Semi-Supervised Construction of General Visualization Hierarchies

Peter Tiňo Yi Sun Ian Nabney
Aston University, Aston Triangle, Birmingham, B4 7ET
United Kingdom

Abstract *We have recently developed a principled approach to interactive non-linear hierarchical visualization [8] based on the Generative Topographic Mapping (GTM). Hierarchical plots are needed when a single visualization plot is not sufficient (e.g. when dealing with large quantities of data). In this paper we extend our system by giving 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 [8], whereas in the automatic mode an unsupervised minimum message length (MML)-driven construction of a mixture of GTMs is used. The latter is particularly useful when the 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. We illustrate our approach on a data set of 2300 18-dimensional points and mention extension of our system to accommodate discrete data types.*

Keywords: Latent trait model, minimum message length, hierarchical models, data visualization

I. Introduction

In general, a single two-dimensional projection of high-dimensional data, even if it is nonlinear, may not be sufficient to capture all of the interesting aspects of the data. Therefore, we have developed a principled approach to interactive construction of non-linear visualization hierarchies [8], the basic building block of which is the Generative Topographic Mapping (GTM) [1].

Since GTM is a generative probabilistic model, we were able to formulate training of the visualization hierarchy in a unified and principled framework of maximum likelihood parameter estimation using the expectation-maximization algorithm [8]. In this study, we present a further development in this direction, again taking advantage of the probabilistic character of GTM. When the user initializes child plots of the current plot they can do so in either interactive or automatic modes. In the interactive mode user decides what subsets of the data are interesting enough to be visualized in a greater detail at lower level plots [8]. In the automatic mode, the number and position of children GTMs are determined in an unsupervised manner using the minimum message length (MML) methodology. This is important, e.g. when dealing with large quantities of data that make visualization plots at higher levels so complicated that the interactive mode cannot be used.

Using a data partitioning technique (e.g. [7]) for segmenting the data set, followed by constructing visualization plots in the individual compartments is not a good alternative – there is no direct connection between the criterion for choosing the quantization regions and making local low-dimensional projections. Using GTM, however, such a connection can be established. GTM is 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 [4]. Hence, given a parent GTM, the number and position of its children is based on the modeling properties of the children themselves, and not some outside ad-hoc criterion.

II. Generative Topographic Mapping

The Generative Topographic Mapping (GTM) is a latent space model, i.e. it models probability distributions in the (observable) data space by means of latent (hidden) variables. In GTM, the visualization space is identified with the latent space (usually a bounded subset of a two-dimensional Euclidean space).

In general, the L-dimensional latent space $\mathcal{H} \subset \Re^L$, in which latent points $\mathbf{x} = (x_1, ..., x_L)^T$ live, is covered by a grid of K latent space centers $\mathbf{x}_i \in \mathcal{H}$, $i = 1, 2, ..., K$. Let the data space $\mathcal{D}$ be the D-dimensional Euclidean space $\Re^D$. We define a non-linear transformation $f : \mathcal{H} \to \mathcal{D}$ as a radial basis function network by covering the latent space with a set of $M - 1$ fixed non-linear basis functions $\phi_j : \mathcal{H} \to \Re$, $j = 1, 2, ..., M - 1$. As usual in the GTM literature, we work with spherical Gaussian functions of the same width σ, positioned on a regular grid. The bias term is included via an additional constant basis function $\phi_M(\cdot) = 1$. Latent space points $\mathbf{x} \in \mathcal{H}$, are mapped into the data space via

$$f(\mathbf{x}) = \mathbf{W}\ \phi(\mathbf{x}), \qquad (1)$$

where $\mathbf{W}$ is a $D \times M$ matrix of weight parameters and $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), ..., \phi_M(\mathbf{x}))^T$.

GTM forms, in the data space, a constrained mixture of K spherical Gaussians $P(\mathbf{t}|\mathbf{x}_i)$ with inverse variance β, centered at the f-images, $f(\mathbf{x}_i)$, of the latent space centers $\mathbf{x}_i \in \mathcal{H}$,

$$P(\mathbf{t}|\ \mathbf{x}_i, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|f(\mathbf{x}_i) - \mathbf{t}\|^2\right\}. \qquad (2)$$

Imposing a uniform prior over $\mathbf{x}_i$, the density model in $\mathcal{D}$ provided by the GTM is

$$P(\mathbf{t}) = 1/K \sum_{i=1}^{K} P(\mathbf{t}|\mathbf{x}_i). \qquad (3)$$

Given a data set $\zeta = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_N\}$ of independently generated points in $\mathcal{D}$, the adjustable parameters $\mathbf{W}$ and β of the model are determined by maximum likelihood using an expectation-maximization (EM) algorithm [1].

For the purpose of data visualization, we use Bayes' theorem to "invert" the transformation f. The posterior distribution on $\mathcal{H}$, given a data point $\mathbf{t}_n \in \mathcal{D}$, is a sum of delta functions centered at centers $\mathbf{x}_i$, with coefficients equal to the posterior probability R_{in} that the i-th Gaussian, corresponding to the latent space center $\mathbf{x}_i$, generated $\mathbf{t}_n$ [1]. The latent space representation of the point $\mathbf{t}_n$, i.e. the *projection of* $\mathbf{t}_n$, is then the mean $\sum_{i-1}^{K} R_{in}\ \mathbf{x}_i$ of the posterior distribution on $\mathcal{H}$.

Following [8], we refer to the f-image of the latent space, $f(\mathcal{H})$, as the projection manifold of the GTM.

A. Hierarchical GTM

In [8], we extended GTM to hierarchies of GTMs, organized in hierarchical trees and interactively constructed in a top down fashion, starting from a single *Root* plot. Let us first concentrate on simple mixtures of GTMs, i.e. on hierarchical trees of depth 1, where the mixture components are children of the *Root*.

Consider a mixture of A GTMs. Each mixture component $P(\mathbf{t}|a)$ has an associated (non-negative) mixture coefficient π_a satisfying $\sum_{a=1}^{A} \pi_a = 1$. The mixture distribution is then given by

$$P(\mathbf{t}) = \sum_{a=1}^{A} \pi_a\ P(\mathbf{t}|a). \qquad (4)$$

The mixture is trained by an EM algorithm. In the **E-step**, given each data point $\mathbf{t}_n \in \mathcal{D}$, we compute the model responsibilities corresponding to the competition among the mixture components

$$P(a|\mathbf{t}_n) = \frac{\pi_a\ P(\mathbf{t}_n|a)}{\sum_{b=1}^{A} \pi_b\ P(\mathbf{t}_n|b)}. \qquad (5)$$

Responsibilities $R_{i,n}^{(a)}$ of the latent space centers $\mathbf{x}_i$, $i = 1, 2, ..., K$, corresponding to the competition among the latent space centers within each GTM a, are calculated as in standard GTM (see [1]).

The free parameters are estimated in the **M-step** using the posterior over hidden variables computed in the E-step. The mixture coefficients are determined by

$$\pi_a = \frac{\sum_{n=1}^{N} P(a|\mathbf{t}_n)}{N}. \tag{6}$$

Weight matrices $\mathbf{W}^{(a)}$ are calculated by solving

$$(\mathbf{\Phi}^T\ \mathbf{B}^{(a)}\ \mathbf{\Phi})\ (\mathbf{W}^{(a)})^T = \mathbf{\Phi}^T\ \mathbf{R}^{(a)}\ \mathbf{T}, \tag{7}$$

where $\mathbf{\Phi}$ is a $K \times M$ matrix with elements $(\mathbf{\Phi})_{ij} = \phi_j(\mathbf{x}_i)$, $\mathbf{T}$ is a $N \times D$ matrix storing the data points $\mathbf{t}_1, ..., \mathbf{t}_N$ as rows, $\mathbf{R}^{(a)}$ is a $K \times N$ matrix containing, for each latent space center $\mathbf{x}_i$, and each data point $\mathbf{t}_n$, *scaled* responsibilities $(\mathbf{R}^{(a)})_{in} = P(a|\mathbf{t}_n) R_{i,n}^{(a)}$, and $\mathbf{B}^{(a)}$ is a $K \times K$ diagonal matrix with diagonal elements corresponding to responsibilities of latent space centers for the whole data sample, $(\mathbf{B})_{ii} = \sum_{n=1}^{N} (\mathbf{R}^{(a)})_{in}$.

The inverse variances are re-estimated using

$$\begin{aligned} \frac{1}{\beta^{(a)}} &= (\sum_{n=1}^{N} P(a|\ \mathbf{t}_n) \sum_{i=1}^{K} R_{i,n}^{(a)} \\ &\quad \|\mathbf{W}^{(a)}\ \phi(\mathbf{x}_i) - \mathbf{t}_n\|^2) \\ &\quad /(D \sum_{n=1}^{N} P(a|\ \mathbf{t}_n)). \end{aligned} \tag{8}$$

Training equations for a full hierarchy of GTMs are more involved, but the only real complication is that for nodes on levels > 2, we also have to consider model responsibilities of the parent nodes, and these are recursively propagated as we incrementally build the hierarchy. We refer the interested reader to [8].

III. MML formulation for unsupervised learning of mixtures and hierarchies of GTMs

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 [9]. Given that the data is modeled by a parametric probabilistic model $P(\zeta|\theta)$, the message consists of two parts – one specifying the model parameters, the other specifying the data given the model:

$$\mathbf{Length}(\theta, \zeta) = \mathbf{Length}(\theta) + \mathbf{Length}(\zeta|\theta). \tag{9}$$

By Shannon's arguments, the first term is no less than $\lceil -\log p(\theta) \rceil$ (based on a prior $P(\theta)$ over the model space), and the second one is no less than $\lceil -\log(P(\zeta|\theta)) \rceil$.

Recently, Figueiredo and Jain [4] extended the MML framework to unsupervised learning of mixture models. The particular form of MML criterion adopted in [4] is of the form $\hat{\theta} = \text{argmin}_\theta\ \mathcal{L}(\theta, \zeta)$, where

$$\begin{aligned} \mathcal{L}(\theta, \zeta) &= \frac{Q}{2} \sum_{a:\pi_a > 0} \log\left(\frac{N\pi_a}{12}\right) + \frac{A_+}{2} \log \frac{N}{12} \\ &+ \frac{A_+(Q+1)}{2} - \log P(\zeta|\theta), \end{aligned} \tag{10}$$

where Q is the number of free parameters of each mixture component. We only code the parameters of mixture components a with positive prior π_a. The number of such components is denoted by A_+. For details concerning derivation of (10), we refer the reader to [4]. We briefly mention that the result follows from adopting a specific form of MML, replacing Fisher information matrix of the mixture by the complete-data Fisher matrix (including binary mixture component indicators), and imposing non-informative Jeffreys' prior on both the vector of mixing coefficients $\{\pi_a\}$ and the parameters $\theta_\mathbf{a}$ of individual mixture components (we assume that these priors are independent).

Minimization of (10), with A_+ fixed, leads to the following re-estimation of mixture coefficients [4]: for $a = 1, 2, \ldots, A_+$,

$$\hat{\pi}_a(t+1) = \frac{\max\left\{0,\ -\frac{Q}{2} + \sum_{n=1}^{N} P(a|\mathbf{t}_n)\right\}}{\sum_{b=1}^{A_+} \max\left\{0,\ -\frac{Q}{2} + \sum_{n=1}^{N} P(b|\mathbf{t}_n)\right\}}, \tag{11}$$

where component responsibilities $P(a|\mathbf{t}_n)$ are computed using (5). Free parameters $\theta_\mathbf{a} =$

$(\mathbf{W}^{(\mathbf{a})}, \beta^{(\mathbf{a})})$ of the individual GTMs are fitted to the data ζ using the EM algorithm outlined in section II-A. Note that GTMs corresponding to zero $\hat{\pi}_a$ become irrelevant and so (11) effectively performs component annihilation [4].

To start the training process, we choose the maximum number of components A_{max} we are willing to consider. Then, we initiate the component GTMs around randomly selected points $\mathbf{c}_1, ..., \mathbf{c}_{A_{max}}$, from ζ. These "centers" induce a Voronoi tessellation $\{V_a\}$ in the data space. Following [8], each GTM $a \in \{1, ..., A_{max}\}$ is initialized to approximate the local eigenspace $E_a^{(2)}$ spanned by the first 2 eigenvectors of the local covariance matrix of points from ζ belonging to the Voronoi compartment V_a.

As in [4], we adopt the component-wise EM (CEM) [3], i.e. rather than simultaneously updating all the GTMs; we first update the parameters $\theta_\mathbf{1}$ of the first GTM (7–8), while parameters of the remaining GTMs are fixed, then we recompute the model responsibilities $\{P(a|\mathbf{t}_n)\}_{a=1}^{A}$ (5) for the whole mixture. After this, we move to the second component, update in the same manner $\theta_\mathbf{2}$, and recompute $\{P(a|\mathbf{t}_n)\}_{a=1}^{A}$, etc., looping through the mixture components. If one of the component GTMs dies ($\hat{\pi}_a = 0$), 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 GTMs (down to $A_+ = 1$). This is done by iteratively killing off the weakest GTM (with the smallest $\hat{\pi}_a$) and re-running CEM until convergence. Finally, the winning mixture of GTMs is the one that leads to the shortest message length $\mathcal{L}(\theta, \zeta)$ (10).

Empirically, we observed that "strong" GTMs that survived for longer time periods tended to be overtrained. One does not encounter such problems when dealing with simple mixtures of Gaussians, as was the case in [4]. However, GTM is a constrained mixture of Gaussians and the low-dimensional manifold containing centers of Gaussian noise models (projection manifold [8]) tended to form complicated folds. Simple introduction of a stronger regularization term [1] was not of much help, since then the individual GTMs were rather stiff and did not realize the full potential of having a mixture of nonlinear projections. Therefore, we adopted the following technique: after a component GTM has been eliminated and before starting a new competition of the remaining GTMs for the data explained by it, we re-initialize the remaining GTMs so that they remain in their respective positions, but have a "fresh start" with less complicated projection manifolds. For each GTM we collect the data points for which that GTM has responsibility (eq. (5)) higher than a threshold $\Delta = 0.85$. We then initialize and train individual GTMs for 1 epoch in the traditional way [1], each on the corresponding model-restricted set, as if they were not members of a mixture. After this re-initialization step, the CEM algorithm is applied to the mixture on the whole data set.

The proposed system for constructing hierarchies of non-linear visualization plots is similar to the one described in [8]. 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 he wants the children to be constructed in the interactive or unsupervised way. In the unsupervised case, we use the MML technique to decide the "appropriate" number and approximate position of children GTMs[1] and view the resulting local mixture as an initialization for the full EM algorithm for training hierarchies of GTMs [8].

[1] We collect data points from ζ for which the parent GTM has responsibility higher than the threshold Δ. We then run MML-based learning of mixtures of GTMs on this reduced data set.

IV. Illustrative example

As an example we visualize in figure 1 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*. The parameters of GTMs were as follows: latent space $[-1, 1]^2$, $K = 15 \times 15$ latent space centers, $M = 4 \times 4 + 1$ RBF spherical Gaussian kernels of width 1, "weight-decay" regularization coefficient 0.1 [1]. For a complete information on presentation of the visualization hierarchy, we refer the reader to [8].

We organize the plots of the hierarchy in a hierarchical tree. In non-leaf plots, provided the child models were initialized in the interactive mode, we show the latent space points $\mathbf{c}_i$ that were chosen to be the "centers" of the regions of interest to be modeled in greater detail at lower levels. These are shown as circles labeled by numbers. The numbers determine the order of the corresponding child GTM subplots (left-to-right).

We adopt the strategy, suggested in [2], of plotting all the data points on every plot, but modifying the intensity in proportion to the responsibility (posterior model probability) $P(\mathcal{M}|\ \mathbf{t}_n)$ 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.

The user can visualize the regions captured by a particular child GTM $\mathcal{M}$, by modifying the plot of its parent, $Parent(\mathcal{M})$, so that instead of the parent responsibilities, $P(Parent(\mathcal{M})|\ \mathbf{t}_n)$, the responsibilities of the model $\mathcal{M}$, $P(\mathcal{M}|\ \mathbf{t}_n)$, are used. Alternatively, the user can modulate with responsibilities $P(\mathcal{M}|\ \mathbf{t}_n)$ all the ancestor plots up to *Root*. As shown in [8] , such a modulation of ancestor plots is an important tool to help the user relate child plots to their parents.

The *Root* plot contains dense clusters of overlapping projections. Six plots at the second level were constructed using the unsupervised MML technique ($A_{max} = 10$). Note that the classes are already fairly well-separated. We further detailed the second plot in the interactive mode, by selecting centers (shown as circles) of 2 regions of interest. Since the fifth plot contains a region of overlapping projections, we use again the MML technique for constructing its children plots. The resulting children plots are readable enough to be further detailed in the interactive mode. We stress that all useful tools for understanding the visualization hierarchy described in [8], such as children-modulated parent plots, magnification factor and directional curvature plots can also be used in the proposed system.

V. Discrete data types

In another line of development, we have extended the basic hierarchical GTM [8] to deal with noise models from the general exponential family of distributions [6]. This is important for visualizing other than continuous data types, e.g. binary or count data, where Bernoulli or multinomial noise models can be used.

We briefly mention, that by employing MML technique into such generalized hierarchical visualization system, we can perform e.g. semi-supervised hierarchical document mining. The documents are represented as high dimensional discrete vectors through the key-word technique. The visualization hierarchy is now composed of so-called *latent trait models* [5], which are basically GTMs endowed with noise models from the exponential family of distributions (in our example Bernoulli/multinomial). Other tools aimed at improving our understanding of the plots, like listing the most probable dictionary (key) words for each latent space center $\mathbf{x_i}$ [5], are also incorporated in the system.

VI. Conclusion

We have described a principled approach to semi-supervised data visualization. The proposed system gives the user a choice of initializing the child plots of the current plot in either *interactive*, or *automatic* mode. It is par-

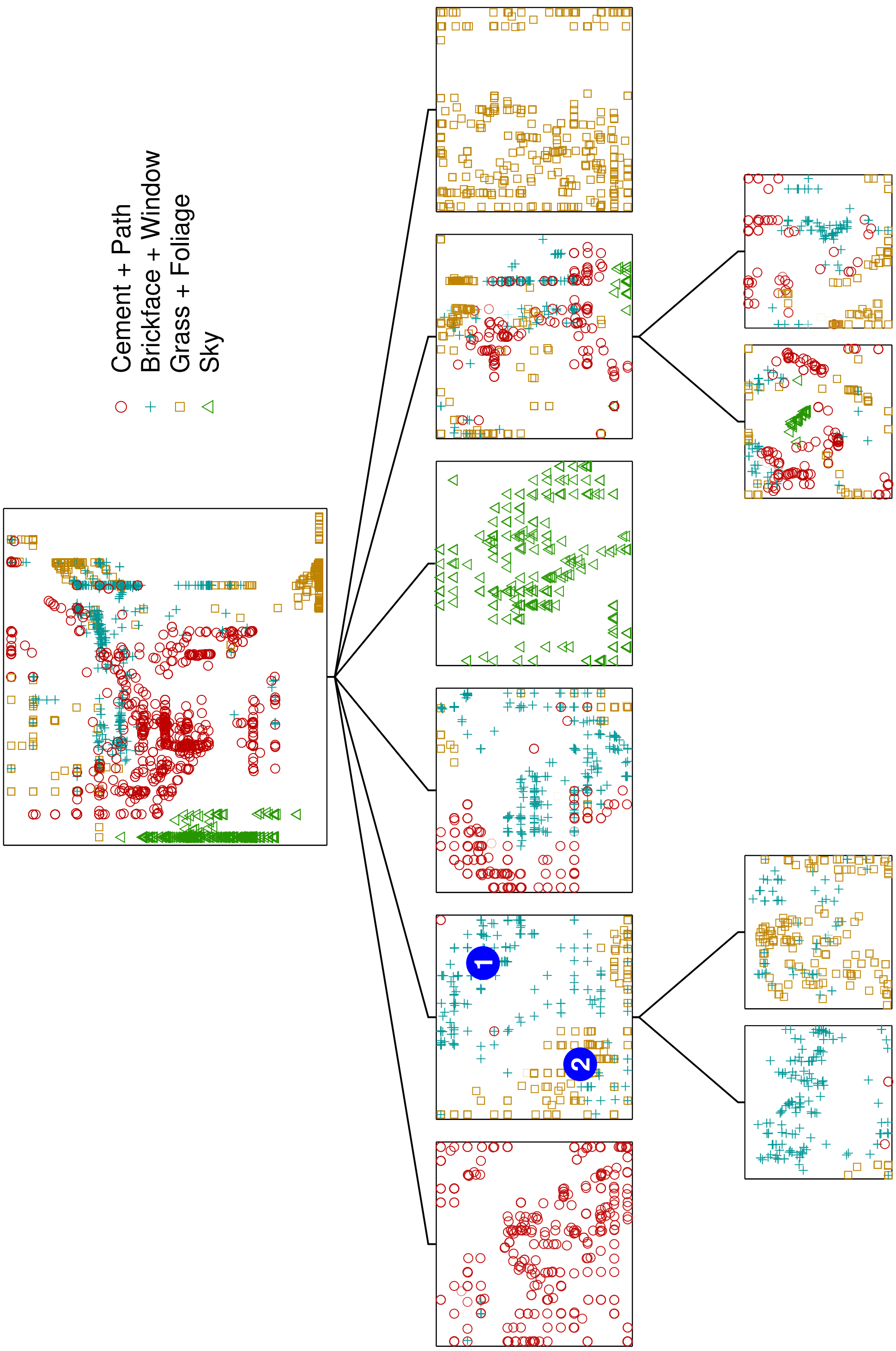

Fig. 1. Hierarchical visualization of the image segmentation data constructed in a semi-interactive way.

ticularly useful when user has no idea how to choose the area of interest due to highly overlapping dense data projections.

Acknowledgments

This research has been funded by BBSRC grant 92/BIO12093 and Pfizer Central Research. The experiments were carried out with the NETLAB neural network toolbox, available from
http://www.ncrg.aston.ac.uk/netlab.
Yi Sun would like to thank Mário A. T. Figueiredo for providing his codes.

REFERENCES

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

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

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

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

[5] Kabán A., Girolami, M.: A combined latent class and trait model for the analysis and visualization of discrete data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**: 859–872, 2001.

[6] Kabán A., Tiňo P., Girolami, M.: General Framework for a Principled Hierarchical Visualization of High-Dimensional Data. *International Conference on Intelligent Data Engineering and Automated Learning - IDEAL 2002*, in print.

[7] Roberts S.J., Holmes Ch., Denison D.: Minimum-Entropy Data Partitioning Using Reversible Jump Markov Chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**: 909–914, 2001.

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

[9] C.S. Wallace C.S., Dowe D.L.: Minimum Message Length and Kolmogorov Complexity. *The computer Journal*, **42**: 270–283, 1999.