paper shows how PCA can be derived from a maximum–likelihood procedure, based on a specialisation of factor analysis. This is then extended to develop a well–defined mixture model of principal component analyzers, and an expectation–maximisation algorithm for estimating all the model parameters is given.

INTRODUCTION

Principal component analysis (PCA) [8] is a popular technique for dimension reduction, with applications in data compression, image analysis, visualization and time series prediction for example. For a set of observed d-dimensional data points $\{\mathbf{t}_n\}$, $n \in \{1 \ldots N\}$, the *q principal axes* $\mathbf{w}_j$, $j \in \{1 \ldots q\}$, are those orthonormal axes onto which the retained *variance* under projection is maximal. It can be shown that the d-dimensional orthonormal vectors $\mathbf{w}_j$ are given by the q dominant eigenvectors (i.e. those with the largest associated eigenvalues) of the covariance matrix $\mathbf{S} = E[(\mathbf{t} - \boldsymbol{\mu})(\mathbf{t} - \boldsymbol{\mu})^{\mathrm{T}}]$ such that $\mathbf{S}\mathbf{w}_j = \lambda_j \mathbf{w}_j$. The vector $\mathbf{x}_n = \mathbf{W}^{\mathrm{T}}(\mathbf{t}_n - \boldsymbol{\mu})$, where $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_q)$, is thus a reduced q-dimensional representation of the observed vector $\mathbf{t}_n$. The optimal reconstruction of that observed vector, in terms of least–squared error, is given by $\hat{\mathbf{t}}_n = \mathbf{W}\mathbf{x}_n + \boldsymbol{\mu}$.

One limiting disadvantage of PCA as defined above is the absence of an associated probability, or generative, model. Deriving PCA from the perspective of density estimation offers a number of important advantages:

- There is a log–likelihood measure which permits comparison with other density–estimation techniques.
- Bayesian inference methods may be applied (e.g. for model comparison) by combining the likelihood with a prior and marginalising over parameters.
- When PCA is to be used to model the class–conditional densities in a classification problem, for example in [6], the posterior probabilities of class membership may be computed.
- The probability density function gives an indication of the *novelty* of a data vector.
- The single PCA model may be extended to a *mixture* of such models.

This final advantage is of considerable significance. Because PCA only defines a *linear* projection of the data, the scope of its application is necessarily somewhat limited. For realistic datasets, many components may be required to capture all the salient structure. This has naturally motivated various developments of *global nonlinear* principal component analysis in an effort to retain a greater proportion of the variance in fewer components — examples include 'principal curves' [5], multilayer auto-associative neural networks [11] and the Generative Topographic Mapping [2]. However, an alternative paradigm is to model nonlinear structure with a collection, or mixture, of *local linear* sub-models. This philosophy is an attractive one, motivating, for example, the "mixture of experts" technique in regression problems [9], and the approach has indeed been adopted in the context of PCA [10, 3, 7, 6].

A local model implementation of PCA implies the integration of two procedures: a partitioning of the data space into distinct regions, and the estimation of the principal axes within each such region. The question of how to combine these two elements can be problematic as the optimal partitioning depends on the current set of axes, while the optimal axes are determined by the current partition. The choice of method for interleaving these two stages is by necessity somewhat arbitrary. In [10], the expected reconstruction error is used as the distortion measure for a batch vector quantisation (VQ) step, followed by a local PCA in each cell defined by the VQ. In [6], a similar approach is followed, but interestingly, a *soft* version of the algorithm is also presented, where instead of any given data point being assigned exclusively to one region of the space (and thus one principal component analyzer), its 'responsibility' is shared amongst all of the analyzers. If r_{nj} is the responsibility of the jth analyzer for reconstructing data point $\mathbf{t}_n$, then $r_{nj} = \exp(-E_j^2/2\sigma^2)/(\sum_{j'} \exp(-E_{j'}^2/2\sigma^2))$, where E_j is the reconstruction cost of the jth analyzer. While a pseudo-EM (expectation-maximisation) algorithm is derived for fitting the model, it unfortunately requires the introduction of a noise parameter σ^2 whose value must be chosen arbitrarily.

A major benefit, then, of deriving a probabilistic model for PCA is that it is possible to formulate a model comprising a mixture of principal component analyzers in a principled manner. In this paper it is first shown how the principal subspace of a set of data vectors can be obtained through the maximisation of a likelihood function. We then extend this key result to a mixture modelling context and outline an efficient EM algorithm for estimating all of the model parameters. The necessary combination of the partitioning and principal axes estimation occurs automatically. Furthermore, it is also seen that the covariance responsibility–weighting in the soft version of Hinton *et al.* [6] is an implicit element in the algorithm, while the arbitrary parameter σ^2 is a well–defined ingredient of the model and is re–estimated within the EM procedure.

In the next section we describe the concept of latent variable models, and indicate how with a particular choice of noise model, the statistical technique of *factor analysis* can be adapted to extract the principal components of a dataset. An efficient EM algorithm for determining the parameters of a *mixture of principal component analyzers* is then given, and an example of its operation illustrated for a synthetic data set.

LATENT VARIABLE MODELS

In a latent variable model, it is desired to describe the set of d-dimensional observed data vectors $\{\mathbf{t}_n\}$ in terms of a set of q-dimensional latent variables $\{\mathbf{x}_n\}$ according to some model:

$$\mathbf{t} = \mathbf{y}(\mathbf{x}; \mathbf{w}) + \mathbf{u}, \tag{1}$$

where $\mathbf{y}(\cdot,\cdot)$ is a function of some random variable $\mathbf{x}$ with parameters $\mathbf{w}$, and $\mathbf{u}$ is an $\mathbf{x}$–independent noise process. Generally, $q < d$ such that the latent variables offer a more parsimonious description of the data. Such a model may also be termed 'generative', as data vectors $\mathbf{t}$ may be generated by sampling from the $\mathbf{x}$– and $\mathbf{u}$–distributions and applying (1).

The Standard Factor Analysis Model

In standard (normal) factor analysis the model is linear:

$$\mathbf{t} = \mathbf{W}\mathbf{x} + \mathbf{u} + \boldsymbol{\mu}, \tag{2}$$

where $\mathbf{x} \sim N(\mathbf{0}, \mathbf{I})$ are the q-dimensional *common factors* and $\mathbf{u} \sim N(\mathbf{0}, \boldsymbol{\Psi})$ (with $\boldsymbol{\Psi}$ diagonal) the *specific*, or *unique, factors*. The $(d \times q)$ parameter matrix $\mathbf{W}$ contains the *factor loadings*, and $\boldsymbol{\mu}$ is a constant which represents the mean of the data (in general the data will be assumed to have $\bar{\mathbf{t}} = \mathbf{0}$ and thus $\boldsymbol{\mu}$ is ignored). Given this formulation, $\mathbf{t}$ is also normally distributed $N(\boldsymbol{\mu}, \mathbf{C})$, where the model covariance is $\mathbf{C} = \boldsymbol{\Psi} + \mathbf{W}\mathbf{W}^{\mathrm{T}}$. The key motivation for this model is that the observed variables $\mathbf{t}$ are *conditionally independent* given the underlying factors, or *latent variables*, $\mathbf{x}$. Thus the reduced-dimensional distribution $\mathbf{x}$ is intended to model the dependencies between the observed variables while the specific factors model the independent noise. This is in contrast to PCA which treats the inter–variable dependencies and the independent noise identically.

Equation (2) implies a probability distribution over $\mathbf{t}$-space for a given $\mathbf{x}$:

$$p(\mathbf{t}|\mathbf{x}) = (2\pi)^{-d/2}|\boldsymbol{\Psi}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{t}-\mathbf{W}\mathbf{x})^{\mathrm{T}}\boldsymbol{\Psi}^{-1}(\mathbf{t}-\mathbf{W}\mathbf{x})\right\}, \tag{3}$$

and with a Gaussian prior over the latent variables defined by:

$$p(\mathbf{x}) = (2\pi)^{-q/2}\exp\left\{-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{x}\right\}, \tag{4}$$

then:

$$p(\mathbf{t}) = (2\pi)^{-d/2}|\mathbf{C}|^{-1/2}\exp\left\{-\frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t}\right\}, \tag{5}$$

since $p(\mathbf{t}) = \int p(\mathbf{t}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$.

Using Bayes' rule, the *posterior* distribution of the latent variables $\mathbf{x}$ given the observed $\mathbf{t}$ may be calculated:

$$p(\mathbf{x}|\mathbf{t}) = (2\pi)^{-q/2}|\mathbf{P}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{V}\mathbf{t})^{\mathrm{T}}\mathbf{P}^{-1}(\mathbf{x}-\mathbf{V}\mathbf{t})\right\}, \tag{6}$$

where the posterior mean and covariance are determined by:

$$\mathbf{V} = \mathbf{W}^{\mathrm{T}}\mathbf{C}^{-1}, \quad \text{and} \tag{7}$$

$$\mathbf{P} = \mathbf{I} - \mathbf{W}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{W}. \tag{8}$$

The log–likelihood of the data under the standard factor model is:

$$
\begin{aligned}
l &= \sum_{n=1}^{N} \ln\,[p(\mathbf{t}_n)], \\
&= -\frac{Nd}{2}\ln(2\pi) - \frac{N}{2}\ln|\mathbf{C}| - \\
&\quad \frac{N}{2}\mathrm{tr}\left[\mathbf{C}^{-1}\mathbf{S}\right],
\end{aligned} \tag{9}
$$

where $\mathbf{S}$ is the sample covariance matrix of the observed $\mathbf{t}$. An EM algorithm exists for maximising this measure [12].

An Isotropic Noise Model

The key to understanding the bridge from factor analysis to PCA is to realise that if the distribution of the specific factors is assumed *isotropic*, then $\sigma^2\mathbf{I}$ may be substituted for $\mathbf{\Psi}$, the covariance model for the observed data becomes $\mathbf{C} = \sigma^2\mathbf{I} + \mathbf{W}\mathbf{W}^{\mathrm{T}}$ and at the maximum–likelihood solution, the columns of $\mathbf{W}$ span the *principal subspace* of the data. Importantly, this enables us to derive PCA from the perspective of a density model.

We can prove this central result by considering the derivative of (9) with respect to $\mathbf{W}$:

$$
\frac{\partial l}{\partial \mathbf{W}} = N\mathbf{C}^{-1}(\mathbf{C} - \mathbf{S})\mathbf{C}^{-1}\mathbf{W}, \tag{10}
$$

which may be obtained from standard matrix results. By considering the singular value decomposition of the weight matrix

$$
\mathbf{W} = \mathbf{U}_q\mathbf{L}\mathbf{V}^{\mathrm{T}}, \tag{11}
$$

where $\mathbf{U}_q$ is a $(d \times q)$ matrix whose columns are orthonormal, $\mathbf{L}$ is diagonal, and $\mathbf{V}$ is an orthogonal matrix, it may be shown that the only non–zero solutions to (10) are given by:

$$
\mathbf{W} = \mathbf{U}_q(\mathbf{\Lambda}_q - \sigma^2\mathbf{I})^{\frac{1}{2}}\mathbf{V}^{\mathrm{T}}, \tag{12}
$$

where the columns of $\mathbf{U}_q$ are eigenvectors of $\mathbf{S}$, $\mathbf{\Lambda}_q$ is a diagonal matrix whose elements are the corresponding eigenvalues, and $\mathbf{V}$ is an arbitrary orthogonal matrix. Further details of this derivation are given in [14], where it is also shown that the global maximum of the likelihood occurs when $\mathbf{U}_q$ comprises the *principal* eigenvectors of $\mathbf{S}$ and furthermore all other combinations of eigenvectors are saddle–points on the likelihood surface.

Maximum–Likelihood PCA

As a result of these properties, by a simple extension of the EM formulation for parameter estimation in the standard factor model [12], we can obtain a principal component projection by maximising the likelihood function (9). Once the algorithm has converged, and $\mathbf{W}$ has been determined, then the principal subspace may be simply calculated by orthogonalisation of $\mathbf{W}$.

From (6) it can be seen that it is the application of Bayes' rule which effectively performs the analogue of the dimension–reducing projection of PCA, since the *posterior mean* projection of $\mathbf{t}_n$ is given by $\langle\mathbf{x}_n\rangle = \mathbf{W}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{t}_n$. Indeed, when $\sigma^2 \to 0$, $\mathbf{W}\langle\mathbf{x}_n\rangle \to \mathbf{W}(\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-1}\mathbf{W}^{\mathrm{T}}$ and thus becomes an orthogonal projection, so standard PCA is recovered. However, the density model then becomes singular and hence undefined. Conversely, when $\sigma^2 > 0$, the projection onto the manifold becomes skewed 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 $\mathbf{t}_n$. However, each data point may be still optimally reconstructed from the latent variable without further computation by taking this skewing into account [14].

In the next section, we develop an algorithm for fitting mixtures of principal component analyzers to data based on this probabilistic PCA model.

MIXTURES OF PRINCIPAL COMPONENT ANALYZERS

By adopting the isotropic noise model, it is now possible develop a maximum–likelihood formulation of a mixture of principal component analyzers. For such a model, the log–likelihood of observing the data set (i.i.d. assumed) is:

$$
\mathcal{L} = \sum_{n}^{N} \ln\,[p(\mathbf{t}_n)]\,, \tag{13}
$$

$$
= \sum_{n}^{N} \ln\left[\sum_{i=1}^{M} \pi_i p(\mathbf{t}_n|i)\right], \tag{14}
$$

where $p(\mathbf{t}_n|i)$ is a single PCA model and π_i is the mixing proportion, with $\pi_i \geq 0$ and $\sum \pi_i = 1$. Note that a separate mean vector $\boldsymbol{\mu}_i$ is now associated with each component mixture, along with the parameters $\mathbf{W}_i$ and σ_i^2. In addition, there are now M independent latent variables $\mathbf{x}_i$, one associated with each component. A related model has recently been exploited for data visualization [13], while a similar approach, based on the standard factor analysis diagonal ($\mathbf{\Psi}$) noise model, has been employed for handwritten digit recognition [6], although it does not implement PCA.

An EM Algorithm

To maximise the likelihood (13), we employ an iterative expectation–maximisation (EM) approach. We consider the latent variables $\{\mathbf{x}_{ni}\}$ to be 'missing' data and introduce further missing binary variables $\{z_{ni}\}$ to indicate which mixture component was responsible for generating data point $\mathbf{t}_n$. If the values of these missing variables were known, estimation of all the model parameters would be straightforward using standard techniques. However, for a given $\mathbf{t}_n$, we know neither the mixture component, nor the value of $\mathbf{x}_{ni}$, which was responsible for its generation. However, we do know the joint distribution of the observed and latent variables, $p(\mathbf{t}_n, \{\mathbf{x}_{ni}\}, \{z_{ni}\})$, and we can calculate the *expectation* of the corresponding *complete–data log–likelihood*. In the E–step of the EM algorithm this expectation, calculated with respect to the posterior distributions of $\mathbf{x}_{ni}$ and z_{ni} given the observed $\mathbf{t}_n$ and the current parameter values, is computed. In the M–step, new parameter values π_i^{new}, $\boldsymbol{\mu}_i^{new}$, $\mathbf{W}_i^{new}$ and $(\sigma_i^2)^{new}$ are determined which maximize the expected complete–data log–likelihood. This procedure is guaranteed to increase the likelihood we are interested in, (13), unless it is already at a local maximum [4, 1].

In fact, we adopt a two–stage procedure, corresponding to a *generalised* EM algorithm [4], in which each stage comprises both an E–step and an M–step. Firstly, we calculate the expectation with respect to the posterior distribution of z_{ni}, and update the parameters π_i^{new} and $\boldsymbol{\mu}_i^{new}$ only. In the second stage, we calculate the expectation with respect to the posterior distribution of $\mathbf{x}_{ni}$, and update the remaining parameters $\mathbf{W}_i^{new}$ and $(\sigma_i^2)^{new}$. This approach has the advantage that the expectation in the E–step can be computed using the updated values of $\boldsymbol{\mu}_i^{new}$, and results in both an increase in convergence speed and a simplification of the algorithm.

A prescription for the implementation of this EM algorithm is now given. Further mathematical details concerning its derivation are available in [14].

Stage 1

In the E–step, for each mixture component, the expectation with respect to z_{ni} results in the calculation of the *responsibility* R_{ni} of that component for generating data point $\mathbf{t}_n$:

$$\begin{aligned} R_{ni} &= p(i|\mathbf{t}_n), \\ &= \frac{p(\mathbf{t}_n|i)\pi_i}{p(\mathbf{t}_n)}. \end{aligned} \tag{15}$$

Then in the M–step:

$$\pi_i^{new} = \frac{1}{N}\sum_{n=1}^{N} R_{ni}, \tag{16}$$

$$\boldsymbol{\mu}_i^{new} = \frac{\sum_n^N R_{ni}\mathbf{t}_n}{\sum_n^N R_{ni}}. \tag{17}$$

Stage 2

For each mixture component, every data point has a posterior mean projection characterised by the statistics

$$\langle \mathbf{x}_{ni} \rangle = (\sigma_i^2\mathbf{I} + \mathbf{W}_i^{\mathrm{T}}\mathbf{W}_i)^{-1}\mathbf{W}_i^{\mathrm{T}}(\mathbf{t}_n - \boldsymbol{\mu}_i^{new}), \tag{18}$$

and

$$\langle \mathbf{x}_{ni}\mathbf{x}_{ni}^{\mathrm{T}} \rangle = \sigma_i^2(\sigma_i^2\mathbf{I} + \mathbf{W}_i^{\mathrm{T}}\mathbf{W}_i)^{-1} + \langle \mathbf{x}_{ni} \rangle\langle \mathbf{x}_{ni} \rangle^{\mathrm{T}}. \tag{19}$$

These statistics are calculated in the E–step of stage 2 exploiting the *updated* values of $\boldsymbol{\mu}_i^{new}$ computed in stage 1. This leads to the M–Step updates:

$$\begin{aligned} \mathbf{W}_i^{new} = &\left[\sum_n R_{ni}(\mathbf{t}_n - \boldsymbol{\mu}_i^{new})\langle \mathbf{x}_n \rangle^{\mathrm{T}}\right] \times \\ &\left[\sum_n R_{ni}\langle \mathbf{x}_n\mathbf{x}_n^{\mathrm{T}} \rangle\right]^{-1}, \end{aligned} \tag{20}$$

$$\begin{aligned} (\sigma_i^2)^{new} = &\frac{1}{\pi_i^{new} d} \times \\ &\sum_n R_{ni}\mathrm{tr}\Big[(\mathbf{t}_n - \boldsymbol{\mu}_i^{new})(\mathbf{t}_n - \boldsymbol{\mu}_i^{new})^{\mathrm{T}} - \\ &2(\mathbf{t}_n - \boldsymbol{\mu}_i^{new})\langle \mathbf{x}_n \rangle^{\mathrm{T}}(\mathbf{W}_i^{new})^{\mathrm{T}} + \\ &\mathbf{W}_i^{new}\langle \mathbf{x}_n\mathbf{x}_n^{\mathrm{T}} \rangle(\mathbf{W}_i^{new})^{\mathrm{T}}\Big]. \end{aligned} \tag{21}$$

Implementation

At first sight, determination of the necessary M–step updates for $\mathbf{W}_i$ and σ_i^2 from equations (18)-(21) appears computationally demanding. However, expanding for $\langle \mathbf{x}_n \rangle$ and $\langle \mathbf{x}_n\mathbf{x}_n^{\mathrm{T}} \rangle$ with further algebraic manipulation reveals that all the necessary updates can be expressed in terms of the previously calculated responsibilities R_{ni} and the quantities:

$$\mathbf{S}_i = \frac{1}{\pi_i^{new} N}\sum_n R_{ni}(\mathbf{t}_n - \boldsymbol{\mu}_i^{new})(\mathbf{t}_n - \boldsymbol{\mu}_i^{new})^{\mathrm{T}}, \tag{22}$$

$$\mathbf{M}_i = \sigma_i^2\mathbf{I} + \mathbf{W}_i^{\mathrm{T}}\mathbf{W}_i. \tag{23}$$

The matrix $\mathbf{S}_i$ in equation (22) may clearly be interpreted as a responsibility–weighted covariance matrix and is an intrinsic quantity computed within the formal M–step update equations.

The parameter updates for $\mathbf{W}_i$ and σ_i^2 then simplify to

$$\mathbf{W}_i^{new} = \mathbf{S}_i\mathbf{W}_i(\pi_i^{new}\sigma_i^2\mathbf{I} + \mathbf{M}_i^{-1}\mathbf{W}_i^{\mathrm{T}}\mathbf{S}_i\mathbf{W}_i)^{-1}, \tag{24}$$

$$(\sigma_i^2)^{new} = \frac{\mathrm{tr}\left[\mathbf{S}_i - \mathbf{S}_i\mathbf{W}_i\mathbf{M}_i^{-1}(\mathbf{W}_i^{new})^{\mathrm{T}}\right]}{\pi_i^{new}d}. \tag{25}$$

Note that the first instance of $\mathbf{W}_i$ in equation (25) above is the *old* value of the weights, while the second instance is that value calculated from equation (24).

All the necessary parameters for the mixture of PCA model may thus be determined by equations (15)–(17) followed by equations (22)–(25). Iteration of this sequence is guaranteed to find a local maximum of the likelihood (13).

At the maximum–likelihood solution, the model may be interpreted as a combination of local PCA models in that we can show, in a similar fashion to the single PCA model, that the stationary values of $\mathbf{W}_i^{new}$ from (24) occur when $\mathbf{W}_i^{new} = \mathbf{U}_q(\mathbf{\Lambda}_q - \sigma^2\mathbf{I})^{\frac{1}{2}}\mathbf{V}^{\mathrm{T}}$ [14]. In this case, the eigenvectors/values are those of the local responsibility–weighted covariance matrix $\mathbf{S}_i$. Note that the quantity $\mathbf{S}_i$ emerges as an inherent quantity in the EM algorithm and is not an *ad hoc* feature.

Given this result, $\mathbf{W}_i^{new}$, and from that, $(\sigma_i^2)^{new}$, may be calculated analytically for a given $\mathbf{S}_i$. It is therefore possible, for fixed responsibilities R_{ni}, to find the maximum of the complete–data likelihood, with respect to $\mathbf{W}$ and σ^2 in one step, rather than through the iterative re–estimation scheme. Such an approach would allow earlier algorithms to be viewed as an approximation to the a maximum–likelihood procedure.

However, such a procedure requires the determination of the eigenvectors of a $(p \times p)$ matrix at each step, whereas the EM scheme as given by (24) and (25) only necessitates the inversion of a $(q \times q)$ matrix. Both computations scale in the cube of the matrix dimension, so while the eigenvector decomposition of $\mathbf{S}_i$ may give better convergence step–for–step, each such update may require significantly more computations that for the EM updates, particularly if $q \ll p$. Furthermore, much of his computation would be wasted in the early stages of determining the model when the clustering of the data set (through determination of the $\boldsymbol{\mu}_i$) is likely to be sub–optimal.

EXPERIMENT

For a demonstration of the operation of the algorithm, we generated a synthetic dataset comprising 1000 data points sampled uniformly over the surface of a hemisphere, with additive Gaussian noise. Figure 1 shows this data.

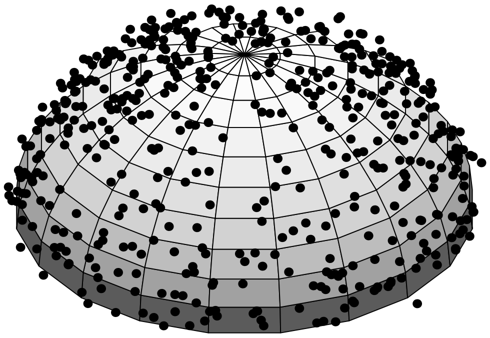

Figure 1: The synthetic data.

A mixture of 9 principal component analyzers was then fitted to the data using the EM algorithm. Because of the probabilistic formalism, there exists a generative model of the data and we emphasise this by generating another set of 1000 data points, but this time sampled from that generative model. These data points are illustrated in figure 2 and may be seen to occupy the region of the true data manifold. Figure 3 shows an alternative projection of the sampled data, with the vectors generated by one individual component of the model highlighted by darker plotting.

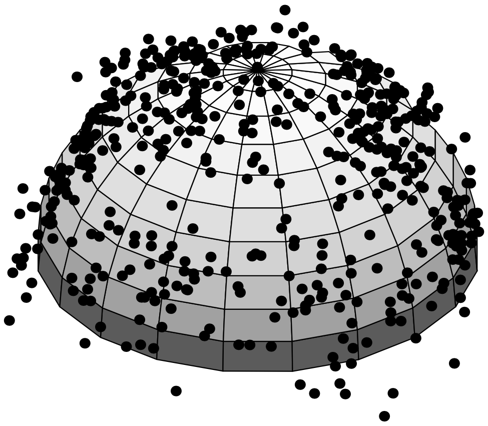

Figure 2: Data sampled from the trained generative model.

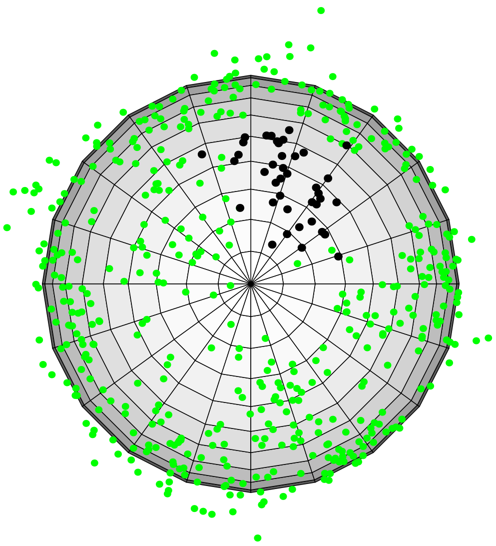

Figure 3: A single component highlighted.

CONCLUSIONS

Modelling complexity in data by a mixture of simple linear models is an attractive paradigm which offers both computational and algorithmic advantages along with increased ease of interpretability. Mixtures of principal component analyzers have thus been developed in several applications, but while algorithms in earlier work may be effective, they often require arbitrary implementational decisions or selection of parameters. We have therefore formulated a mixture of PCA models in terms of the solution to a maximum–likelihood problem, and derived an EM algorithm for estimating the parameters.

In addition to the concomitant advantages of defining a density function in the modelling process, there are also potential computational advantages, as no eigenvector decomposition of the covariance matrix is required. Furthermore, the partitioning and principal axis estimation steps are integrated automatically, and some of the features of earlier algorithms — specifically, soft clustering and responsibility-weighted covariance matrices — are inherent components of the EM update equations.

Acknowledgements: This work was supported by EPSRC contract GR/K51808: *Neural Networks for Visualization of High Dimensional Data*.

REFERENCES

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

[2] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: the Generative Topographic Mapping. accepted for publication in *Neural Computation*, 1997.

[3] C. Bregler and S. M. Omohundro. Nonlinear image interpolation using manifold learning. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 973–980. Cambridge, Mass: MIT Press, 1995.

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

[5] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84:502–516, 1989.

[6] G. E. Hinton, P. Dayan, and M. Revow. Modelling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks*, 8(1):65–74, 1997.

[7] G. E. Hinton, M. Revow, and P. Dayan. Recognizing handwritten digits using mixtures of linear models. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 1015–1022. Cambridge, Mass: MIT Press, 1995.

[8] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[9] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.

[10] N. Kambhatla and T. K. Leen. Fast nonlinear dimension reduction. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 152–159. San Francisco, CA: Morgan Kaufmann, 1994.

[11] M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

[12] D. B. Rubin and D. T. Thayer. EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69–76, 1982.

[13] M. E. Tipping and C. M. Bishop. A hierarchical latent variable model for data visualization. Technical report NCRG/96/028, Neural Computing Research Group, Aston University, Aston Street, Birmingham, B4 7ET, UK, 1996.

[14] M. E. Tipping and C. M. Bishop. Mixtures of principal component analyzers. Technical report NCRG/97/003, Neural Computing Research Group, Aston University, Aston Street, Birmingham, B4 7ET, UK, 1997.