HAL
open science

Locally linear generative topographic mapping

Jakob Verbeek, Nikos Vlassis, Ben Krose

To cite this version:

Jakob Verbeek, Nikos Vlassis, Ben Krose. Locally linear generative topographic mapping. Benelearn: Annual Machine Learning Conference of Belgium and the Netherlands, Dec 2002, Utrecht, Netherlands. pp.79–86. inria-00321501

HAL Id: inria-00321501

https://inria.hal.science/inria-00321501

Submitted on 16 Feb 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Locally Linear Generative Topographic Mapping

J.J. Verbeek and **N. Vlassis** and **B. Kröse**
Intelligent Autonomous Systems Group
Informatics Institute, University of Amsterdam
{*jverbeek,vlassis,krose*}*@science.uva.nl*

Abstract

We propose a method for non-linear data projection that combines Generative Topographic Mapping and Coordinated PCA. We extend the Generative Topographic Mapping by using more complex nodes in the network: each node provides a linear map between the data space and the latent space. The location of a node in the data space is given by a smooth non-linear function of its location in the latent space. Our model provides a piece-wise linear mapping between data and latent space, as opposed to the point-wise coupling of the Generative Topographic Mapping. We provide experimental results comparing this model with GTM.

1 Introduction

The Generative Topographic Mapping (GTM) (Bishop et al., 1998b) is a tool for non-linear data projection that has found many applications since its introduction. The idea is to fix a set of points in a latent space (which is used for visualization), these points are mapped through a Radial Basis Function Network (Ripley, 1996) into the data-space, where they are the means of the components of a Gaussian mixture. For projection, the data posterior probabilities on the mixture components are computed and the latent-space coordinates are averaged accordingly.

In this paper we extend the GTM by using more expressive covariance structure for the Gaussians in the mixture and, more importantly, by attaching to each mixture component a linear map between data and latent space.

The model may also be regarded as a restricted form of the Coordinated Principal Component Analysis (CPCA) model (Verbeek et al., 2002) (on its turn a restricted form of the coordinated factor analyzers model (Roweis et al., 2002)). CPCA extends probabilistic Principal Component Analysis (PCA) (Tipping and Bishop, 1999; Roweis, 1997) by integrating the mixture of probabilistic PCA's into one non-linear mapping between a data and a latent space. This is done by assigning a linear map from each PCA to a single, global, latent space. The objective function of CPCA sums data log-likelihood and a measure of how consistent the predictions of the latent coordinates are. With consistent we mean that PCA's that give high likelihood to a data point in the data space should give similar predictions on the (global) latent coordinate.

Here, we restrict the locations of the mixture components in the D-dimensional data space to be given by a smooth (non-linear) function of their location in the d-dimensional latent space ($d \ll D$). The relation between the proposed model and the CPCA is analogue to the relation between the GTM and Kohonen's Self-Organizing Map (Kohonen, 2001).

CPCA can be used for data visualization, however the model only aims at data log-likelihood maximization and at uni-modal distributions on the latent coordinate given a data point (consistency). What we would also like is uni-modal distributions in the other direction: the density on the data-space given a latent coordinate. By constraining the data-space centers of the Gaussian mixture components to be given by a smooth non-linear mapping of their latent space centers we safeguard CPCA against mixture components that are close in latent space but widely separated in the data space (which can cause multi-modal distributions from latent to data space).

The rest of the paper is organized as follows: in the following section we will discuss GTM and PCA and CPCA in some more detail.

Then, in Section 3 we present the new generative model and an EM-style learning algorithm is given in Section 4. Experimental results are presented and discussed in Section 5. We end the paper with a discussion and some conclusions in Section 6.

2 Generative Topographic Mapping, PCA and Coordinated PCA

Generative Topographic Mapping The Generative Topographic Mapping (GTM) (Bishop et al., 1998b) model is a tool for data visualization and data dimensionality reduction. Typically GTM is used for high dimensional numerical data. The data density is modeled by a mixture distribution that is parameterized in a way that enables low dimensional visualization of the data.

The generative model of GTM is a mixture of k spherical Gaussians, all having the same variance σ^2 in all directions. The components of the Gaussian mixture are also called nodes, and play a similar role as the nodes in Kohonen's Self-Organizing Map (Kohonen, 2001). With each mixture component s, a fixed latent coordinate $\boldsymbol{\kappa}_s$ is associated (typically the latent space is a two dimensional Euclidean space, and the $\boldsymbol{\kappa}_s$ are chosen on a rectangular grid). In the latent space a set of m fixed (non-linear) smooth basis-functions $\phi_i, \ldots, \phi_m$ are defined, the vector $\boldsymbol{\phi}(\boldsymbol{\kappa}_s)$ denotes the response of the m basis functions on input $\boldsymbol{\kappa}_s$. The mean of mixture component s is given by $\mathbf{W}\boldsymbol{\phi}(\boldsymbol{\kappa}_\mathbf{s})$, where $\mathbf{W}$ is a $D \times m$ matrix. Hence, due to the smoothness of the basis-functions and because the multiplication with $\mathbf{W}$ is just a linear map, components with close latent coordinates will have close means in the data space.

Suitable parameters σ and $\mathbf{W}$ of GTM are typically found by employing a simple Expectation-Maximization (EM) algorithm. Like most EM algorithms, the algorithm can get stuck at locally optimal solutions, there is no guarantee to find globally optimal parameters

In order to visualize the high dimensional data in the latent space, for each data point the posterior distribution on the mixture components is computed. The latent coordinates of the nodes are then weighted according to the posterior distribution to give the latent coordinate for the data point.

Principal Component Analysis Principal Component Analysis (PCA) (Jolliffe, 1986) is a widely known and used method for data visualization and dimensionality reduction. Assuming the data has zero mean, the data vectors $\{\mathbf{x}_n\}$ are modeled as $\mathbf{x}_n = \boldsymbol{\Lambda}\mathbf{g}_n + \epsilon_n$. Here, $\mathbf{x}_n$ is a D dimensional data vector, $\mathbf{g}_n$ the corresponding latent coordinate, $\boldsymbol{\Lambda}$ a $D \times d$ dimensional matrix and ϵ_n a residual vector. PCA uses the linear mapping $\boldsymbol{\Lambda}$ that minimizes sum of the squared norms of the residual vectors to map the data into a d dimensional representation.

The PCA linear map (matrix) $\boldsymbol{\Lambda}$ can also be found as the limiting case of $\sigma \to 0$ of fitting a Gaussian density to the data where the covariance matrix is constrained to be of the form $\sigma^2\mathbf{I} + \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top$ The latter model is known as Probabilistic PCA (Roweis, 1997).

Coordinated PCA Several Probabilistic PCA's can be combined in a mixture, by taking a weighted sum of the component densities (Tipping and Bishop, 1999). In the mixture case a set local PCA's is fitted to the data. Data can now be mapped in either one of the PCA models, however coordinates of the different PCA spaces cannot be related to each other. The coordinated PCA model (Roweis et al., 2002; Verbeek et al., 2002) resolves this problem. Each PCA space is mapped linearly into a global low dimensional space. The global low dimensional space provides a single low dimensional coordinate system for the data. The question is which linear maps between the PCA's and the global latent space we would like.

It turns out that the data log-likelihood is invariant for how the different PCA spaces are coordinated, i.e. mapped into the global space. Therefore in (Roweis et al., 2002; Verbeek et al., 2002) a penalty term is added to data log-likelihood to find a coordinated mixture of PCA's. For every data point $\mathbf{x}_n$ we can compute the posterior probability on every PCA s. Furthermore, a prediction $p(\mathbf{g}_n|s, \mathbf{x}_n)$ on the global latent coordinate can be made using either PCA. The penalty term measures for every data point the ambiguity of the prediction on its global latent coordinate as given by $p(\mathbf{g}_n|\mathbf{x}_n) = \sum_s p(s|\mathbf{x}_n)p(\mathbf{g}_n|s, \mathbf{x}_n)$. The measure of ambiguity is the smallest Kullback-Leibler divergence between $p(\mathbf{g}_n|\mathbf{x}_n)$ and any Gaussian density.

It turns out that adding this penalty term

is exactly the same as using a variational EM algorithm (Neal and Hinton, 1998). Hence, it is relatively straightforward to derive EM algorithms to optimize the parameters of the model for this objective function.

In the next section we describe a generative model that combines the ideas of GTM and Coordinated PCA.

3 The Generative Model of Locally Linear GTM

The generative model is best understood by starting in the latent space. First one of the k 'units' is chosen uniformly random, let's call it s. Then, a latent coordinate is drawn from $p(\mathbf{g} \mid s)$. Finally, an observation $\mathbf{x}$ is drawn from $p(\mathbf{x} \mid \mathbf{g}, s)$. Using $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to denote a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the distributions, collectively parameterized by $\boldsymbol{\theta} = \{\sigma, \rho, \alpha, \mathbf{W}, \boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_k\}$, are:

$$\begin{aligned} p(s) &= \frac{1}{k} \\ p(\mathbf{g} \mid s) &= \mathcal{N}(\boldsymbol{\kappa}_s, \alpha^2 \rho \sigma^2) \\ p(\mathbf{x} \mid \mathbf{g}, s) &= \mathcal{N}(\mathbf{W}\boldsymbol{\phi}(\boldsymbol{\kappa}_s) + \boldsymbol{\Lambda}_s(\mathbf{g} - \boldsymbol{\kappa}_s)/\alpha, \sigma^2 \mathbf{I}) \end{aligned}$$

The marginal distribution on $\mathbf{x}$ given s reads:

$$p(\mathbf{x} \mid s) = \mathcal{N}(\mathbf{W}\boldsymbol{\phi}(\boldsymbol{\kappa}_s), \sigma^2(\mathbf{I} + \rho \boldsymbol{\Lambda}_s \boldsymbol{\Lambda}_s^\top))$$

Each mixture component $p(\mathbf{x} \mid s)$ is a Gaussian with variance σ^2 in all directions except those directions given by the columns of $\boldsymbol{\Lambda}_s$ where the variance is ρ times larger. The form of the covariance matrix is that of probabilistic PCA, however here the vectors spanning the PCA subspace (given by the columns of $\boldsymbol{\Lambda}$) all have the same norm.

The distribution on latent coordinates is a mixture of isotropic Gaussians (spherical covariance matrix) with equal mixing weights. The columns of the matrix $\boldsymbol{\Lambda}_s$ span the PCA subspace of mixture component s, and provide the mapping for component s between the latent space and the data space. The parameter α is a magnification factor between latent and data space.

The generative model is almost the same as in (Verbeek et al., 2002), the only difference is that we constrain the data-space mean of the mixture component s to $\mathbf{W}\boldsymbol{\phi}(\boldsymbol{\kappa}_s)$ and keep the $\{\boldsymbol{\kappa}_s\}$ fixed. The mapping is a linear sum, parameterized by the $D \times m$ matrix $\mathbf{W}$, of m non-linear basis functions $\phi_1, \ldots, \phi_m$ of d inputs. The output of the function $\phi_i(\boldsymbol{\kappa})$ is the i-th element of the vector $\boldsymbol{\phi}(\boldsymbol{\kappa})$. In fact, we also add one constant basis function and d linear functions which just output one of the components of the input vector. These basis-functions can be used to capture linear trends in the data. In our model we consider the basis functions $\{\phi_i\}$ to be fixed as well as the $\{\boldsymbol{\kappa}_s\}$, all other parameters are fitted to a given data set.

To compare, the generative model of GTM, which is a special case with $\rho = 0$ of the model used here, looks like:

$$\begin{aligned} p(s) &= \frac{1}{k} \\ p(\mathbf{g}|s) &= \delta(\boldsymbol{\kappa}_s) \\ p(\mathbf{x}|s, \mathbf{g}) &= \mathcal{N}(\mathbf{W}\boldsymbol{\phi}(\boldsymbol{\kappa}_s), \sigma^2 \mathbf{I}), \end{aligned}$$

where $\delta(\boldsymbol{\kappa}_s)$ denotes the distribution that puts all mass at $\boldsymbol{\kappa}_s$. This is similar to the model used in this paper, but (1) GTM uses a mixture of spherical Gaussians for the data space model and (2) GTM uses a mixture of delta peaks to model the density in the latent space.

Note that in our model, given a specific mixture component, the expected value of $\mathbf{x}$ depends linearly on $\mathbf{g}$ and vice versa, see Section 4.3. Whereas using GTM, given a mixture component s, there is no modeling of uncertainty in the latent coordinate $\mathbf{g}$ corresponding to a data point $\mathbf{x}$, it is simply assumed that $\mathbf{g} = \boldsymbol{\kappa}_s$.

4 Learning

The learning algorithm tries to achieve two goals. On the one hand we want a good fit on the observed data, i.e. high data log-likelihood. On the other hand, we want the mappings from data to latent space to give 'consistent' predictions, i.e. if a data point is well modeled by two mixture components in the data space, then these should give similar predictions on the corresponding latent coordinate of the data point. In the following subsections we first discuss the objective function, then we describe how we initialize all the model parameters, and finally we discuss the learning algorithm.

4.1 The Objective Function

As shown by Roweis et. al. (Roweis et al., 2002) the double objective can be pursued with a variational Expectation-Maximization (EM) algorithm. The free-energy interpretation of EM (Neal and Hinton, 1998) clarifies how we can use EM-like algorithms to optimize penalized log-likelihood objectives. The penalty term is the Kullback-Leibler (KL) divergence between the posterior on the hidden variables and a specific family of distributions, in our case uni-modal Gaussian distributions. The objective function sums data log-likelihood and the KL divergence which is a measure of how uni-modal the predictions on the latent coordinate are.

The objective $\mathbf{\Phi}$ can be decomposed in two ways, given in equations (1) and (2) below, which can be associated respectively with the E and M step of EM. The first is the log-likelihood minus the KL-divergence, which is convenient for the E-step. The second decomposition isolates an entropy term independent of $\boldsymbol{\theta}$, which is convenient in the M-step.

$$\begin{aligned}\mathbf{\Phi} &= -\frac{\lambda}{2} \parallel \mathbf{W} \parallel^2 + \sum_n \log p(\mathbf{x}_n; \boldsymbol{\theta}) \\ &\quad - \sum_n \mathcal{D}_{KL}(Q_n(\mathbf{g}, s) \parallel p(\mathbf{g}, s \mid \mathbf{x}_n; \boldsymbol{\theta})) \quad (1) \\ &= -\frac{\lambda}{2} \parallel \mathbf{W} \parallel^2 + \sum_n \mathrm{E}_{Q_n} \log p(\mathbf{x}_n, \mathbf{g}, s; \boldsymbol{\theta}) \\ &\quad + \sum_n \mathcal{H}(Q_n(\mathbf{g}, s)), \quad (2)\end{aligned}$$

where $\mathcal{D}_{KL}$ and $\mathcal{H}$ are used respectively to denote Kullback-Leibler divergence and entropy (Cover and Thomas, 1991). The first term in both decompositions implements a Gaussian prior distribution over the elements of the matrix $\mathbf{W}$, with inverse variance λ on each element. It makes sense to put a uniform prior on on the weights for the linear and constant basis-functions. However, to keep notation simple, we assume equal prior on all weights here.

The distributions $Q_n(\mathbf{g}, s)$ over mixture components s and latent coordinates $\mathbf{g}$ are restricted to be independent over s and $\mathbf{g}$, and of the form:

$$Q_n(\mathbf{g}, s) = q_{ns} Q_n(\mathbf{g}) \quad \text{and} \quad Q_n(\mathbf{g}) = \mathcal{N}(\mathbf{g}_n, \mathbf{\Sigma}_n)$$

It turns out that $\mathbf{\Phi}$ is maximized w.r.t. $\mathbf{\Sigma}_n$ if $\mathbf{\Sigma}_n = v^{-1}\mathbf{I}$, where $v = (\rho + 1)/(\alpha^2 \rho \sigma^2)$ is the inverse variance of $p(\mathbf{g} \mid \mathbf{x}, s)$. Hence, in the following we assume this equality and do not use $\mathbf{\Sigma}_n$ as a free parameter anymore. The $\{q_{ns}\}$ play the role of the 'responsibilities' in normal EM. Finally, $\mathbf{g}_n$ is the expected latent coordinate for data point $\mathbf{x}_n$, if we approximate the true distribution on the latent coordinate with a uni-modal Gaussian. Hence, we might use the $\mathbf{g}_n$ for visualization purposes.

4.2 Initialization

We initialize the model using Principal Component Analysis (PCA) (Jolliffe, 1986). Suppose, without loss of generality, that the data has zero mean. We use a projection of the data to the two dimensional PCA subspace to initialize the locations $\mathbf{g}_n$ of the distributions $Q_n(\mathbf{g})$. The columns of the loading matrices $\{\mathbf{\Lambda}_s\}$ are initialized as the principal eigenvectors of the data covariance matrix. The variance σ^2 is initialized as the mean variance in directions outside the PCA subspace. The $\{\boldsymbol{\kappa}_s\}$ are initialized on a square grid such that they have zero mean and the trace of their covariance matrix matches that of the covariance of the $\{\mathbf{g}_n\}$. The $\mathbf{W}$ is then taken as to map the $\{\boldsymbol{\kappa}_s\}$ onto $\{\mathbf{\Lambda}_s \boldsymbol{\kappa}_s\}$. Then, we assign the data point n to the component with $\boldsymbol{\kappa}_s$ closest to $\mathbf{g}_n$. Using the non-empty clusters, we compute the ρ and α. To initialize ρ, we use the mean value (over all non-empty clusters) of the total variance in the cluster in the data space minus $D\sigma^2$ and divide the sum over $d\sigma^2$. For α^2 we use the average of the mean variance in the cluster in the latent space and divide it over $\rho\sigma^2$. Finally, we can compute optimal q_{ns} and start the learning algorithm with a M-step of the EM-algorithm described in the next section.

The inverse variance λ of the prior on the elements of $\mathbf{W}$ can be made larger to get smoother functions. In our experiments we used $\lambda = 1$.

4.3 Updating the Parameters

Using the notation:

$$\begin{aligned}\mathcal{E}_{ns} = \frac{1}{2\sigma^2}\mathbf{x}_{ns}^\top \mathbf{x}_{ns} + \frac{v}{2}\mathbf{g}_{ns}^\top \mathbf{g}_{ns} - \sigma^{-2}\alpha^{-1}\mathbf{x}_{ns}^\top \mathbf{\Lambda}_s \mathbf{g}_{ns} \\ + D \log \sigma + \frac{d}{2}\log(\rho + 1),\end{aligned}$$

$$\mathbf{x}_{ns} = \mathbf{x}_n - \mathbf{W}\boldsymbol{\phi}(\boldsymbol{\kappa}_s), \quad \text{and} \quad \mathbf{g}_{ns} = \mathbf{g}_n - \boldsymbol{\kappa}_s.$$

The objective (2) can be written as:

$$\mathbf{\Phi} = -\frac{\lambda}{2} \parallel \mathbf{W} \parallel^2 - \sum_{ns} q_{ns} \Big[\log q_{ns} + \mathcal{E}_{ns} \Big] + c,$$

where c is some constant in the parameters. All the update equations below can be found by setting the derivative of the objective to zero for the different parameters. First, we give the E-step update equations:

$$q_{ns} = \frac{e^{-\mathcal{E}_{ns}}}{\sum_{s'} e^{-\mathcal{E}_{ns'}}}, \qquad \mathbf{g}_n = \sum_s q_{ns} \langle \mathbf{g}_n \rangle_s,$$

where for all N data points the expected latent coordinate given component s is:

$$\langle \mathbf{g}_n \rangle_s = \mathrm{E}_{p(\mathbf{g}|\mathbf{x},s)}[\mathbf{g}] = \boldsymbol{\kappa}_s + \frac{\alpha\rho}{\rho+1} \mathbf{\Lambda}_s^\top \mathbf{x}_{ns}.$$

Next, we give the M-step update equations. For $\mathbf{\Lambda}_s$ we have to minimize:

$$\sum_n q_{ns} \mathbf{x}_{ns}^\top \mathbf{\Lambda}_s \mathbf{g}_{ns}.$$

This problem is known as the 'weighted Procrustes rotation' (Cox and Cox, 1994). Let

$$\mathbf{C} = [\sqrt{q_{1s}}\mathbf{x}_{1s} \cdots \sqrt{q_{Ns}}\mathbf{x}_{Ns}][\sqrt{q_{1s}}\mathbf{g}_{1s} \cdots \sqrt{q_{Ns}}\mathbf{g}_{Ns}]^\top$$

with SVD: $\mathbf{C} = \mathbf{U}\mathbf{L}\mathbf{\Gamma}^\top$, (the $\mathbf{g}_{ns}$ have been padded with zeros to form D-dimensional vectors). The optimal $\mathbf{\Lambda}_s$ is given by the first d columns of $\mathbf{U}\mathbf{\Gamma}^\top$. For $\mathbf{W}$ we find:

$$\mathbf{W} = \Big[\sum_{ns} q_{ns} (\mathbf{x}_n - \alpha_s^{-1} \mathbf{\Lambda}_s \mathbf{g}_{ns}) \boldsymbol{\phi}(\boldsymbol{\kappa}_s)^\top \Big] \times \Big[\sum_{ns} q_{ns} \boldsymbol{\phi}(\boldsymbol{\kappa}_s) \boldsymbol{\phi}(\boldsymbol{\kappa}_s)^\top + \lambda\sigma^2 \mathbf{I} \Big]^{-1}.$$

For the updates of the three variance parameters we use the following notation:

$$A = \sum_{ns} q_{ns} \mathbf{x}_{ns}^\top \mathbf{\Lambda}_s \mathbf{g}_{ns}, \quad B = \sum_{ns} q_{ns} \mathbf{x}_{ns}^\top \mathbf{x}_{ns}, \quad C = \sum_{ns} q_{ns} \mathbf{g}_{ns}^\top \mathbf{g}_{ns}.$$

Simultaneously setting the derivatives of $\mathbf{\Phi}$ w.r.t. α, ρ and σ to zero gives:

$$\rho + 1 = \frac{(D-d)A^2}{d(BC - A^2)}, \quad \alpha = \frac{C(\rho+1)}{A\rho}, \quad \sigma^2 = \frac{B - \alpha^{-1}A}{ND}$$

5 Experimental Illustration

In this section we provide experimental results comparing the presented model with GTM. The first two experiments use artificially generated data. In the last experiment we use chemical chromatogram data from the Unilever research department.

Artificial data The first data set consisted of points in $\mathbb{R}^2$ drawn along the sine function. We used GTM (20 units, 5 basis-functions) and our new model (5 units, 3 basis-functions) to map the data to a one dimensional latent space. The fitted models and the latent space representations are given in Figure 1. We see that for GTM the distribution of the latent coordinates clusters around the mixture component locations. For our model, while using 4 times fewer mixture components, the latent coordinates do not show this clustering due to the piece-wise linear maps.

Next, we generated a data set in $\mathbb{R}^3$ by first drawing uniformly random from the unit square in $\mathbb{R}^2$, then to every data point $\mathbf{x}_n$ we then added a third coordinate $\mathbf{x}_n^{(3)}$ with:

$$\mathbf{x}_n^{(3)} = (\mathbf{x}_n^{(1)} - 1/2)^2 + 2(\mathbf{x}_n^{(2)} - 1/2)^3 + \epsilon,$$

where ϵ is a random variable with distribution $\mathcal{N}(\epsilon; 0, 100^{-1})$. In Figure 2 we show the projections found by our method and GTM. Again we see that GTM concentrates the latent coordinates around the locations of the units, the projection 'clumps' around the units. Whereas our method gives a projection that matches the original distribution on the unit square better.

Chromatogram data We demonstrate the performance of the proposed method and GTM on a problem from high-throughput screening of Maillard reactions, which involves the heating of sugars and amino acids, giving rise to complex mixtures of volatile flavor compounds. The reaction product mixtures are characterized through fast gas chromatography. At Unilever R&D labs, it was found that a single GTM plot of the chromatographic data, based on two latent variables, is equally informative as the screening of many score plots from various PCA combinations. The data was preprocessed by mapping it with PCA onto a 20 dimensional linear subspace of the original 2800 dimensional space.

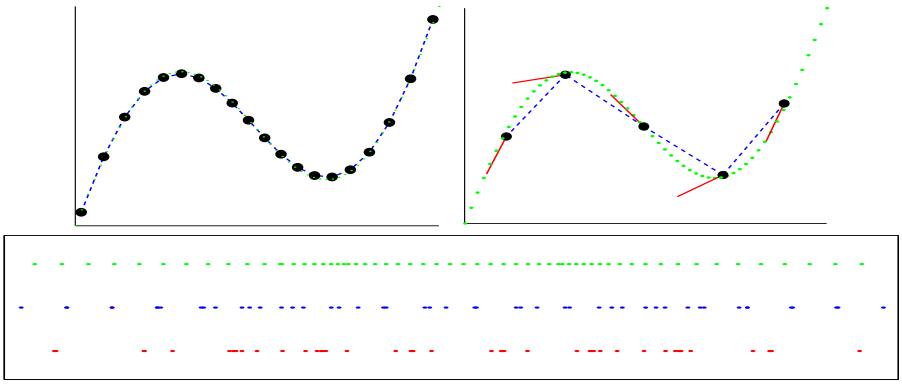

Figure 1: Top left panel: GTM fit (disks give means of mixture components, dashed line the ordering of the components in latent space) on data (dots). Top right panel: our new model (the sticks depict vectors $\sqrt{\rho}\sigma\mathbf{\Lambda}_s$ used for the linear maps). Bottom panel: (top) latent coordinates for our method, and GTM (middle 20 nodes, bottom 10 nodes). Both sets of latent points have the same variance.

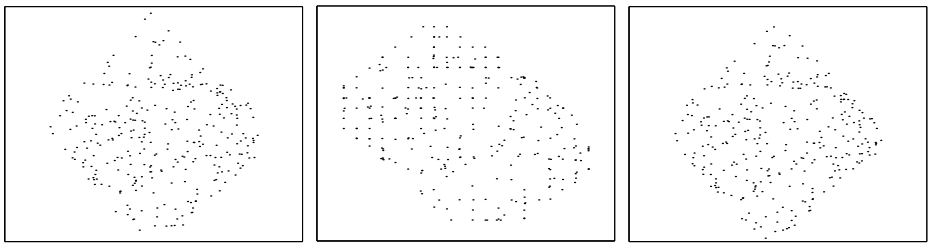

Figure 2: Projections of (from left to right) the proposed model, GTM and original 2D coordinates (which are rotated).

We call an ingredient separable if all the measurements on compounds which contain, or do not contain, the ingredient form a cluster in the projected data. In our experiment we found that most ingredients which were separable using GTM were also separable with our new projection method. In the experiments GTM used 400 nodes and 16 basis-functions, our model used only 36 units and 16 basis functions.

6 Discussion and Conclusions

Discussion Constraining the means of the mixture components to the manifold spanned by a smooth mapping ensures that components that are close in latent space will also be close in the data space. This makes the constrained version of Coordinated PCA more suitable for visualization purposes.

Comparing the presented model with GTM, we observe that we extended the point-wise coupling between data and latent space to a locally linear coupling. This removes the effect of 'clumpy' projections, as discussed in the previous section.

Several variations on the presented model are possible. For example, we might model the variances per mixture component and make the $\{\boldsymbol{\kappa}_s\}$ free parameters. However, here we wanted to keep the model close to GTM. As for the last option, the optimization w.r.t. the $\{\boldsymbol{\kappa}_s\}$ is a somewhat more involved due to the non-linear mappings $\boldsymbol{\phi}$. However, using local linear approximations of the non-linear functions we could do the optimization.

Note that the component covariance structure is not aligned with the non-linear function. However, in practice we *do* expect the local subspaces to be more or less aligned with the manifold due to the coordination (see Figure 1). To speed-up the algorithm, we might set the $\{\mathbf{\Lambda}_s\}$

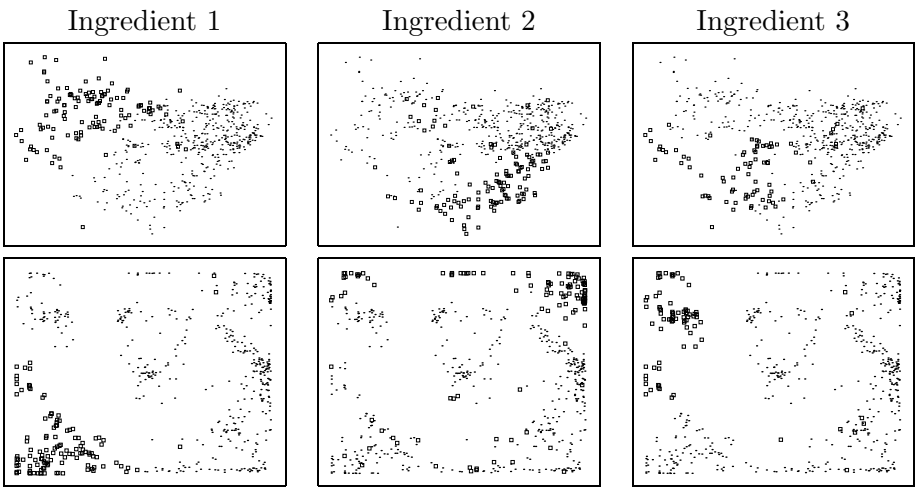

Figure 3: Projections of all the data for the proposed model (top) and GTM (bottom). From left to right we highlight compounds with one of three ingredient with boxes.

as manifold aligned and check whether this increases the objective function. We only need re-compute the $\{\mathbf{\Lambda}_s\}$ with SVD if taking them manifold aligned does not increase the objective function.

In (Bishop et al., 1998a) an extension of GTM is discussed where the the covariance matrix of $p(\mathbf{x}|s)$ is taken to be of the form $\sigma^2\mathbf{I} + \eta\mathbf{D}\mathbf{D}^\top$, where the d columns of the $D \times d$ matrix $\mathbf{D}$ contain the partial derivatives of the RBF network.[1] This generative model has a similar form of the covariance matrix in the data space as we use, however in the latent space it uses the same discrete distribution as GTM. Therefore, the extension described in (Bishop et al., 1998a), just like normal GTM, doesn't have the linear dependence of the expected value of $\mathbf{g}$ on $\mathbf{x}$ given s.

Conclusions We proposed a generative model and corresponding learning algorithm for visualization and projection of high-dimensional data. The model integrates the GTM and Coordinated PCA.

Experimental results show that the model does not suffer from the 'clumpy' projections of GTM. Furthermore, in the latent-space we have a true density function where GTM only provides a discrete 'spike' distribution on the latent-space. Also, the proposed model provides a density $p(\mathbf{x} \mid \mathbf{g})$, whereas for GTM this distribution is not clearly defined.

Finally, the proposed model provides a measure of uncertainty on the found latent coordinates summarized in the distributions $Q_n(\mathbf{g})$.

[1]The M step of the proposed EM algorithm is not exact, due to the dependence of $\mathbf{D}$ on $\mathbf{W}$.

Acknowledgments

We would like to thank Dick de Ridder and Sam Roweis for fruitful discussions. We would also like to thank Claire Boucon and Sijmen de Jong (Unilever R&D Vlaardingen, The Netherlands) for providing the Maillard reaction data and experimental results. This research is supported by the Technology Foundation STW (project nr. AIF4997) applied science division of NWO and the technology program of the Dutch Ministry of Economic Affairs.

References

C. M. Bishop, M. Svensén, and C. K. I. Williams. 1998a. Developments of the generative topographic mapping. *Neurocomputing*, 21:203–224.

C. M. Bishop, M. Svensén, and C. K. I Williams. 1998b. GTM: The generative topographic mapping. *Neural Computation*, 10:215–234.

T. Cover and J. Thomas. 1991. *Elements of Information Theory*. Wiley.

T.F. Cox and M.A.A. Cox. 1994. *Multidimensional Scaling*. Number 59 in Monographs on statistics and applied probability. Chapman & Hall.

I.T. Jolliffe. 1986. *Principal Component Analysis*. Springer-Verlag.

T. Kohonen. 2001. *Self-Organizing Maps*. Springer-Verlag.

R. M. Neal and G. E. Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M.I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer.

B.D. Ripley. 1996. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, U.K.

S.T. Roweis, L.K. Saul, and G.E. Hinton. 2002. Global coordination of local linear models. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, USA. MIT Press.

S.T. Roweis. 1997. EM Algorithms for PCA and SPCA. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 626–632. MIT Press.

M.E. Tipping and C.M. Bishop. 1999. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482.

J.J. Verbeek, N. Vlassis, and B. Kröse. 2002. Coordinating Principal Component Analyzers. In J.R. Dorronsoro, editor, *Proceedings of Int. Conf. on Artificial Neural Networks*, pages 914–919, Madrid, Spain. Springer.