Semi-Supervised Learning of Hierarchical Latent Trait Models for Data Visualisation

Yi Sun   Peter Tiňo

Neural Computing Research Group

Aston University, Birmingham, B4 7ET, UK

suny,tinop@aston.ac.uk

Ata Kabán

Department of Information Systems

Eötvös Loránd University,

Budapest, Hungary

ata@ullman.inf.elte.hu

Ian Nabney

Neural Computing Research Group

Aston University, Birmingham, B4 7ET, UK

nabneyit@aston.ac.uk

Keywords: Hierarchical model, Latent trait model, Magnification factors, Data visualisation, Document mining

Corresponding author: Peter Tiňo

email: tinop@aston.ac.uk, Phone: +44 121 359 333 4631, Fax: +44 121 333 4586

Abstract

Recently, we have developed an interactive hierarchical Generative Topographic Mapping (HGTM) for visualisation of large high-dimensional real-valued data sets. In this paper, we build a more general visualisation system by extending HGTM in 3 directions: **(1)** We generalize HGTM to noise models from the exponential family of distributions. The basic building block is the Latent Trait Model (LTM). This enables us to visualise data of inherently discrete nature, e.g. collections of documents. **(2)** We give the user a choice of initializing the child plots of the current plot in either *interactive*, or *automatic* mode. In the interactive mode the user interactively selects "regions of interest" as in HGTM, whereas in the automatic mode an unsupervised minimum message length (MML)-driven construction of a mixture of LTMs is employed. The unsupervised construction is particularly useful when high-level plots are covered with dense clusters of highly overlapping data projections, making it difficult to use the interactive mode. Such a situation often arises when visualizing large data sets. **(3)** We derive general formulas for magnification factors in latent trait models. Magnification factors are a useful tool to improve our understanding of the visualisation plots, since they can highlight the boundaries between data clusters.

We illustrate our approach on a toy example and apply our system to three more complex real data sets.

1 Introduction

Topographic visualisation of multi-dimensional data has been an important method of data analysis and data mining [4, 12]. In a complex setting, however, a single two-dimensional projection of high-dimensional data may not be sufficient to capture all of the interesting aspects of the data. Therefore, hierarchical extensions of visualisation methods [6, 14] have been developed. Recently, we have developed a principled approach to interactive construction of non-linear visualisation hierarchies [16], the basic building block of which is the Generative Topographic Mapping (GTM) [4]. GTM is a probabilistic reformulation of the self-organizing map (SOM) [11] in the form of a non-linear latent variable model with a spherical Gaussian noise model.

Here we extend the hierarchical GTM (HGTM) visualisation system to noise models from the exponential family of distributions by employing the more general Latent Trait Model (LTM) developed in [10] as a starting point. This enables us to visualise data of inherently discrete nature, e.g. collections of documents.

In addition, we provide the user with a choice of initializing the child plots of the current plot in either *interactive*, or *automatic* manner within the same principled probabilistic framework. In the interactive mode, employed also in [16], the sub-plots ('child plots') must be initialised interactively by the user; they decide which subsets of the data are interesting enough to be visualized in a greater detail in sub-plots [16]. The automatic mode is a new feature incorporated now into the system, which allows us to determine both the number and the position of children LTMs in an *unsupervised* manner using the minimum message length (MML) methodology. This is particularly valuable when dealing with large quantities of data that make visualisation plots at higher levels complex and difficult to deal with in an interactive manner.

An intuitively simple but flawed approach would be to use a data partitioning technique (e.g. [15]) for segmenting the data set, followed by constructing visualisation plots in the individual compartments. Clearly, in this case there would be no direct connection between the criterion for choosing the quantization regions and that of making the local low-dimensional projections. By employing LTM, however, such a connection can be established in a principled manner. This is achieved by exploiting this model as a generative probabilistic model, which enables us to use a principled minimum message length (MML)-based learning of mixture models with an embedded model selection criterion [8]. Hence, given a parent LTM, the number and position of its children is based on the modelling properties of the children themselves – without any ad-hoc criteria which would be exterior to the model.

Previous experience has indicated that magnification factors may provide a potentially valuable additional information to our understanding of the visualisation plots, since they can highlight the boundaries between data clusters. In [5], formulas for magnification factors were only derived for the GTM. In this paper, we derive formulas for magnification factors in full generality for latent trait models.

In the next section we briefly review the latent trait model. In Section 3, the hierarchical extension of this model is provided. Section 4 presents the model selection criterion based on minimum message length that we apply to mixtures of LTMs. Section 5 presents and discusses experimental results. We derive a general formula for magnification factors in LTMs in Section 6. Finally, Section 7 summarizes the key contributions of the paper.

2 The Latent Trait Model (LTM)

Latent trait models [10] are generative models which provide powerful and principled tools of data analysis and visualisation. Being a generalisation of the Generative Topographic Mapping (GTM) [4], the latent trait model family [10] offers the formal treatment which includes the definition of the appropriate probability models for the cases of discrete observations.

Consider an L-dimensional latent space $\mathcal{H}$, which, for visualisation purposes is typically a bounded 2-D Euclidean domain, e.g. $[-1,1] \times [-1,1]$. The aim is to represent multi-dimensional data vectors $\{\mathbf{t}_n\}_{n=1,\ldots,N}$ using the latent space so that 'important' structural characteristics are revealed. A non-linear relation is allowed between the latent space and the data space $\mathcal{D} = \Re^D$. The latent plane becomes a (non-linear) 2-D manifold in the high dimensional data space.

For tractability, in practice the latent space is discretised by introducing a regular array (grid) of K latent points $\boldsymbol{x}_k \in \mathcal{H}, k = 1, \ldots, K$ (which are analogous to the nodes of the SOM [12]). A uniform prior is imposed over the latent points $\boldsymbol{x}_k$, leading to $p(\mathbf{t}) = \sum_{k=1}^{K} p(\mathbf{t}|\boldsymbol{x}_k)p(\boldsymbol{x}_k) = K^{-1}\sum_{k=1}^{K} p(\mathbf{t}|\boldsymbol{x}_k)$.

The conditional data distribution, $p(\mathbf{t}|\boldsymbol{x}_k)$, is modelled as a member of the exponential family in a parameterised functional form [2]

$$p_B(\mathbf{t}|\boldsymbol{x}_k, \boldsymbol{\Theta}) = \exp\left\{\boldsymbol{f}_{\boldsymbol{\Theta}}(\boldsymbol{x}_k)\mathbf{t} - B(\boldsymbol{f}_{\boldsymbol{\Theta}}(\boldsymbol{x}_k))\right\} p_0(\mathbf{t}). \tag{1}$$

Here $B(\boldsymbol{f}_{\boldsymbol{\Theta}}(\boldsymbol{x}_k)) = \ln \int \exp(\boldsymbol{f}_{\boldsymbol{\Theta}}(\boldsymbol{x}_k)\mathbf{t})p_0(\mathbf{t})d\mathbf{t}$ denotes the cumulant generating function of $p(\mathbf{t}|\boldsymbol{x}_k)$, $p_0(\mathbf{t})$ is a factor independent of the parameter $\boldsymbol{\Theta}$, $\mathbf{t} \in \Re^D$ denotes an observed data point, $\boldsymbol{x}_k \in \mathcal{H}$ is the k-th latent space point, and the nonlinearity $\boldsymbol{f}(\cdot)$ is, for convenience, of the form $\boldsymbol{f}_{\boldsymbol{\Theta}}(\boldsymbol{x}_k) = \boldsymbol{\Theta}\boldsymbol{\phi}(\boldsymbol{x}_k)$, where $\boldsymbol{\Theta} \in \Re^{D\times M}$ is a parameter matrix and $\boldsymbol{\phi}(\cdot) = (\phi_1(\cdot), ..., \phi_M(\cdot))^T, \phi_m(\cdot) : \mathcal{H} \to \Re$, is a fixed set of M non-parametric nonlinear basis functions. These could be any smooth functions; typically Gaussian radial basis functions are employed. A linear basis function $\phi_0(\boldsymbol{x}) = 1, \forall \boldsymbol{x}$, may be included to account for the bias term. The notation $\boldsymbol{\phi}_k = \boldsymbol{\phi}(\boldsymbol{x_k})$ will be used as a shorthand.

LTMs are trained to maximize the likelihood of the training set $\zeta = \{\mathbf{t}_1, ..., \mathbf{t}_N\}$ via an EM algorithm [10], the M-step of which consists of solving

$$\mathbf{T}\mathbf{R}^T\boldsymbol{\Phi}^T = \mathbf{b}(\boldsymbol{\Theta}\boldsymbol{\Phi})\boldsymbol{G}\boldsymbol{\Phi}^T, \tag{2}$$

where the function $\mathbf{b}(\cdot)$ denotes the gradient of the cumulant function $B(\cdot)$[1], $\boldsymbol{\Phi}$ is an $M \times K$ matrix with $\boldsymbol{\phi}_k$ in its k-th column, $\mathbf{T}$ is the data matrix including N data vectors $\{\mathbf{t}_n\}$ as columns, $\mathbf{R} = (R_{kn})_{k=1,...,K,n=1,...,N}$ and $\boldsymbol{G}$ is a diagonal matrix with elements $g_{kk} = \sum_{n=1}^{N} R_{kn}$, where R_{kn}, computed via Bayes' theorem in the E-step,

$$R_{kn} = p(\boldsymbol{x}_k|\mathbf{t}_n) = \frac{p(\mathbf{t}_n|\boldsymbol{x}_k, \boldsymbol{\Theta})p(\boldsymbol{x}_k)}{\sum_{k'=1}^{K} p(\mathbf{t}_n|\boldsymbol{x}_{k'}, \boldsymbol{\Theta})p(\boldsymbol{x}_{k'})}, \tag{3}$$

[1] it is the inverse link function [13] of the noise distribution.

is the ‘responsibility’ of the latent point $\boldsymbol{x}_k$ for generating $\mathbf{t}_n$.

For visualisation purposes, the latent space representation of a point $\mathbf{t}_n$ is taken to be the mean of the posterior distribution $p(\boldsymbol{x}_k|\mathbf{t}_n)$ over the latent space points.

Note that the generative latent trait model defines a density in the data space, using a smooth mapping from the latent space to the data space,

$$\boldsymbol{z} : \mathcal{H} \rightarrow \Re^D, \boldsymbol{z}(\boldsymbol{x}_k) = \boldsymbol{b}(\boldsymbol{\Theta\Phi}(\boldsymbol{x}_k)). \tag{4}$$

We refer to the manifold $\boldsymbol{z}(\mathcal{H})$ as the *projection manifold* of the LTM.

3 General Framework for Hierarchical Latent Trait Models

When dealing with large and complex data sets, a single global visualisation plot is often not sufficient. To be able to capture the interesting intrinsic information when visualizing complex data sets as much as possible, hierarchical visualisation systems have been proposed and developed in the literature, [6], [16]. In [6], a locally linear hierarchical visualisation system is introduced. We have recently extended this system to non-linear GTM projection manifolds in [16]. In this section we provide a general formulation of hierarchical latent trait mixture models.

The hierarchical LTM arranges a set of LTMs and their corresponding plots in a tree structure $\mathcal{T}$. The *Root* is at level 1, children of level-ℓ models are at level $\ell+1$.

Each model $\mathcal{M}$ in the hierarchy, except for *Root*, has an associated parent-conditional mixture coefficient, or prior, $\pi(\mathcal{M}|Parent(\mathcal{M}))$. The priors are non-negative and satisfy the consistency condition: $\sum_{\mathcal{M}\in Children(\mathcal{N})} \pi(\mathcal{M}|\mathcal{N}) = 1$. Unconditional priors for the models are recursively calculated as follows: $\pi(Root) = 1$, and for all other models

$$\pi(\mathcal{M}) = \prod_{i=2}^{Level(\mathcal{M})} \pi(Path(\mathcal{M})_i|Path(\mathcal{M})_{i-1}), \tag{5}$$

where $Path(\mathcal{M}) = (Root, \ldots, \mathcal{M})$ is the $\mathcal{P}$-tuple of nodes defining the path of length $\mathcal{P}$ in $\mathcal{T}$ from *Root* to $\mathcal{M}$.

The distribution given by the hierarchical model is a mixture of leaves[2] of $\mathcal{T}$

$$P(\mathbf{t}|\mathcal{T}) = \sum_{\mathcal{M} \in Leaves(\mathcal{T})} \pi(\mathcal{M}) P(\mathbf{t}|\mathcal{M}). \tag{6}$$

Non-leaf models not only play their role in the process of creating the hierarchical model, but in the context of data visualisation can be useful for determining the relationship between sub-plots in the hierarchy.

3.1 Training

The hierarchical LTM is trained using EM to maximize its likelihood with respect to the data sample $\zeta = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_N\}$. Training of a hierarchy of LTMs proceeds in a recursive fashion. First, the *Root* LTM is trained and used to visualize the data. Then the user identifies interesting regions on the visualisation plot that they would like to model in a greater detail[3].

Having trained models $\mathcal{N}$ at level ℓ, the expectation of the complete data likelihood of level-$(\ell + 1)$ is

$$\begin{aligned} < \mathcal{L}_{comp}^{\ell+1} > \;=\; & \sum_{n=1}^{N} \sum_{\mathcal{N} \in Nodes(l)} P(\mathcal{N}|\mathbf{t}_n) \sum_{\mathcal{M} \in Children(\mathcal{N})} P(\mathcal{M}|\mathcal{N}, \mathbf{t}_n) \\ & \sum_{k=1}^{K_{\mathcal{M}}} R_{kn}^{\mathcal{M}} \ln\{\pi(\mathcal{N})\pi(\mathcal{M}|\mathcal{N}) P(\mathbf{t}_n, \boldsymbol{x}_k^{\mathcal{M}})\} \end{aligned} \tag{7}$$

3.1.1 E-step

In the E-step, we estimate the posterior distribution of all hidden variables, using the "old" values of LTM parameters. Given a data point $\mathbf{t}_n$, we compute the model responsibilities corresponding to the competition among models belonging to the

[2] Leaves($\mathcal{T}$) is the set of nodes of $\mathcal{T}$ without children.

[3] We will describe the sub-model initialisation in section 3.2.

same parent as

$$P(\mathcal{M}|Parent(\mathcal{M}), \mathbf{t}_n) = \frac{\pi(\mathcal{M}|Parent(\mathcal{M}))P(\mathbf{t}_n|\mathcal{M})}{\sum_{\mathcal{M}' \in [\mathcal{M}]} \pi(\mathcal{M}'|Parent(\mathcal{M}))P(\mathbf{t}_n|\mathcal{M}')}, \tag{8}$$

where

$$[\mathcal{M}] = Children(Parent(\mathcal{M})). \tag{9}$$

Imposing $P(Root|\mathbf{t}_n) = 1$, the unconditional (on parent) model responsibilities are recursively determined by

$$P(\mathcal{M}|\mathbf{t}_n) = P(\mathcal{M}|Parent(\mathcal{M}), \mathbf{t}_n)P(Parent(\mathcal{M})|\mathbf{t}_n). \tag{10}$$

Responsibilities of the latent space centres $\boldsymbol{x}_k^{\mathcal{M}}$, $k = 1, 2, ..., K_{\mathcal{M}}$, corresponding to the competition among the latent space centres in each model $\mathcal{M}$, are calculated using (3).

3.1.2 M-step

In the M-step, we estimate the parameters using the posterior over hidden variables computed in the E-step.

Parent-conditional mixture coefficients are determined by

$$\pi(\mathcal{M}|Parent(\mathcal{M})) = \frac{\sum_{n=1}^{N} P(\mathcal{M}|\mathbf{t}_n)}{\sum_{n=1}^{N} P(Parent(\mathcal{M})|\mathbf{t}_n)}. \tag{11}$$

Parameters $\mathbf{\Theta}^{(\mathcal{M})}$ of the LTM $\mathcal{M}$ are calculated by solving

$$\mathbf{T}\mathbf{R}^{(\mathcal{M})T}\mathbf{\Phi}^T = \mathbf{b}(\mathbf{\Theta}^{(\mathcal{M})}\mathbf{\Phi})\boldsymbol{G}^{(\mathcal{M})}\mathbf{\Phi}^T \tag{12}$$

where $\mathbf{R}^{(\mathcal{M})} = (R_{kn}^{\mathcal{M}})_{k=1,...,K,n=1,...,N}$. $R_{kn}^{\mathcal{M}}$ are scaled (by (10)) responsibilities (3), $R_{kn}^{\mathcal{M}} = P(\mathcal{M}|\mathbf{t}_n)R_{kn}$; $\boldsymbol{G}^{(\mathcal{M})}$ is a diagonal matrix with elements $g_{kk}^{\mathcal{M}} = \sum_{n=1}^{N} R_{kn}^{\mathcal{M}}$.

When solving (12), if the link function $\mathbf{b}(\cdot)$ is the identity, one gets the closed form M-step of HGTM[4] [16], but in general a non-linear optimization algorithm is

[4] Even though we treat GTM as a special case of LTM with spherical Gaussian noise model,

required. In the simplest case, we may employ a gradient inner loop M-step[5]:

$$\Delta\mathbf{\Theta}^{(\mathcal{M})} \propto \left\{ \mathbf{T}\mathbf{R}^{(\mathcal{M})T} - \mathbf{b}(\mathbf{\Theta}^{(\mathcal{M})}\mathbf{\Phi})\boldsymbol{G}^{(\mathcal{M})} \right\} \mathbf{\Phi}^T. \tag{13}$$

3.2 Model initialization

When initializing sub-models there are two things to determine: the number of sub-models and the initial parameters of the sub-models. We view the problem of initializing sub-model parameters primarily as one of locating which region each sub-model should be responsible for. To do this, regions of interest are defined by the user in the latent (visualisation) space. The points $\mathbf{c}_i$ selected in the latent space $\mathcal{H}$ correspond to the "centres" of these regions.

The "regions of interest" are transformed into the data space as Voronoi compartments [1] defined by the mapped points $\boldsymbol{z}(\mathbf{c}_i) \in \mathcal{D}$, where $\boldsymbol{z}$ is the map (4) of the corresponding LTM. In the case of a Gaussian noise model, the child LTMs are initialized by local PCA in the corresponding Voronoi compartments [16]. When using other noise models such as Bernoulli or multinomial distributions, the PCA-initialised LTMs are in addition individually trained (section 2) in their corresponding Voronoi compartments for 1 EM iteration. The EM iteration "settles" the component LTMs to their corresponding modelling regions. Empirically, this initialisation strategy works very well. We perform the additional initialization step when the PCA initialisation alone does not "match" the noise distribution well, e.g. when the noise distribution is non-symmetric and data space is discrete.

After the initialisation, a full hierarchical training described in section 3.1 is used.

(1) does not account for the "width" parameter. We decided to use the simplified formulation (1), because it is sufficient for all other interesting noise models, such as Bernoulli, Poisson, Multinomial etc. In the case of spherical Gaussian noise model, solving (12) sets the means of the Gaussians and the width parameter needs to be updated as in [16].

[5]in this partial M-step we could alternatively use iterative reweighted least squares [18].

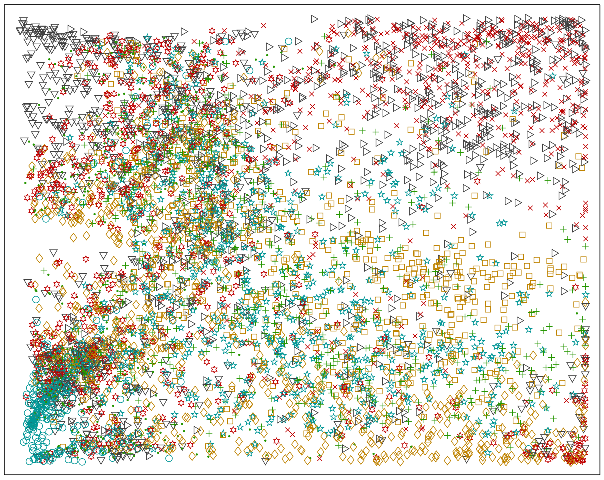

Figure 1: An example of strongly overlapping clusters

3.3 Plotting the projections

We adopt the strategy used in [6, 16] that suggests to take advantage of the probabilistic nature of our model by plotting projections of all the data points on every plot, but modifying the intensity in proportion to the responsibility $P(\mathcal{M}|\ \mathbf{t}_n)$ (10) which each plot (sub-model $\mathcal{M}$) has for the data point $\mathbf{t}_n$. Points that are not well captured by a particular plot will appear with low intensity.

4 Unsupervised learning of mixtures of LTMs

So far, we have developed a general framework for a visualisation hierarchy. The user selects the 'regions of interest' to refine the visualisation model. This method is powerful when the clusters are separated clearly in the 2-D latent space. On the other hand, when facing a "messy" plot like that in Figure 1, where thousands of data points are shown (with densely clustered and overlapping projections), the user may be unable to determine where sub-models should be placed. In order to resolve this problem, we extend our current algorithm by providing an automated technique for deciding the number of sub-models and initialising their location.

Note that in this section we will focus solely on the algorithm for mixture models.

4.1 MML formulation for unsupervised learning of mixture models

Given a set $\zeta = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\}$ of data points, minimum message length (MML) strategies select, among the models inferred from ζ, the one which minimizes length of the message transmitting ζ [17]. Given that the data is modeled by a parametric probabilistic model $P(\zeta|\boldsymbol{\theta})$, the message consists of two parts – one specifying the model parameters, the other specifying the data given the model: $\text{Length}(\boldsymbol{\theta}, \zeta) = \text{Length}(\boldsymbol{\theta}) + \text{Length}(\zeta|\boldsymbol{\theta})$.

Recently, Figueiredo and Jain [8] extended the MML framework to unsupervised learning of mixture models; the algorithm selects the "appropriate" number of components while the parameters of each model are estimated in the usual way. The novelty of their proposed approach is that parameter estimation and model selection are integrated in a single algorithm, rather than using a model selection criterion on a set of pre-estimated candidate models.

The particular form of MML criterion adopted in [8] is of the form $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \boldsymbol{\mathcal{L}}(\boldsymbol{\theta}, \zeta)$, where

$$\boldsymbol{\mathcal{L}}(\boldsymbol{\theta}, \zeta) = -\log P(\boldsymbol{\theta}) - \log P(\zeta|\boldsymbol{\theta}) + \frac{1}{2}\log|\mathbf{I}(\boldsymbol{\theta})| + \frac{c}{2}\left(1 + \log\frac{1}{12}\right), \tag{14}$$

where $\mathbf{I}(\boldsymbol{\theta})$ is the expected Fisher information matrix, $|\mathbf{I}(\boldsymbol{\theta})|$ is its determinant, and c is the number of free parameters, i.e. the dimension of $\boldsymbol{\theta}$.

By imposing a non-informative Jeffreys' prior [3] on both the vector of mixing coefficients $\{\pi(\mathcal{M})\}$ and the parameters $\boldsymbol{\Theta}^{(\mathcal{M})}$ of individual mixture components [8], the equation (14) becomes

$$\boldsymbol{\mathcal{L}}(\boldsymbol{\theta}, \zeta) = \frac{Q}{2}\sum_{\pi(\mathcal{M})>0}\log\left(\frac{N\cdot\pi(\mathcal{M})}{12}\right) + \frac{A}{2}\log\frac{N}{12} + \frac{A(Q+1)}{2} - \log P(\zeta|\boldsymbol{\theta}), \tag{15}$$

where A is the number of mixture components with positive prior $\pi(\mathcal{M}) > 0$ and Q is the number of free parameters of each individual mixture component.

Minimizing (15) with respect to $\pi(\mathcal{M})$ under the constraint that the priors $\pi(\mathcal{M})$ sum to 1, the following re-estimation formulas are obtained [8]:

$$\hat{\pi}(\mathcal{M}) = \frac{\max\left\{0,\ -\frac{Q}{2} + \sum_{n=1}^{N} P(\mathcal{M}|\mathbf{t}_n)\right\}}{\sum_{\mathcal{M}'} \max\left\{0,\ -\frac{Q}{2} + \sum_{n=1}^{N} P(\mathcal{M}'|\mathbf{t}_n)\right\}}, \tag{16}$$

where component responsibilities $P(\mathcal{M}|\mathbf{t}_n)$ are determined by

$$P(\mathcal{M}|\mathbf{t}_n) = \frac{\pi(\mathcal{M}) P(\mathbf{t}_n|\mathcal{M})}{\sum_{\mathcal{M}'} \pi(\mathcal{M}') P(\mathbf{t}_n|\mathcal{M}')}, \tag{17}$$

$$\pi(\mathcal{M}) = \frac{\sum_{n=1}^{N} P(\mathbf{t}_n|\mathcal{M})}{\sum_{\mathcal{M}'=1}^{A} \sum_{n=1}^{N} P(\mathbf{t}_n|\mathcal{M}')} \tag{18}$$

Free parameters of the individual LTMs are fitted to the data ζ using the EM algorithm outlined in section 3 applied to mixtures of LTMs[6]. Note that LTMs corresponding to zero $\hat{\pi}(\mathcal{M})$ become irrelevant and so (16) effectively performs component annihilation [8].

4.2 The algorithm

Given the training data $\zeta = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\}$, we use the MML approach to find the "appropriate" number of mixture component LTMs that "explain" ζ in a probabilistic manner. LTMs that are good probabilistic generating models of the data capture the data distribution well and hence yield "good" visualisation plots[7]. To start the training process, we choose the maximum number of components A_{max} we are willing to consider. Then, we initiate the component LTMs by randomly selecting A_{max} points from ζ and applying the method described in section 3.2. The

[6]A mixture of LTMs can be considered a two-level hierarchical LTM. Mixture components are children of the *root*.

[7]This is a tricky issue, since while we can measure the quality of probabilistic models e.g. via likelihood, there is no universal quality measure for visualisation plots. But intuitively, good probabilistic properties of a LTM mean that the projection manifold follows closely the data distribution and so the visualisation plot is a "good" representation of the data distribution.

selected A_{max} points act as centres of regions of interest in the data space. In other words, they play the role of vectors $\boldsymbol{z}(\mathbf{c}_i)$ from section 3.2.

As in [8], we adopt the component-wise EM (CEM) algorithm [7], i.e. rather than simultaneously updating all the LTMs, we first update the parameters $\boldsymbol{\Theta}^{(1)}$ of the first LTM (12), while parameters of the remaining LTMs are fixed, then we recompute the component responsibilities $\{P(\mathcal{M}|\mathbf{t}_n)\}$ (17) and mixture coefficients $\{\hat{\pi}(\mathcal{M})\}$ (16) for all components in the mixture. After this, we move to the second component, update $\boldsymbol{\Theta}^{(2)}$ in the same way, and recompute $\{P(\mathcal{M}|\mathbf{t}_n)\}$, $\{\hat{\pi}(\mathcal{M})\}$, etc., looping through all mixture components. If one of the component LTMs dies ($\hat{\pi}(\mathcal{M}) = 0$), redistribution of its probability mass to the remaining components increases their chance of survival. After convergence of CEM, we still have to check whether a shorter message length can be achieved by having a smaller number of mixture LTMs (down to $A = 1$).[8] This is achieved by iteratively killing off the weakest LTM (with the smallest $\hat{\pi}(\mathcal{M})$) and re-running CEM until convergence. Finally, the winning mixture of LTMs is the one that leads to the shortest message length $\boldsymbol{\mathcal{L}}(\boldsymbol{\theta}, \zeta)$ (15).

To demonstrate this algorithm, we did an experiment on a toy data set of 800 points $\mathbf{t} = (t_1, t_2, t_3)^T$ lying on four two-dimensional manifolds ("humps") (see Figure 2 (a)). We associated the points in the four "humps" with four different classes, $\mathcal{C}_i$, $i = 1, 2, 3, 4$, having four different labels. After training[9] ($A_{max} = 10$), a 7-component mixture was constructed. Projection manifolds of the 7 LTMs are shown in Figure 2 (b). Note that 7 child plots provide understandable subgroups of the data; and that the 6 projection manifolds closely approximate the four "humps" of the original generating manifold. The corresponding hierarchy of visualisation plots can be seen in Figure 3.

[8] If we knew that the number of mixture components was no less than some number A_{min}, we would stop at $A = A_{min}$ [8].

[9] we used LTMs with Gaussian noise model

We stress that there is no contradiction between the number of components (7) in the final mixture of LTMs and the dataset composed of four "humps". There is no driving force in the MML formalism to achieve this and this is not the point of our study. The important thing is that the MML method finds a good number of subplots so that the overall probability of the data set is high (good projections) and the mixture model is not too complex (unnecessarily high number of subplots). At the same time, the MML method *automatically* finds appropriate *positions* of the projection manifolds in the data space.

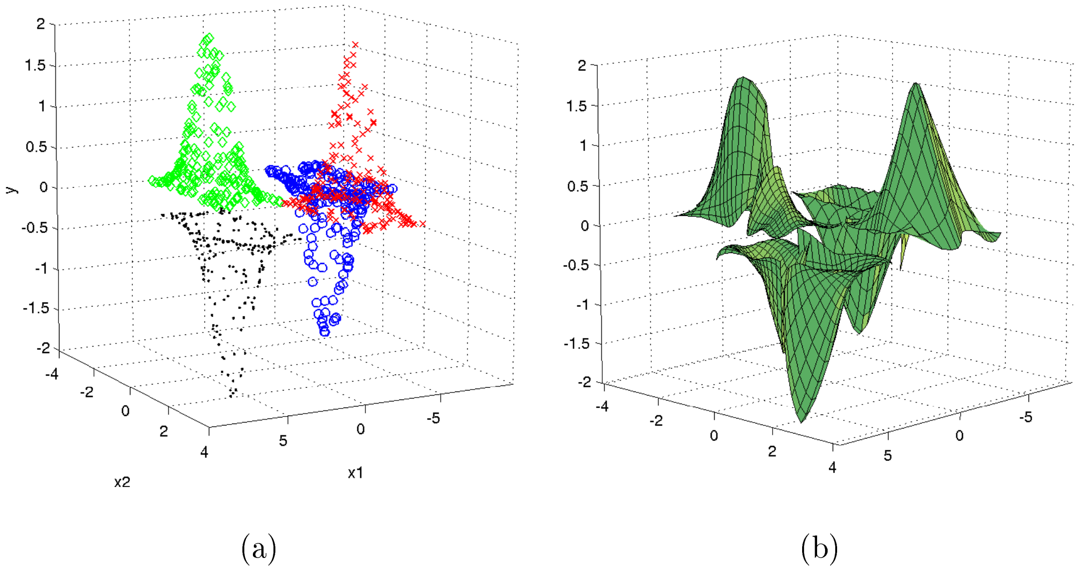

Figure 2: (a) A two dimensional manifolds in data space; (b) Projection manifolds in data space of the second-level LTMs trained on the toy data.

5 Semi-Supervised Learning of Visualisation Hierarchies

The procedure for unsupervised learning of mixture models discussed in section 4 becomes more complex for nodes (subplots) in hierarchical models at levels > 2. In this case, we should consider model responsibilities of parent nodes for the data

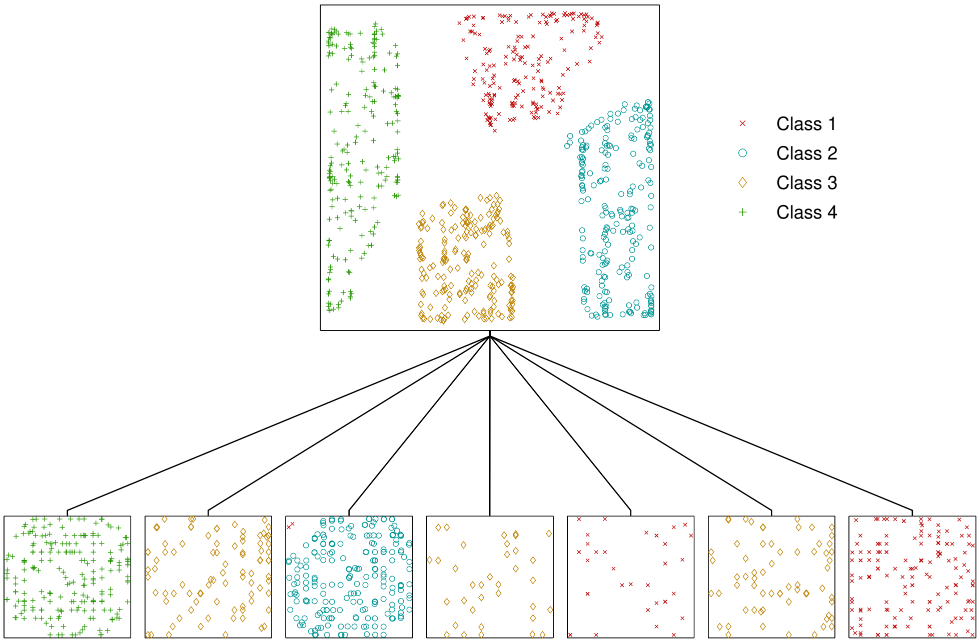

Figure 3: Visualisation of the toy data constructed with unsupervised MML.

points and these are recursively propagated as we incrementally build the hierarchy. So equations (8) and (10) are used in hierarchical models instead of equation (17) used in the simple mixture case. Also equation (5) is applied in place of equation (18).

The proposed system for constructing hierarchies of non-linear visualisation plots is similar to the one described in [16]. The important difference is that now, given a parent plot, its children are not always constructed in the interactive way by letting the user identify "regions of interest" for the sub-plots. In densely populated higher-level plots with many overlapping projections, this may not be possible. Instead, we let the user decide whether they want the children to be constructed in an interactive or unsupervised way.

In the unsupervised case, we use the MML technique to decide the "appropriate" number and approximate position of children LTMs. We collect data points from ζ for which the parent LTM has responsibility higher than a threshold Δ (in our experiments Δ was set to[10] 0.9). We then run MML-based learning of *mixtures* of

[10] Other values for Δ, e.g. $\Delta = 0.8$, could have been used. However, the final local mixture of

LTMs (section 4.2) on this reduced data set. The resulting local mixture is viewed as an *initialization* for the full EM algorithm for training *hierarchies* of LTMs described in section 3.1. This way, the "appropriate" number of LTMs is determined along with their initial locations.

5.1 Experiments

In this section we illustrate the semi-supervised hierarchical LTM visualisation algorithm on three "real-world" data collections.

Although the algorithm is derived in a general setting in which individual LTMs $\mathcal{M}$ in the hierarchy can have different sets of latent points $\boldsymbol{x}_k^{\mathcal{M}}$, $k = 1, 2, ..., K_{\mathcal{M}}$, and basis functions ϕ_j, $j = 1, 2, ..., M_{\mathcal{M}}$, in the experiments reported here, we used a common configuration for all models in the hierarchy. In particular, the latent space $\mathcal{H}$ was taken to be the two-dimensional interval $\mathcal{H} = [-1, 1] \times [-1, 1]$, the latent points $\boldsymbol{x}_k^{\mathcal{M}} \in \mathcal{H}$ were positioned on a regular 15×15 square grid and there were 16 radial basis functions ϕ_j centered on a regular 4×4 square grid. As usual in the GTM literature, the basis functions were spherical Gaussians of the same width[11] $\sigma = 1.0$. We account for a bias term by using an additional constant basis function $\phi_0(\boldsymbol{x}) = 1$, for all $\boldsymbol{x} \in \mathcal{H}$. If the noise model in LTM is Gaussian, we always consider only spherical Gaussians, as in the original formulation of GTM [4]. Complete training equations for hierarchical GTM can be found in [16].

Note that in the interactive mode, the "centres" of the regions of interest are shown as circles labeled by numbers. These numbers determine the order of the corresponding child LTM subplots from left to right.

LTMs in the hierarchy is not very sensitive to the exact value of Δ, since this is just an initialization step, before running full EM for hierarchical LTM.

[11] The width of the basis functions is related to the "flexibility" of the generalized linear regression, $\boldsymbol{f}_{\boldsymbol{\Theta}}(\boldsymbol{x}) = \boldsymbol{\Theta}\boldsymbol{\phi}(\boldsymbol{x})$, from the latent space to the data space. For a discussion on appropriate values for σ see [4, 16].

5.1.1 Image segmentation data

As the first example we visualize image segmentation data obtained by randomly sampling patches of 3x3 pixels from a database of outdoor images. The patches are characterized by 18 continuous attributes and are classified into 4 classes: *cement + path*, *brickface + window*, *grass + foliage* and *sky* (see [16]). The final visualisation plot of hierarchical LTM with Gaussian noise model can be seen in Figure 4. The *Root* plot contains clusters of overlapping projections. Six plots at the second level were constructed using the unsupervised MML technique ($A_{max} = 10$). Note that the second-level LTMs already separate the four classes fairly well and are readable enough to be analysed further in the interactive mode. For example, we selected two and four "centres" for regions of interest (shown as circles) in the second and fifth level-two plots, respectively.

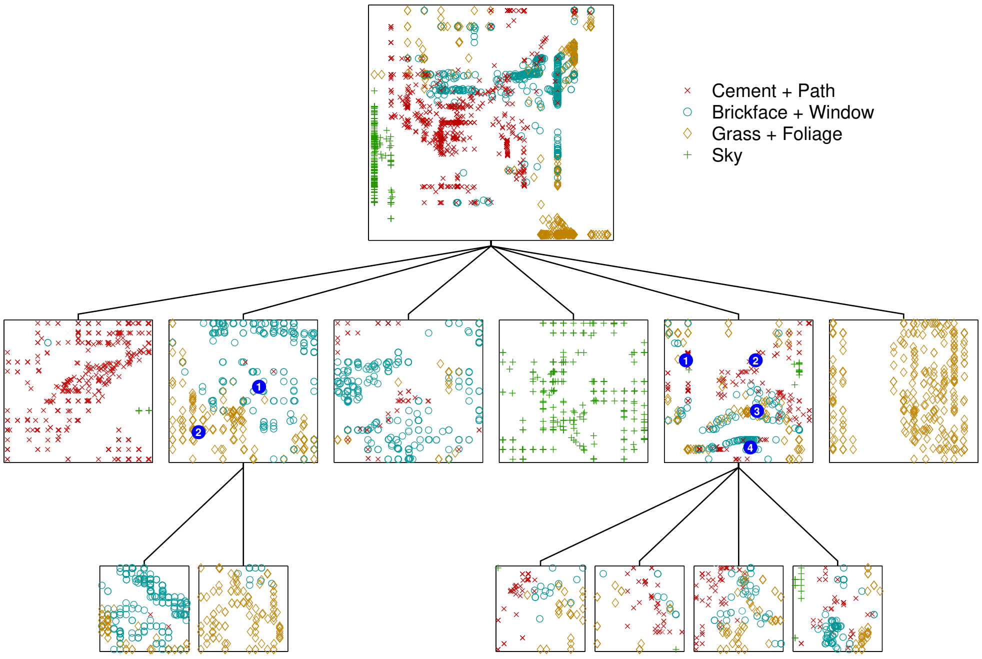

Figure 4: Hierarchical visualisation of the image segmentation data constructed in a semi-interactive way.

5.1.2 Text data set

Since our system is based on the LTM, it can deal with discrete data sets. As an illustration, we tested our system on a text-collection of 8000 documents formed by 10 topic classes from the newsgroup[12] text corpus. The documents were binary encoded over a dictionary of $D = 100$ words. The initial pre-processing, word-stemming and removal of 'stop-words' was done using the Bow toolkit[13]. To account for binary encoding, the Bernoulli noise model was employed.

The visualisation plot generated in a semi-interactive way is shown in Figure 5. The 'Root' is extremely densely populated with highly overlapping data projections. After using the unsupervised MML technique ($A_{max} = 10$), a 4-component mixture of LTMs was obtained on the second level. Sub-clusters in these four level-two plots are decipherable. The user can now choose more detailed regions of interest by using the interactive mode.

As in [16], this system also includes the child-modulated ancestor plot technique, which can visualise the regions captured by a particular child LTM $\mathcal{M}$. This is done by modifying all the ancestor plots up to the *Root*, so that instead of the ancestor responsibilities, the responsibilities of the model $\mathcal{M}$, $P(\mathcal{M}|\mathbf{t}_n)$, are used in every plot on the path from $\mathcal{M}$ to *Root* . This improves the understanding of the relationships among sub-plots in the visualisation hierarchy. In Figure 6, we highlight the visualisation plots which include the data points from the topic 'sci.space', captured by the first model on the 4th-level.

[12] http://www.cs.cmu.edu/˜textlearning

[13] http://www-2.cs.cmu.edu/˜mccalum/bow

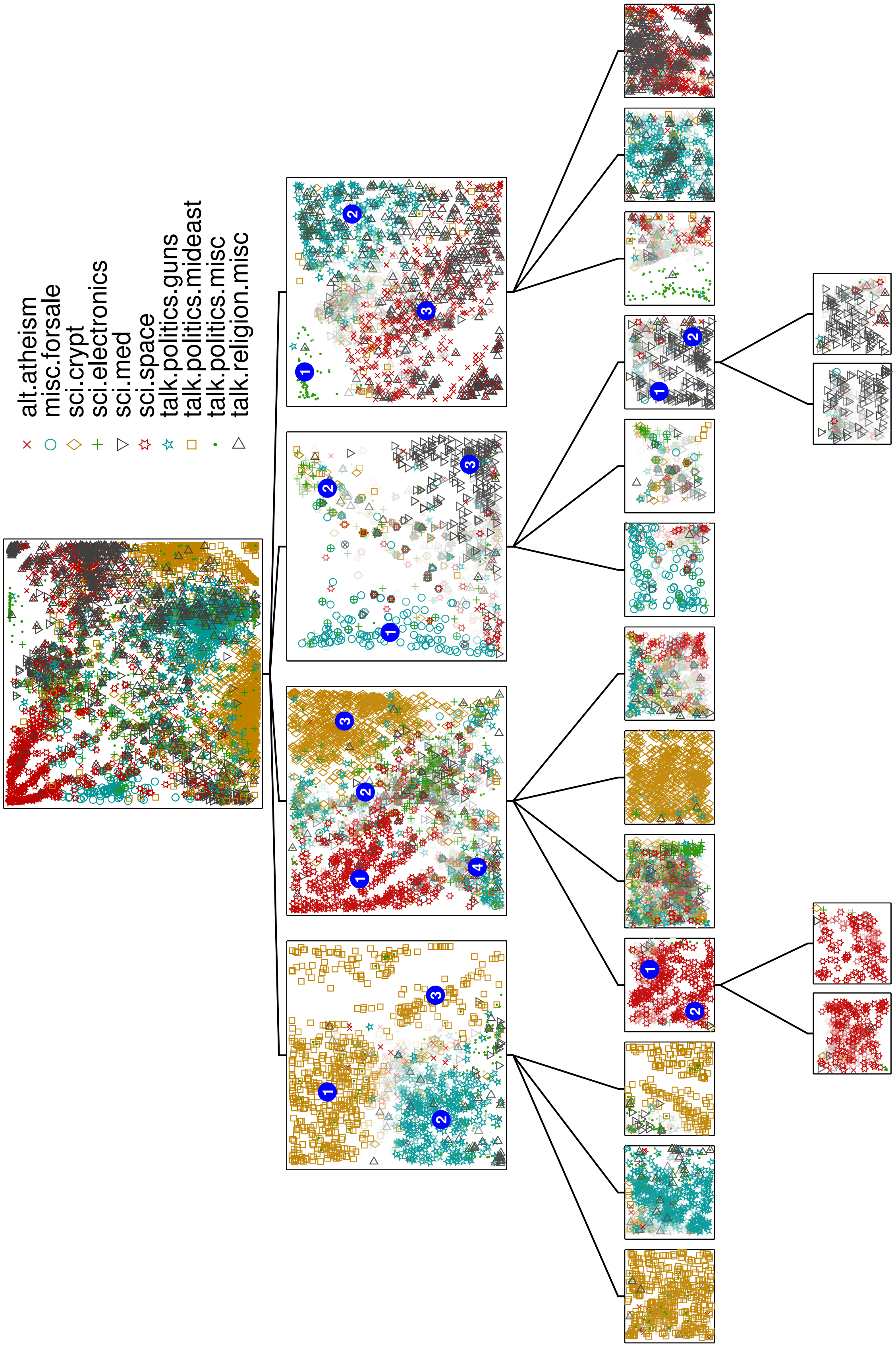

Figure 5: Hierarchical visualisation of the document data constructed in a semi-interactive way.

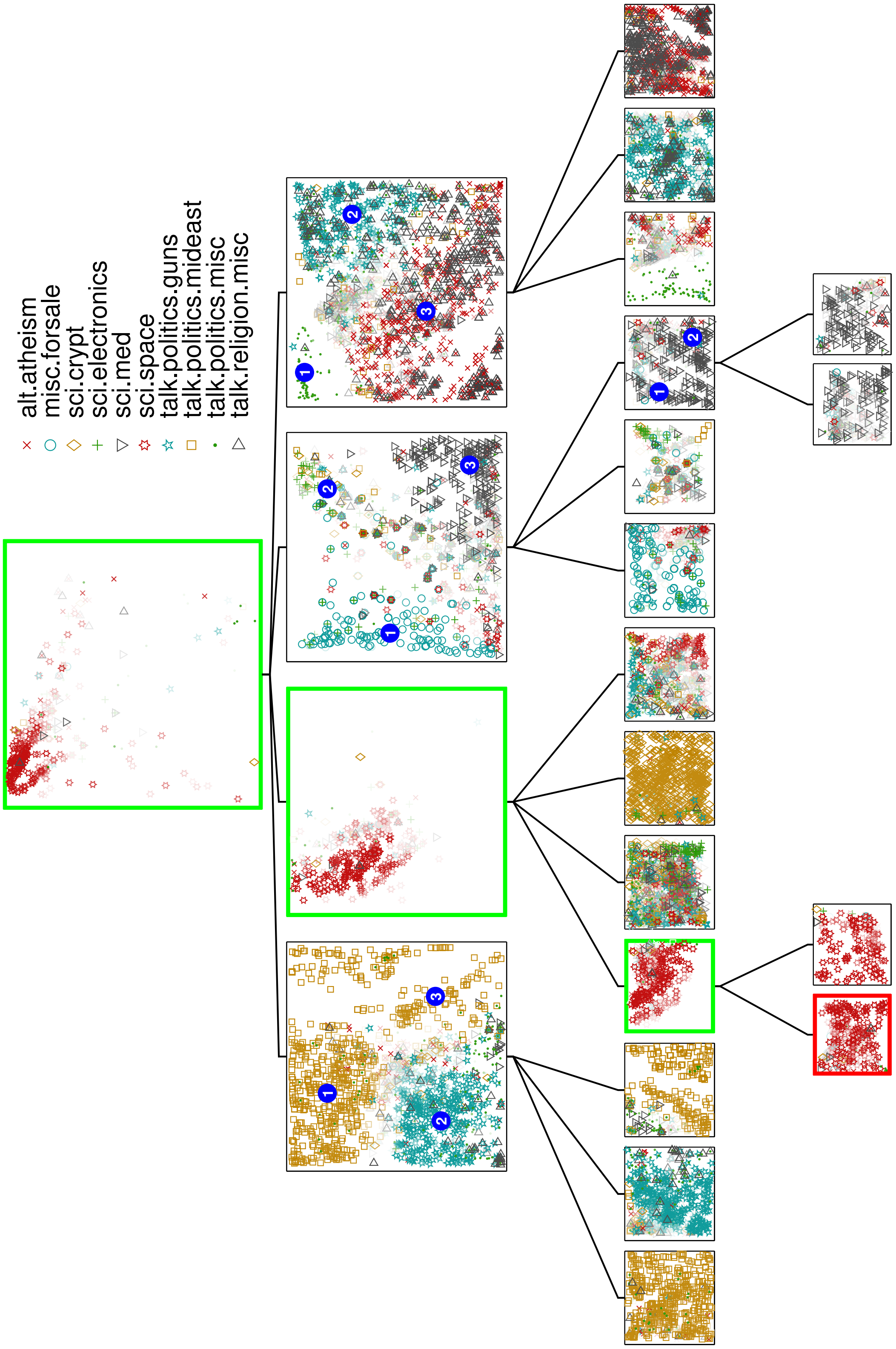

Figure 6: Hierarchical visualisation of the document data constructed in a semi-interactive way. The set of points captured by the first LTM at level 4 of the hierarchy is highlighted in the visualisation plots of all its ancestors.

5.1.3 Protein localization site data set

In the last experiment we visualise a data set of 1484 proteins encoded as real-valued vectors[14]. The 6-dimensional data points[15] are classified into 10 classes (localization sites). The class names are shown in the legend of Figure 7. Here we demonstrate the application of the unsupervised MML technique at a lower level in the hierarchy.

We trained a four-level hierarchy of LTMs (Gaussian noise model) on the protein data and the resulting projections are displayed in Figure 7. Again, the *Root* plot looks 'messy'. Two plots at the second level were constructed using the unsupervised MML technique ($A_{max} = 10$). The first level-two plot is legible enough for the user to select the 'centres' in the interactive mode (as shown in the figure). We used the MML algorithm as an initialisation technique for constructing child plots of the second level-two plot ($A_{max} = 5$). Two resulting child plots included readable clusters.

Note that visualisation plots for this data set do not provide a good separation, even at lower levels of the hierarchy. It follows that the features used to describe the data are not discriminative enough with respect to the 10 binding site classes. The classes are highly overlapping. Our system enables the user to detect such situations by understanding the underlying data distribution. Our findings are confirmed by poor classification results (around 55% accuracy) obtained on this data set using various classification techniques [9].

[14] The data set can be downloaded from the UCI Machine Learning page: ftp://ftp.ics.uci.edu/pub/machine-learning-databases/yeast/

[15] The original data is 8-dimensional. Two of the dimensions are effectively constant and were removed.

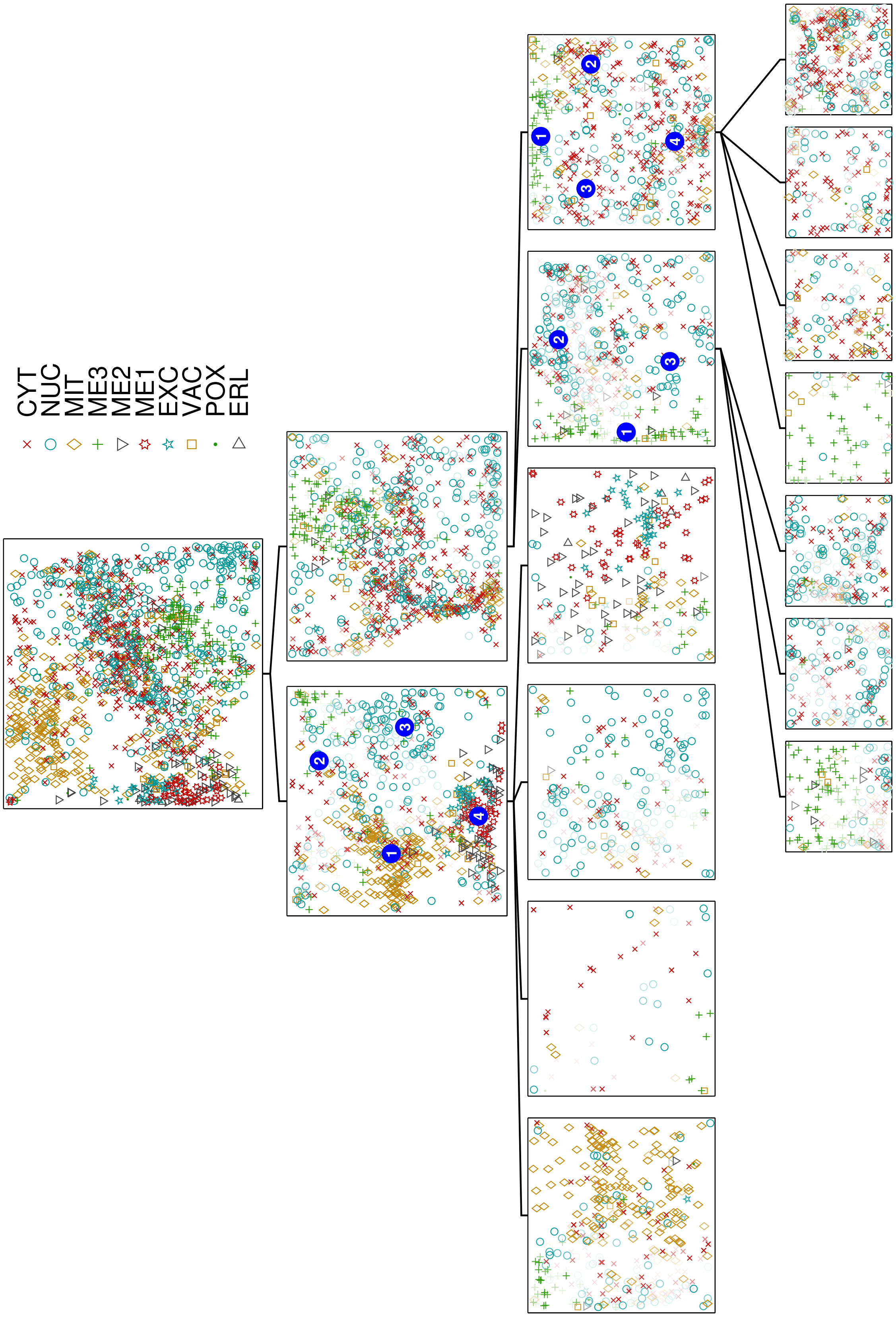

Figure 7: Hierarchical visualisation of the protein data set constructed in a semi-interactive way.

6 Local Magnification Factors of the Latent Trait Manifolds

The term 'magnification factor' [5] refers to the degree of stretching or compression of the latent space when embedded into the data space. Previous experience has indicated that magnification factors are a useful tool for interpreting 2-D non-linear visualisation plots. For example, projections of well-separated dense clusters of data points will be occupy compressed regions on the visualisation plot (small magnification factors), separated by a band of highly stretched area (high magnification factors).

Let us consider the Cartesian coordinate system defined on the latent space and the mapping of this space to a curvilinear coordinate system defined on the manifold embedded in the data space. It has been shown in [5] that for the original GTM formulation, the local magnification factor corresponding to a point $\boldsymbol{x}_0$ in the latent space, defined as the ratio between the area of an infinitesimal rectangle in the latent Cartesian space and the area generated by mapping it through (4) on the projection manifold, is $\sqrt{|\boldsymbol{S}(\boldsymbol{x}_0)|}$, where $|\boldsymbol{S}(\boldsymbol{x}_0)|$ is the determinant of the metric tensor $\boldsymbol{S} = \boldsymbol{\Gamma}^T\boldsymbol{\Gamma}$, where $\boldsymbol{\Gamma}$ denotes the Jacobian of the mapping (4) at $\boldsymbol{x}_0$.

In general,

$$\boldsymbol{\Gamma} = \frac{\partial \boldsymbol{z}(\boldsymbol{x}_0)}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{b}(\boldsymbol{\Theta}\boldsymbol{\phi}(\boldsymbol{x}_0))}{\partial \boldsymbol{x}} = \boldsymbol{F}\boldsymbol{\Theta}\boldsymbol{V} \tag{19}$$

where the $M \times L$ matrix $\boldsymbol{V}$ is equal to $\left(\frac{\partial \phi_m(\boldsymbol{x})}{\partial x_l}_{|\boldsymbol{x}=\boldsymbol{x}_0}\right)_{m=1,\dots,M,l=1,\dots,L}$, and the $D \times D$ matrix $\boldsymbol{F} = \left(\frac{b_{d'}(\boldsymbol{y})}{\partial y_d}_{|\boldsymbol{y}=\Theta\boldsymbol{\phi}(\boldsymbol{x}_0)}\right)_{d'=1,\dots,D,d=1,\dots,D}$ is the Fisher information matrix of the noise distribution. If Gaussian radial basis functions are utilized as $\phi(\cdot)$ then the (l, m)-th element of the matrix $\boldsymbol{V}$ will be $v_{l,m} = -\phi_m(\boldsymbol{x}_0)(x_l - c_{m,l})\sigma^{-2}$ where $c_{m,l}$ denotes the l-th coordinate of the radial basis centre which corresponds to the m-th basis function.

In summary, the magnification factor associated with a point $\boldsymbol{x}_0$ in the latent space is $\sqrt{|\boldsymbol{V}^T\boldsymbol{\Theta}^T\boldsymbol{F}^T\boldsymbol{F}\boldsymbol{\Theta}\boldsymbol{V}|}$.

In the case of Gaussian noise models, the matrix $\boldsymbol{F}^T\boldsymbol{F}$ is the identity matrix. Note also that in all independent noise models this matrix will be diagonal; therefore the increase in computational complexity will not be significant. However, this is not the case for the multinomial trait model (as can be seen in appendix A.3).

As an example we show in Figure 8 the magnification factor plots (log scaled) for the projection hierarchy of the text data set in Figure 5. In general, dark bands in the plots indicate well-separated clusters of points in the data space. For example, there is a dark band slightly left of the center of the eleventh level-three model. The band divides different topics in the data space. From the corresponding model in Figure 5, we see that the left region mostly involves topic 'talk.politics.misc', and the right region contains a mixture of topics.

For a detailed analysis, we focus on the fourth level-three LTM model in Figure 8. The corresponding projection plot in Figure 5 contains mostly documents from a single topic, 'sci.space'. An enlarged (locally scaled) view of the magnification factor plot is presented in Figure 9. There is a dark band around the diagonal line of the plot. Hence, we infer that documents on either side of the band correspond to different clusters and that a change of *sub-topic* happens. The list of 5 most probable dictionary words for each latent space centre of the corresponding LTM is shown in Figure 10. With reference to Figure 9, two clusters can be found on each side of the separating band. Key words for each latent space centre inside the region bounded by the solid border are completely the same and have the same ordering. They appear to refer to documents relating to space shuttle launches. While key words inside the region with the dashed border seem likely to be associated with articles concerning space orbits.

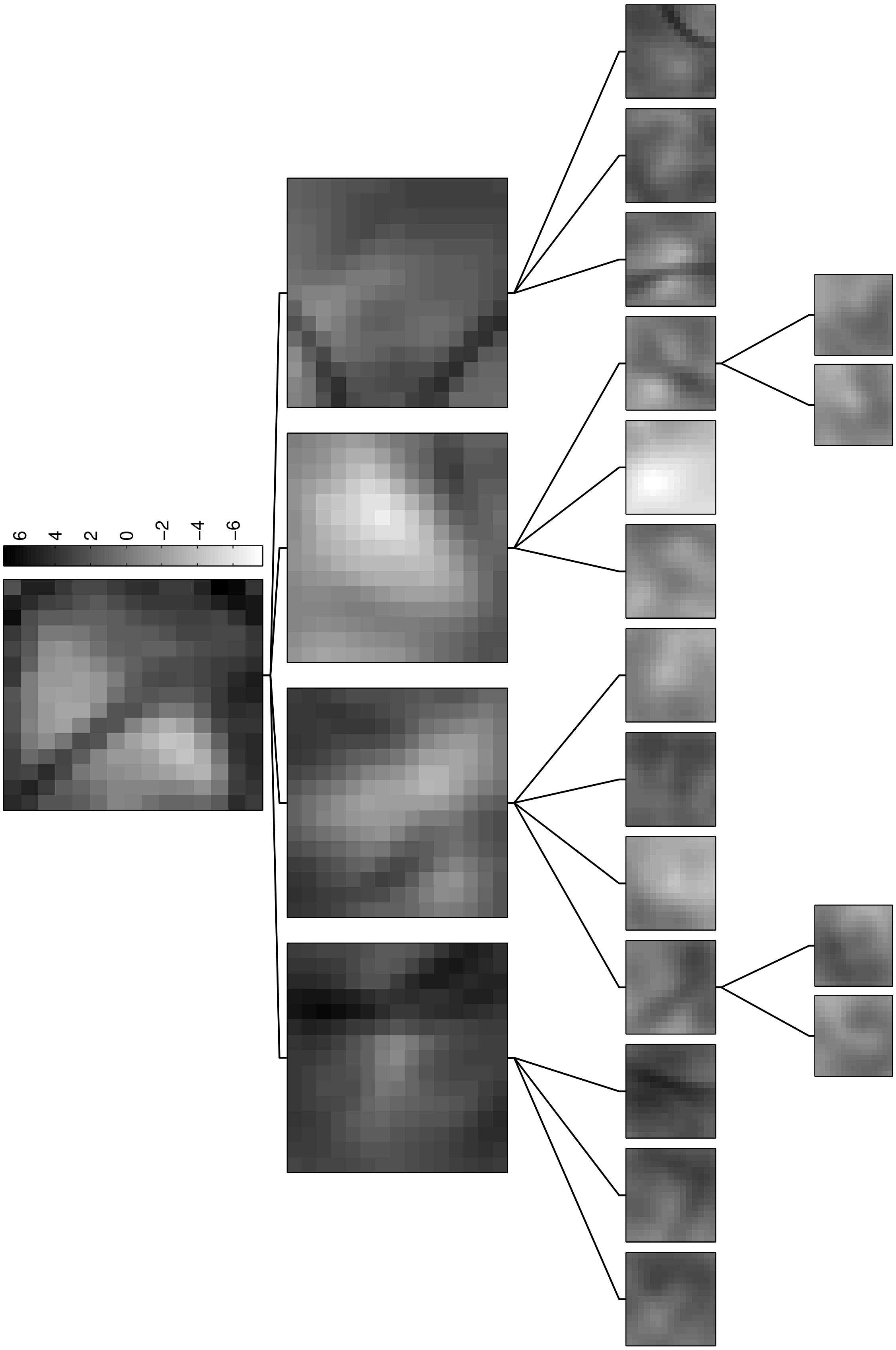

Figure 8: Plots of magnification factors (log2 scaled) in the hierarchy of LTMs fitted on the document data.

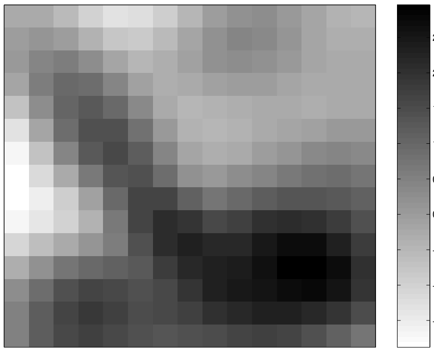

Figure 9: A rescaled visualisation plot of magnification factors for the 4th LTM at level 3 in the hierarchy shown in figure 8.

7 Conclusion

In this paper we have presented a general system for hierarchical visualisation of large data sets which may be of either continuous or discrete type. We also derived formulas for magnification factors in latent trait models. The proposed system gives the user a choice of initializing the child plots of the current plot in either *interactive*, or *automatic* mode. This latter feature of our system is particularly useful when the user has no idea how to choose the area of interest due to highly overlapping dense data projections. The system can be used in many different fields, such as document data mining, tele-communications, bio-informatics, market-basket analysis or information retrieval.

Acknowledgments

This research has been funded by BBSRC grant 92/BIO12093 and Pfizer Central Research. Ata Kaban has been partially supported by re:source, The Council for Museums, Archives and Libraries, Grant Number RE/092, within the University of Paisley. The experiments were carried out with the NETLAB neural network toolbox, available from **http://www.ncrg.aston.ac.uk/netlab**. Yi Sun would

orbit write articl earth launch	write orbit articl earth launch	write orbit articl earth launch	write articl orbit earth nasa	write articl orbit earth nasa	write articl orbit earth nasa	write articl orbit earth nasa	write articl orbit earth nasa	write articl orbit earth nasa	write articl earth orbit nasa	write articl earth space nasa	write articl earth peopl space	write articl earth peopl space	write articl peopl earth space	write articl peopl earth space
orbit write articl earth shuttl	write orbit articl earth launch	write orbit articl earth nasa	write articl orbit earth nasa	write articl orbit earth nasa	write articl orbit earth nasa	write articl orbit earth nasa	write articl orbit earth nasa	write articl orbit earth nasa	write articl earth nasa space	write articl earth space nasa	write articl earth peopl space	write articl peopl earth space	write articl peopl earth space	write articl peopl earth space
orbit write shuttl articl nasa	write orbit articl shuttl nasa	write orbit articl nasa shuttl	write articl orbit nasa launch	write articl orbit nasa earth	write articl orbit nasa earth	write articl orbit nasa earth	write articl orbit nasa earth	write articl nasa orbit earth	write articl nasa space earth	write articl space nasa earth	write articl space peopl earth	write articl peopl space earth	write articl peopl space earth	write articl peopl space earth
orbit write shuttl nasa space	write orbit articl shuttl nasa	write orbit articl shuttl nasa	write articl orbit nasa shuttl	write articl orbit nasa shuttl	write articl orbit nasa space	write articl orbit nasa space	write articl nasa orbit space	write articl nasa space orbit	write articl nasa space earth	write articl space nasa peopl	write articl space peopl nasa	write articl peopl space nasa	write articl peopl space nasa	write articl peopl space earth
shuttl orbit space nasa write	shuttl write orbit nasa space	write shuttl orbit nasa articl	write articl orbit nasa shuttl	write articl orbit nasa shuttl	write articl nasa orbit space	write articl nasa space orbit	write articl nasa space orbit	write articl nasa space orbit	write articl space nasa earth	write articl space nasa peopl	write articl space peopl nasa	write articl peopl space nasa	write articl peopl space nasa	write articl peopl space nation
shuttl space nasa orbit launch	shuttl space nasa orbit write	shuttl space nasa write orbit	write shuttl space nasa articl	write articl nasa space shuttl	write articl nasa space orbit	write articl nasa space orbit	write articl nasa space orbit	write articl space nasa orbit	write articl space nasa peopl	write articl space nasa peopl	write articl space peopl nasa	write articl peopl space nasa	write articl peopl space nasa	write articl peopl space nation
shuttl space nasa launch orbit	shuttl space nasa launch orbit	space shuttl nasa launch orbit	space shuttl nasa launch orbit	write space nasa shuttl articl	write space articl nasa shuttl	write articl space nasa orbit	write articl space nasa orbit	write articl space nasa orbit	write articl space nasa peopl	write articl space nasa peopl	write articl space nasa peopl	write articl space peopl nasa	write articl peopl space nasa	write articl peopl space nation
space shuttl nasa launch orbit	space shuttl nasa launch orbit	space shuttl nasa launch orbit	space shuttl nasa launch orbit	space nasa shuttl write launch	space write nasa articl shuttl	write space articl nasa orbit	write space articl nasa orbit	write articl space nasa orbit	write articl space nasa peopl	write articl space nasa peopl	write articl space nasa peopl	write articl space peopl nasa	write articl peopl space nasa	write articl peopl space nation
space shuttl nasa launch orbit	space shuttl nasa launch orbit	space shuttl nasa launch orbit	space shuttl nasa launch orbit	space nasa shuttl launch orbit	space nasa write shuttl orbit	space write nasa articl orbit	space write nasa articl orbit	write space articl nasa orbit	write space articl nasa orbit	write space articl nasa peopl	write articl space nasa peopl	write articl space peopl nasa	write articl space peopl nasa	write peopl articl space nation
space shuttl nasa launch peopl	space shuttl nasa launch orbit	space shuttl nasa launch orbit	space shuttl nasa launch orbit	space nasa shuttl launch orbit	space nasa orbit launch shuttl	space nasa orbit write launch	space write nasa orbit articl	space write nasa articl orbit	write space articl nasa orbit	write space articl nasa peopl	write space articl nasa peopl	write space articl nasa peopl	write space articl peopl nasa	write peopl space articl nation
space shuttl nasa launch peopl	space shuttl nasa launch orbit	space shuttl nasa launch orbit	space nasa shuttl launch orbit	space nasa orbit launch shuttl	space nasa orbit launch shuttl	space orbit nasa launch earth	space orbit nasa launch earth	space nasa orbit write articl	space write nasa articl orbit	space write nasa articl earth	write space articl nasa peopl	write space articl nasa peopl	write space articl peopl nasa	write space peopl articl nation
space shuttl nasa launch peopl	space shuttl nasa launch peopl	space shuttl nasa launch orbit	space nasa shuttl launch orbit	space orbit nasa launch shuttl	space orbit nasa launch earth	space orbit nasa launch earth	space orbit nasa earth launch	space orbit nasa earth launch	space orbit nasa earth write	space write nasa orbit earth	space write nasa articl earth	space write nasa articl peopl	write space nasa peopl articl	write space peopl nation articl
space shuttl nasa launch peopl	space shuttl nasa nation launch	space shuttl nasa launch orbit	space orbit launch nasa shuttl	space orbit launch nasa earth	space orbit launch earth nasa	space orbit earth launch nasa	space orbit earth launch nasa	space orbit earth launch nasa	space orbit earth nasa launch	space orbit earth nasa launch	space earth nasa orbit write	space write nasa earth nation	space write nasa nation peopl	space write nation peopl nasa
space nation shuttl peopl nasa	space nation shuttl nasa peopl	space nation orbit launch nasa	space orbit launch nation earth	space orbit launch earth nasa	space orbit earth launch nasa	space orbit earth launch nasa	space orbit earth launch nasa	space orbit earth launch nasa	space orbit earth launch nasa	space orbit earth launch nasa	space orbit earth nasa launch	space earth nasa orbit nation	space nasa earth nation write	space nation write nasa peopl
space nation peopl shuttl nasa	space nation peopl shuttl nasa	space nation orbit launch earth	space orbit nation earth launch	space orbit earth launch nation	space orbit earth launch nation	space orbit earth launch nation	space orbit earth launch nasa	space orbit earth launch nasa	space orbit earth launch nasa	space orbit earth launch nasa	space orbit earth launch nasa	space earth orbit nasa nation	space earth nation nasa orbit	space nation earth nasa govern

Figure 10: The most probable words formed in each of the 15 by 15 latent grid points by the Bernoulli latent trait model obtained in the experiments on text documents data.

like to thank Mário A. T. Figueiredo for providing his software.

A Quantities required for computing magnification factors in the reported experimental settings

The exact form of the matrices $\boldsymbol{F}$ is dependent on the specific noise-model being employed. These quantities require the computation of the first derivatives of the inverse link function $b()$. In this appendix we will provide the expressions for those members of the exponential model family which have been employed in the reported experimental settings.

A.1 Independent Gaussian noise model

The Gaussian model is the only member of the exponential family of distributions which is characterised by a quadratic cumulant function

$$B_t(\boldsymbol{y}) = \frac{1}{2} y_t^2. \tag{20}$$

Therefore, it has a linear inverse-link function and higher derivatives vanish.

$$b_{t'}(\boldsymbol{y}) = y_{t'}, \tag{21}$$

$$\frac{\partial b_{t'}(\boldsymbol{y})}{\partial y_t} = 0. \tag{22}$$

A.2 Independent Bernoulli noise model

In the case of the Bernoulli model, the cumulant function has the following form:

$$B_t(\boldsymbol{y}) = \log(1 + exp(y_t)). \tag{23}$$

The required derivatives are then computed as follows:

$$b_{t'}(\boldsymbol{y}) = \frac{\exp(y_{t'})}{1+\exp(y_{t'})}, \tag{24}$$

$$\frac{\partial b_{t'}(\boldsymbol{y})}{\partial y_t} = \begin{cases} 0 & t \neq t' \\ b_t(\boldsymbol{y})(1-b_t(\boldsymbol{y})) & t = t'. \end{cases} \tag{25}$$

It can be seen that for independent noise models, the Fisher information matrix $\boldsymbol{F}$ is diagonal.

A.3 Multinomial noise model

The multinomial distribution is identified by the following cumulant function:

$$B(\boldsymbol{y}) = \log\left(\sum_{t=1:T} exp(y_t)\right). \tag{26}$$

Accordingly, the derivatives are given by

$$b_{t'}(\boldsymbol{y}) = \frac{\exp(y_{t'})}{\sum_{t''=1}^{T}\exp(y_{t''})}, \tag{27}$$

$$\frac{\partial b_{t'}(\boldsymbol{y})}{\partial y_t} = \begin{cases} -b_{t'}(\boldsymbol{y})b_t(\boldsymbol{y}) & t \neq t' \\ b_{t'}(\boldsymbol{y}) - b_{t'}(\boldsymbol{y})b_t(\boldsymbol{y}) & t = t'. \end{cases} \tag{28}$$

References

[1] F. Aurenhammer. Voronoi diagrams - survey of a fundamental geometric data structure. *ACM Computing Surveys*, 3:345–405, 1991.

[2] O. Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory.* Wiley, Chichester, 1978.

[3] J. Bernardo and A. Smith. *Bayesian Theory.* Chichester, UK: J. Wiley & Sons, 1994.

[4] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: The Generative Topographic Mapping. *Neural Computation*, 10(1):215–235, 1998.

[5] C.M. Bishop, M. Svensén, and C. Williams. Magnification factors for the SOM and GTM algorithms. In *Proceedings 1997 Workshop on Self-Organizing Maps*, Helsinki, Finland, 1997.

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

[7] G. Celeux, F. Forbes S. Chrétien, and A. Mkhadri. A Component-Wise EM Algorithm for Mixtures. *J. Comput. Graphical Statistics*, 10:699–712, 2001.

[8] M. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24:381–396, 2002.

[9] P. Horton and K. Nakai. A probabilistic classification system for predicting the cellular localization sites of proteins. *Intelligent Systems in Molecular Biology*, (4):109–115, 1996.

[10] A. Kabán and M. Girolami. A combined latent class and trait model for the analysis and visualization of discrete data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):859–872, 2001.

[11] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1479, 1990.

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

[13] P. McCullagh and L. Nelder. *Generalized Linear Models.* Chapman and Hall, 1985.

[14] R. Miikkulainen. Script recognition with hierarchical feature maps. *Connection Science*, 2:83–101, 1990.

[15] S.J. Roberts, Ch. Holmes, and D. Denison. Minimum-entropy data partitioning using reversible jump Markov Chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:909–914, 2001.

[16] P. Tiňo and I. Nabney. Hierarchical GTM: Constructing localized nonlinear projection manifolds in a principled way. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24:639–656, 2002.

[17] C.S. Wallace and D.L. Dowe. Minimum message length and Kolmogorov complexity. *The computer Journal*, 42:270–283, 1999.

[18] R. Wolke and H. Schwetlick. Iterative reweighted least squares: Algorithms, convergence analysis, and numerical comparisons. *SIAM Journal on Scientific and Statistical Computing*, 9(5):907–921, 1988.