Mach Learn (2007) 68: 63–95
DOI 10.1007/s10994-007-5008-8

Predictive Modelling of Heterogeneous Sequence Collections by Topographic Ordering of Histories

Ata Kabán

Received: 30 March 2006 / Accepted: 25 February 2007 /
Revised: 20 January 2007 / Published online: 15 May 2007

Abstract We propose a model-based approach to the twofold problem of prediction and exploratory analysis of heterogeneous symbolic sequence collections. Our model is based on seeking low entropy local representations joined together with a smooth nonlinear mixing process. Low entropy components are desirable, as they tend to be both more interpretable and more predictable. The nonlinear mixing in turn acts as a regulariser, and in addition, it creates a topographic ordering of the sequence histories, which is useful for exploratory purposes. The combination of these two modelling elements is performed through the generative probabilistic formalism, which ensures a flexible and technically sound predictive modelling framework. Unlike previous generative topographic modelling approaches for discrete data, the estimation algorithm associated with our model is designed to scale to large data sets by exploiting data sparseness. In addition, local convergence is guaranteed without the need for tuning optimisation parameters or making approximations to the non-Gaussian likelihood. These characteristics make it the first generative topographic model for discrete symbolic data with large scale real-world applicability. We analyse and discuss the relationship of our approach with a number of models and methods. We empirically demonstrate robustness against varying sample sizes, leading to significant improvements in terms of predictive performance over the state of the art. Finally we detail an application to the prediction and exploratory analysis of a large real-world web navigation sequence collection.

Keywords Probabilistic modelling · Generative topographic mapping · Generalisation across multiple sequences · Data prediction · Data explanation · Visualisation

Editor: Zoubin Ghahramani.

A. Kabán (✉)
School of Computer Science Edgbaston, The University of Birmingham, Birmingham B15 2TT, UK
e-mail: A.Kaban@cs.bham.ac.uk

1 Introduction

Understanding high dimensional data through low dimensional representations is of practical interest in any field where multivariate data analysis is required. The main purpose is to retain the informative structural patterns and relationships from the data in an automated manner. The study of approaches to this problem has a long history, ranging over several methodological frameworks, including neural networks (Kohonen 1999; Kaski et al. 1998), statistical learning (Hastie et al. 2001), linear algebraic or spectral methods (Bengio et al. 2004) and probabilistic density modelling (Bishop et al. 1998a), to name just a few. The solution is often given by some suitable (non-linear) transformation of the multivariate data set. Smooth transforms preserve the local topological relationships and the representation created by methods that utilise such transforms are referred to as topological orderings.

While it is well-known that probabilistic generative density-based approaches provide a flexible and technically sound framework of *predictive* model building—e.g. the Generative Topographic Mapping (GTM) (Bishop et al. 1998a) has been a powerful tool of principled data visualisation and prediction (Carreira-Perpiñán & Renals 1998)—much of the recent advances have been concentrating on non-probabilistic formulations (Bengio et al. 2004; Iwata et al. 2005), due to their appealing computational advantages. However, the lack of a clear density formulation deprives such methods from the predictive abilities and the flexibility of probabilistic model formulations (Roweis et al. 2002; Bishop et al. 1998b). In consequence, they cannot straightforwardly serve predictive purposes and (without further tweaking) their functionality is essentially limited to data exploratory tasks. This is a serious limitation for two reasons: Obviously, one reason is efficiency, since one would need to use different methods for different tasks. Secondly, even though visual data analysis is potentially powerful, it holds the risk of being very subjective. Indeed, there is no objective quality measure for visualisations alone and it is not straightforward to reason about data outside the training set (Bengio et al. 2004). In consequence it is not straightforward to assess the extent to which the representation patterns extracted from the data—and used in our efforts to understand the data—are actually significant and generalise beyond the limited amount of evidence gathered in a certain fixed set of data.

Hence, in our view, when both data prediction and data explanation are required, these two functions should be conceptually closely interconnected and should ideally be based on a common representation model. Examples where the automation of both explanatory and predictive tasks are needed may be found from text and user modelling (Blei et al. 2003; Hofmann 2000; Cadez et al. 2003) to various scientific data mining problems (Ramakrishnan and Grama 2001). To give a concrete example, a site administrator would not only want to visualise and explore a pool of navigation sequences produced (Cadez et al. 2003) but also to make accurate predictions of the individual users' preferences. Conversely, rather than black-box prediction machines, one would often like to receive additional explanatory information. This calls for multi-objective solution designs, and the generative probabilistic formalism is well suited for this purpose.

However, to date, the computational complexity of existing generative model-based non-linear data compression approaches (Bishop et al. 1998b; Roweis et al. 2002; Kabán and Girolami 2001) renders them impractical to use with large amounts of high dimensional data. This weakness is more pronounced in the case of non-Gaussian noise models (Bishop et al. 1998b; Kabán and Girolami 2001; Tiňo et al. 2004) such as those that are statistically appropriate for discrete or symbolic data sets. The state of the art models of multiple symbolic sequence collections that scale to realistic data sets are currently linear (Cadez et al.

2003; Girolami and Kabán 2005). Their flexibility is therefore limited and they lack the capability to model any correlations or complex dependencies between their representational components.

In this paper we develop a novel model of multiple symbolic sequences that addresses the above concerns and allows us to build nonlinear models of large and sparse real-world sequence collections that are suitable for both predictive and exploratory analysis. A short preliminary version of our approach appears in (Kabán 2005). The combination of our modelling elements is realised through the generative probabilistic formalism and this will be presented in Sect. 2. Further alternative views and interpretations will be highlighted in Sect. 3, where it will be shown that both the predictive and explanatory functionalities achieved may be viewed as seeking low entropy representation components of the data, corroborated with a smooth nonlinear mixing model. Indeed, low entropy components are of interest, as each of these tend to be both interpretable and predictable. The nonlinear mixing model in turn acts as a regulariser, and in addition, it produces a topographic ordering of the sequence histories, useful for exploratory purposes. Various connections may be followed with a number of previous approaches, and these will be elaborated upon in Sect. 4. While the principle of topographic ordering is inherited from both density-based (Bishop et al. 1998a) and channel-noise based (Hofmann 2000) models, our approach brings in a number of novel and unified advantageous features over previous topographic models: (1) The parameters of our model are interpretable probabilistic quantities and as such, they offer new explanatory information about the data and the predictive process. (2) Secondly, in contrast with previous generative topographic models for discrete data, neither approximations to the non-Gaussian likelihood nor tuning of optimisation parameters is required, and the estimation algorithm associated with our model is designed to scale to large data sets by exploiting data sparseness. These characteristics confer it large scale real-world applicability, and enables leveraging sound generative topographic modelling principles to large realistic discrete data modelling and prediction problems for the first time. Section 5 empirically demonstrates the robustness of our approach against varying sample sizes. Due to this, significant improvements are obtained in terms of predictive performance over the state of the art. We then detail the application of our approach to the prediction and exploratory analysis of a large real-world web navigation sequence collection in Sect. 6, and conclude in the last section.

2 A Predictive Topographic Model for Sparse Sequence Collections

Consider a set of independent symbolic sequences over a common state space and let us denote the n-th instance of the collection by S_n, where $n = 1, \ldots, N$ and N is the number of sequences. Examples include sequences of words within text documents, sequences of activity logs in traces left by users while interacting with an electronic environment, sequences of events in a musical piece, etc. The simplest model to represent such data is the random sequence model, employed in the popular 'bag of words' representation of text documents. If the temporal order contains important information, then Markov models may be of interest. To keep the notation simple while still allowing some generality, we will adopt the first order Markov assumption in this paper. Formally, a possible adaptation to either the 0-th order case (bag of words) or to the higher order Markov case can be accomplished straightforwardly by removing or adding indices. We can also make a direct connection to popular text document representation models, such as PLSA or LDA (Blei et al. 2003), essentially

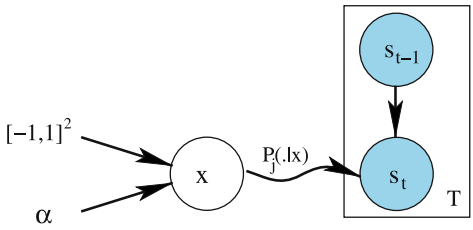

Fig. 1 Graphical representation of the generative process. Nodes represent random variables, arrows denote conditional dependencies and plates are repetitive structures. A symbolic sequence is generated by a smooth nonlinear transform of a continuous latent point $\boldsymbol{x}$

by considering transitions (bigrams) being counted in sequences just as words are counted in text documents. Throughout this work, we assume that all states are observable.

In order to devise a both predictive and explanatory model for representing the collection of data sequences we begin with defining a latent space $\boldsymbol{x} \in [-1, 1]^L$, $L = 2$—chosen to be a bounded Euclidean space that will be useful for visualising the obtained data representation, as in (Bishop et al. 1998b; Kabán and Girolami 2001). In a document modelling context, this may be thought of as some conceptual or topical space. Further, assuming a prior latent density $p(\boldsymbol{x})$ (which may be uniform but not necessarily) we define a generative model in the following way. The corresponding graphical representation is shown in Fig. 1. To generate a sequence S_n,

- generate a point $\boldsymbol{x}$ in the latent space from the prior density $p(\boldsymbol{x})$
- project this point to a probability distribution over K fixed Gaussian kernels, by computing

$$\phi_k(\boldsymbol{x}) = \frac{\exp(-\frac{1}{2\sigma^2}|\boldsymbol{y}_k - \boldsymbol{x}|^2)}{\sum_{k'} \exp(-\frac{1}{2\sigma^2}|\boldsymbol{y}_{k'} - \boldsymbol{x}|^2)}$$

  for all $k = 1 : K$, where $\{\boldsymbol{y}_k\}$ and K are fixed a priori. These are smooth nonlinear functions of $\boldsymbol{x}$ that may also be intuitively thought of as a neighbourhood probability distribution associated with $\boldsymbol{x}$.
- apply a stochastic translation to the above distribution, using the stochastic parameters of the model, $P_j(i|k)$, i.e. the probability of transitioning from j to i, to obtain $P_j(i|\boldsymbol{x}) \equiv \sum_k P_j(i|k)\phi_k(\boldsymbol{x})$, for all $i, j = 1, \ldots, |S|$ (where $|S|$ denotes the size of the state space).
- generate a sequence from the resulted probability transition model $\{P_j(i|\boldsymbol{x})\}_{i,j=1,\ldots,|S|}$. In a document modelling context, these are probabilities of words or terms conditioned on a particular topic.

For the sake of consistent notations later, the state space S will contain, besides the actual symbol dictionary, an additional 'start' symbol, so that the first observed symbol of any sequence is generated as a transition from the 'start' symbol. The width parameter above, σ, will be fixed to twice the maximum distance between neighboring centres $\boldsymbol{y}_k$, so that the Gaussian kernels span a uniform density.

Some analogy with the GTM (Bishop et al. 1998a), and in particular the multinomial latent trait model (LTM) (Kabán and Girolami 2001; Bishop et al. 1998b) is evident, and this will be discussed in more detail later. The main formal difference is, though, that our nonlinear 'basis' functions $\phi_k(.)$ are designed to perform a transformation from the Euclidean latent space into a $(K-1)$-dimensional simplex, rather than to another Euclidean space. This will turn out to have important consequences in terms of both parameter interpretability and scalability of the algorithm.

2.1 Parameter Estimation and Inference

The probability of a sequence $S_n = (s_{1n}, s_{2n}, \ldots, s_{tn}, \ldots, s_{T_n,n})$ under the generative model defined above is the following:

$$P(S_n) = \int d\boldsymbol{x}\, p(\boldsymbol{x}) \prod_{t=1}^{T_n} \sum_k P(s_{t-1,n} \to s_{tn}|k)\phi_k(\boldsymbol{x}) \tag{1}$$

$$= \int d\boldsymbol{x}\, p(\boldsymbol{x}) \prod_{i=1}^{|S|} \prod_{j=1}^{|S|} \left\{ \sum_k P_j(i|k)\phi_k(\boldsymbol{x}) \right\}^{N_{ij}^n} \tag{2}$$

where N_{ij}^n is the frequency of occurrence of the subsequence $\{j, i\}$ in sequence S_n. (At $t = 0$, we always have the 'start' symbol.)

To further place this model in context, we note that if $\phi_k(.)$ were identity functions and $p(\boldsymbol{x})$ was a Dirichlet, then (2) would reduce to a generative aspect model (Blei et al. 2003; Buntine 2002; Girolami and Kabán 2005) (or simplicial mixture) which has been a quite popular modelling scheme recently. However, while this connection is insightful, we will see that the mentioned differences have a major impact on the predictive performance. Not only does the distribution of $\boldsymbol{\phi}(\boldsymbol{x})$ offer more flexibility than a Dirichlet, but in addition, due to the nonlinearity, our model is able to capture correlations in the latent space and this results in robustness against finite sample sizes. This issue will be extensively demonstrated in the later sections.

Now we turn to identifying the model, i.e. to infer $\boldsymbol{x}$ and estimate the parameters $P_j(i|k)$ from the data. For tractability reasons, it is convenient to discretise the latent space into a regular grid of M points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$, in which case the latent prior becomes a multinomial over the sample points, or a mixture of Dirac delta functions $p(\boldsymbol{x}) = \sum_m P(\boldsymbol{x}_m)\delta(\boldsymbol{x} - \boldsymbol{x}_m)$, where typically we will work with $M > K$ samples. As in GTM (Bishop et al. 1998a), we may choose to fix the mixing coefficients $P(\boldsymbol{x}_m) = 1/M$, if we have reasons to believe that a uniform latent density describes the data well. Alternatively, if we believe that there may be regions of uneven data density, e.g. distinct clusters in the data, then we may estimate the mixing coefficients from the data. The data consists of a set of sequences $S_1, S_2, \ldots, S_N$.

Performing an approximate integration by summing over the latent space samples, the data likelihood is now the following.

$$P(S_n) = \sum_m P(\boldsymbol{x}_m) \prod_i \prod_j \left\{ \sum_k P_j(i|k)\phi_k(\boldsymbol{x}_m) \right\}^{N_{ij}^n}. \tag{3}$$

Using the latent samples, the complete data likelihood follows.

$$\mathcal{L}^C = P(S_n, \boldsymbol{x}) = \prod_m P(\boldsymbol{x}_m)^{\delta(\boldsymbol{x}-\boldsymbol{x}_m)} \prod_i \prod_j \left\{ \sum_k P_j(i|k)\phi_k(\boldsymbol{x}) \right\}^{N_{ij}^n \delta(\boldsymbol{x}-\boldsymbol{x}_m)}. \tag{4}$$

Adopting the EM methodology (McLachlan and Krishnan 1997), we maximise the expectation of the log of (4), taken w.r.t. the posteriors of the latent space samples $r_{mn} \equiv P(\boldsymbol{x}_m|S_n)$, as a function of the model parameters. This is the following.

$$Q = E[\log \mathcal{L}^C] = \sum_n \sum_m r_{mn} \left\{ \sum_{i,j} N_{ij}^n \log \sum_k P_j(i|k)\phi_k(\boldsymbol{x}_m) + \log p(\boldsymbol{x}_m) \right\}. \tag{5}$$

The posterior probabilities r_{mn} are computed in the E-step, using the old parameters $P_j(i|k)$ and $P(\boldsymbol{x}_m)$:

$$r_{mn} = \frac{\prod_i \prod_j \{\sum_k P_j(i|k)\phi_k(\boldsymbol{x}_m)\}^{N_{ij}^n} P(\boldsymbol{x}_m)}{\sum_{m'} \prod_i \prod_j \{\sum_k P_j(i|k)\phi_k(\boldsymbol{x}_{m'})\}^{N_{ij}^n} P(\boldsymbol{x}_{m'})} \tag{6}$$

followed by re-estimating $P_j(i|k)$ (and optionally $P(\boldsymbol{x}_m)$), by maximising (5) in the M-step, subject to the required constraints $\sum_i P_j(i|k) = 1,\ \forall j = 1, \ldots, |S|,\ \forall k = 1, \ldots, K$ and $\sum_m P(\boldsymbol{x}_m) = 1$, while holding r_{mn} fixed. The Lagrangian to be maximised in the M-step is thus the following.

$$\tilde{Q} = E[\log \mathcal{L}^C] - \sum_j \sum_k u_{jk} \left(\sum_i P_j(i|k) - 1 \right) - v \left(\sum_m P(\boldsymbol{x}_m) - 1 \right) \tag{7}$$

where u_{jk} and v are Lagrange multipliers. Computing the stationary equations for the parameter $P_j(i|k)$, we obtain

$$\frac{\delta \tilde{Q}}{\delta P_j(i|k)} = \sum_n \sum_m r_{mn} N_{ij}^n \frac{\phi_k(\boldsymbol{x}_m)}{\sum_{k'} P_j(i|k')\phi_{k'}(\boldsymbol{x}_m)} - u_{jk} = 0. \tag{8}$$

Multiplying both sides by $P_j(i|k)$ yields

$$P_j(i|k) = \frac{1}{u_{jk}} P_j(i|k) \sum_n \sum_m \frac{r_{mn} N_{ij}^n}{\sum_{k'} P_j(i|k')\phi_{k'}(\boldsymbol{x}_m)} \phi_k(\boldsymbol{x}_m) \tag{9}$$

where u_{jk} is non-zero, and by summing both sides over i, we get

$$u_{jk} = \sum_i P_j(i|k) \sum_n \sum_m \frac{r_{mn} N_{ij}^n}{\sum_{k'} P_j(i|k')\phi_{k'}(\boldsymbol{x}_m)} \phi_k(\boldsymbol{x}_m).$$

If the r.h.s. of (9) is a contraction (in some metric), then the above can be solved by fixed point iterations to give a unique optimal solution. This is indeed the case here, noting that the expected log complete likelihood objective (5) is convex in the parameters[1] $P_j(i|k)$ and the constraint is also convex—so there is only one optimum as long as r_{mn} are held fixed—and furthermore noting (see later in Sect. 3) that each fixed point iteration is guaranteed not to decrease the objective. However, rather than carrying out this complete iterative M-step, we can employ a partial M-step instead. That is, we are only required to improve, and not necessarily to maximise the likelihood. In our implementation, we use one iteration of (9) as a partial M-step and this is interleaved with the E-step (6) and possibly the re-estimation of $P(\boldsymbol{x}_m)$.

$$P(\boldsymbol{x}_m) = \frac{1}{N} \sum_n r_{mn} \tag{10}$$

[1]This is immediate, using the definition of convexity and applying Jensen's inequality. Denoting $y_{ijm} \equiv \sum_n r_{mn} N_{ij}^n$, $a_{ijk} \equiv P_j(i|k)$ and $\phi_{km} \equiv \phi_k(\boldsymbol{x}_m)$, we have $\sum_m \sum_{ij} y_{ijm} \log \sum_k (\alpha a_{ijk} + (1-\alpha) b_{ijk}) \phi_{km} \leq \alpha \sum_m \sum_{ij} y_{ijk} \log \sum_k a_{ijk}\phi_{km} + (1-\alpha) \sum_m \sum_{ij} y_{ijk} \log \sum_k b_{ijk}\phi_{km}$ for any $\alpha \in (0,1)$ and any $\{b_{ijk}\}$ from the same space as $\{a_{ijk}\}$. Recall, the posteriors r_{mn} are held constant at this point. Convexity does not hold when r_{mn} are also unknown.

(unless we choose to keep the latter fixed at $1/M$). Overall this corresponds to a generalised EM procedure (McLachlan and Krishnan 1997) and is therefore guaranteed to converge to a local optimum of the data likelihood (3). In our experiments, we typically observed convergence within at most 35–40 iterations, in terms of a visually indistinguishable change in the mapping (and usually in 25–30 iterations to a tolerance of 10^{-3}, terms of the difference between two consecutive log likelihood values).

2.2 Summary of the Algorithm

The following notation is now employed for summarising the obtained algorithm in matrix form. The reshaped stochastic parameters $P_j(i|k)$ will be organised into a matrix $\boldsymbol{A}$ of $|S|^2$ rows and K columns. Further, the images of the latent space samples $\boldsymbol{x}_m$ through $\phi_k(.)$ will be the elements of a $K \times M$ matrix $\boldsymbol{\Phi}$. Finally, the $M \times N$ matrix $\boldsymbol{R}$ will contain the posterior probabilities r_{mn}, and $\boldsymbol{\alpha}$ is the vector $(P(\boldsymbol{x}_1), \ldots, P(\boldsymbol{x}_M))^T$ of the mixing coefficients. The bigram frequency counts from each data sequence, reshaped into a column of the data matrix are denoted by $\boldsymbol{D}$. Then our algorithm can be summarised as a loop till convergence, over the following updates:

$$\boldsymbol{R} \propto \exp\{\log(\boldsymbol{A}\boldsymbol{\Phi})^T \boldsymbol{D} + \log(\boldsymbol{\alpha})\mathbf{1}\}, \tag{11}$$

$$\boldsymbol{A} \propto \boldsymbol{A} \odot \{[\boldsymbol{D}\boldsymbol{R}^T] \oslash [\boldsymbol{A}\boldsymbol{\Phi}]\}\boldsymbol{\Phi}^T, \tag{12}$$

$$\boldsymbol{\alpha} = \boldsymbol{R}\mathbf{1}^T/N \tag{13}$$

where $\propto$ stands for proportionality, $\odot$ denotes element-wise matrix multiplication, $\oslash$ denotes element-wise division and $\mathbf{1}$ is an N-dimensional row-vector of ones. The proportionality in (12) should of course be understood block-wise, i.e. for each fixed j and k, the transitions must satisfy $\sum_i P_j(i|k) = 1$.

2.3 Scaling

One of the important strengths of this approach is that the resulting algorithm can exploit data sparseness. If $\boldsymbol{D}$ is sparse, then the matrix multiplication in the numerator takes $\mathcal{O}(N_D M)$ where N_D denotes the number of nonzero elements in $\boldsymbol{D}$ and M is the number of samples used for approximating the uniform latent space. Further, the matrix $\boldsymbol{\Phi}$ can also be made sparse by zeroing small probabilities below some threshold and re-normalising (recall, this matrix is fixed a priori), since distant neighbourhoods are unlikely by the model design. Then the remaining matrix multiplications are also able to exploit the sparsity of $\boldsymbol{\Phi}$. (Indicatively, in the experiments reported, we experienced no harm by using a threshold of 0.01.) Denoting the number of non-zero elements of $\boldsymbol{\Phi}$ by N_Φ, and the number of data features by F ($F = |S|^2$ if we work with first order transitions), the overall scaling of an E-step is $\mathcal{O}(FN_\Phi + FN_D)$ and that of an M-step is $\mathcal{O}(N_D M + F(N_\Phi + M + N_\Phi))$—both are multilinear, and in the case of large data sets, the dominant term is expected to be N_D. Therefore to summarise, we can say that the scaling is linear in the number of non-zero entries in the data matrix.

In contrast, existing forms of GTM (Bishop et al. 1998a) that are applicable to symbolic data, i.e. the multinomial latent trait model (Bishop et al. 1998b; Kabán and Girolami 2001), require numerical methods for nonlinear optimisation to be employed within M-steps. (The concrete form of the associated stationary equation will be discussed in Sect. 4.1.) As discussed in (Bishop et al. 1998b), iterative re-weighted least squares (IRLS) could be applied,

but is impractical when the symbol space is large, due to matrix inversions,[2] which result in a scaling of $\mathcal{O}(K^3F)$. Data sparsity cannot help us to gain efficiency in this case. Additionally, IRLS and certain other Newton-type optimisation methods are based on a local quadratic approximation of the log likelihood and are not guaranteed to monotonically increase the true log likelihood. This may cause convergence and stability problems. A gradient-based partial M-step was then suggested as a possible alternative (Bishop et al. 1998b; Kabán and Girolami 2001). We can observe that the actual evaluation of the gradient term can make use of sparse matrix multiplications (for the same reasons as above) and so this part of the computation scales exactly the same as one of our M-steps. However, in addition to this, the gradient procedure also requires a suitable learning rate parameter to be set, which is a serious practical limitation. Convergence may be very slow (if the learning rate is small) or not happening at all (if the learning rate is too large) (Bishop 1995). The local convergence guarantee can be met by setting the learning rate cf. the Robbins–Monro criterion, but this makes the progress towards convergence prohibitively slow and hence the overall computation time is substantially increased. One could of course employ line search methods (Kelley 1995; Bishop 1995) (although this has not been specifically discussed in the multinomial GTM literature), which may be expected to be more efficient. However, as it will be seen later in the experiments, the overall time required to convergence is still inhibiting in the case of large realistic data sets. The high dimensional and unconstrained parameter space of multinomial GTM appears to be more difficult to search than that of models with a positively constrained parameter space. This may be one of the reasons why linear models, such as PLSA and LDA (Blei et al. 2003), are more popular in the literature for modelling large discrete multinomial data sets. Another reason is certainly the lack of direct interpretability of the parameters.

In turn, our approach presented in the previous section does not require any learning rate parameter to be tuned. Note also that we did not need to make approximations to our data likelihood definition. Yet, a monotonic increase in log likelihood towards a local optimum is guaranteed within a nice and sound probabilistic generative framework. Despite the sampling-based inference employed, and the non-linearity of our model, our algorithm scales to large data sets and can exploit the sparseness of the data. Our parameters are also directly interpretable as probabilities. Hereafter, we will refer to this scalable approach as the Sparse Sequence-GTM (SGTM). We now detail the use of this approach for both visualisation and prediction.

2.4 Visualisation

As in (Bishop et al. 1998b; Kabán and Girolami 2001), the posterior expectations of the latent variable, $E[\boldsymbol{x}|S_n] \approx \sum_m \boldsymbol{x}_m r_{mn}$ may be employed to obtain 2D visualisation plots of the data collection, as a whole. Each sequence will correspond to one point in the latent space and the proximity relations between points will necessarily reflect those of the sequences via the model likelihood definition (3): The log of the likelihood term in (3), i.e. $\sum_{ij} N_{ij}^n \log \sum_k P_j(i|k)\phi_k(\boldsymbol{x}_m)$ by definition represents the negative Kullback–Leibler divergence (Cover and Thomas 1991) between the n-th observed sequence and the m-th com-

[2] Even if the off-diagonal elements of the class-conditional Fisher information matrices are discarded, a separate non-diagonal $K \times K$ matrix needs to be inverted for each feature.

ponent model, up to constants.[3] This is indeed the canonical divergence between Markov chains and it has been previously employed in (Hollmén et al. 1999) in a somewhat heuristic manner within a self-organising map of Markov chains. In contrast, here the Kullback–Leibler divergence falls out naturally from our generative model definition.

2.5 Prediction

The great benefit of having a generative model for multiple sequences is that once the parameters of the model are estimated from a collection of training sequences, we can make predictions for a previously unseen sequence, based on its history and our model. Assume S_n is a test sequence unseen at training. Then the predicted next symbol of S_n is computed as follows.

$$\begin{aligned} P(s_{\text{next},n}|S_n) &= \int d\boldsymbol{x}\, P_{s_{\text{last},n}}(s_{\text{next},n}|\boldsymbol{x})\, P(\boldsymbol{x}|S_n) \\ &= \int d\boldsymbol{x} \sum_k P_{s_{\text{last},n}}(s_{\text{next},n}|k)\phi_k(\boldsymbol{x})\, P(\boldsymbol{x}|S_n) \\ &= \sum_k P_{s_{\text{last},n}}(s_{\text{next},n}|k)\, E[\phi_k(\boldsymbol{x})|S_n] \\ &\approx \sum_k P_{s_{\text{last},n}}(s_{\text{next},n}|k) \sum_m \phi_k(\boldsymbol{x}_m) r_{mn}. \end{aligned} \tag{14}$$

It is insightful to observe that essentially this is a convex combination of individual prediction probabilities made by each component model in turn, weighted by the expectation $E[\phi_k(\boldsymbol{x})|S_n]$. It should be mentioned that even though each of these component models were first order Markovian throughout this paper, the entire history enters into the mentioned posterior expectation. In consequence there is no single Markovian model of any order that could replace the model of multiple sequences.

Formally, the same is true for simplicial mixtures for multiple sequences (Girolami and Kabán 2005) and mixtures of multiple sequences (Cadez et al. 2003), these being the only existing probabilistic models formulated explicitly for Markovian sequence collections to the best of our knowledge. However, because these models are linear and so they do not model any correlations between the components, then their posterior distributions typically tend to be sharper than those of a topographic model. That is, fewer components participate in 'explaining' the sequence history—in the case of mixtures, essentially just one. In turn, in our topographic model, the inferred position of a test sequence on a continuous topographic space encodes its relative position to the sequences seen at training. We expect this will regularise and improve the predictions. In the experimental sections we will demonstrate that this is indeed the case.

[3] Denoting $\hat{c}_j = \sum_i N_{ij}^n$, $\hat{p}_{ij} \equiv N_{ij}^n/\hat{c}_j$ and $\hat{\boldsymbol{p}}_{.j} \equiv (\hat{p}_{1j}, \hat{p}_{2j}, \ldots, \hat{p}_{|S|j})$, we have

$$\begin{aligned} \sum_{ij} N_{ij}^n \log \sum_k P_j(i|k)\phi_k(\boldsymbol{x}_m) &= \sum_j \hat{c}_j \sum_i \hat{p}_{ij} \log \sum_k P_j(i|k)\phi_k(\boldsymbol{x}_m) \\ &= \sum_j \hat{c}_j \left[-KL\left(\hat{\boldsymbol{p}}_{.j} || \sum_k P_j(.|k)\phi_k(\boldsymbol{x}_m)\right) - H(\hat{\boldsymbol{p}}_{.j}) \right], \end{aligned}$$

where both the data entropy $H(\hat{\boldsymbol{p}}_{.,j})$ and the term $\hat{c}_j$ are constants w.r.t. the model parameters.

3 Alternative Views and Analysis of Representation

From (3), the formulated model can essentially be seen as a constrained mixture, where the mean parameter of the k-th mixture component is the following.

$$\mu_{ij,m} = \sum_k P_j(i|k)\phi_k(\boldsymbol{x}_m). \tag{15}$$

Observe these are proper probabilities too: $\sum_i \mu_{ij,m} = 1 \ \forall j, m$. Thus, once the parameters $P_j(i|k)$ are estimated, *prototypical representations* (or local averages) are also obtained as (15), by the stochastic translation of the component parameters $\boldsymbol{A} = \{P_j(i|k)\}_{i,j,k}$ via $\boldsymbol{\Phi}$. Contrarily to GTM and LTM, however, the parameters of the model, $\boldsymbol{A}$, are now also readily interpretable probabilities. Moreover, in this section we show that they are in fact low entropy *components* of the data.

As an illustration, the left hand plots of Fig. 2 show the prototypes (analogous to reference-vectors in the SOM (Kaski et al. 1998)) created by the proposed algorithm from a set of grey-scale face images (Roweis et al. 2002). For the sake of this example, each image was taken as a histogram of grey levels over the pixel locations—i.e. the grey level of each pixel is treated analogously to the observed count for a symbol associated with that pixel. A 10×10 latent grid has been utilised and subsequently sub-sampled to display each third prototype. The right-hand plots of Fig. 2 show the parameters of the model, $\boldsymbol{A}$ (suitably reshaped), sub-sampled from a $K = 5 \times 5$ grid. As already mentioned in the text after (2), these are somewhat analogous to the component parameters of the so-called aspect models (Hofmann 2000; Blei et al. 2003). They are much sparser compared to the mean parameters $\boldsymbol{\mu}_m$. In the given face image example, they seem to retain the main characteristics of the face expressions only. To see why this is so, we will start from highlighting an alternative view of the presented model.

Since for each latent point $\boldsymbol{x}$, the nonlinear mapping $\phi_{\cdot}(\boldsymbol{x})$ creates a conditional multinomial probability $\phi_k(\boldsymbol{x}_m) = P(\boldsymbol{y} = \boldsymbol{y}_k|\boldsymbol{x} = \boldsymbol{x}_m)$ we may interpret the set of centres $\boldsymbol{y}_k$ as the discretisation of another continuous latent variable. Then the complete data likelihood with two latent variables may be written as the following.

$$\mathcal{L}'^C = P(S_n, \boldsymbol{x}, \boldsymbol{y}) = \prod_m P(\boldsymbol{x}_m)^{\delta(\boldsymbol{x}-\boldsymbol{x}_m)} \prod_{i,j,k} \{P_j(i|k)\phi_k(\boldsymbol{x})\}^{N_{ij}^n \delta(\boldsymbol{x}-\boldsymbol{x}_m, \boldsymbol{y}-\boldsymbol{y}_k)}. \tag{16}$$

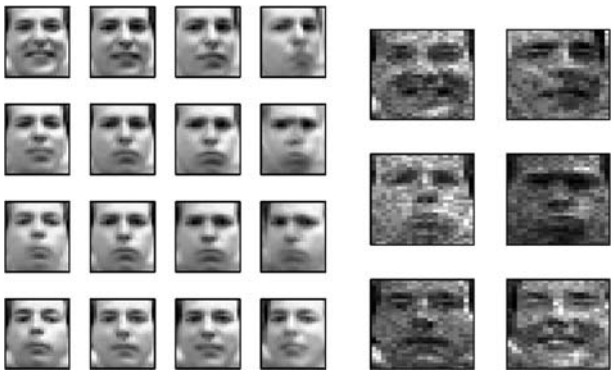

Fig. 2 Illustration of parameter interpretability: Prototypes μ_m (*left*) versus low entropy components $P(.|k)$ (*right*). White = 1, black = 0

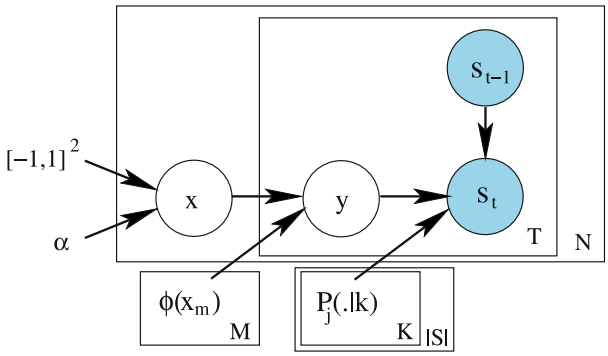

Fig. 3 Plate diagram of the alternative view of our generative model, with two latent variables $\boldsymbol{x}$ and $\boldsymbol{y}$

In this view, the expected complete data log likelihood is:

$$E[\log \mathcal{L}'^{C}] = \sum_{n,m} r_{mn} \left\{ \sum_{i,j,k} r_{kmij} N_{ij}^{n} \log P_j(i|k)\phi_k(\boldsymbol{x}_m) + \log p(\boldsymbol{x}_m) \right\} \tag{17}$$

where, as before, $r_{mn} \equiv P(\boldsymbol{x}_m|S_n)$ and using a similar notation, we have in addition the posterior $r_{kmij} \equiv P(\boldsymbol{y}_k|\boldsymbol{x}_m, s_{tn} = i, s_{t-1,n} = j)$.

There is an intuitive generative process associated to this alternative view, having analogies to the 'noisy channel' (Hofmann 2000; Hofmann and Buhmann 1998) based coding and data transmission models. From this analogy, our model may also be seen as one possible probabilistic generative version of noisy channel coding. (Note that neither (Hofmann 2000) nor (Hofmann and Buhmann 1998) are probabilistic models with a generative semantics.) The generative process associated with (16) is the following and the corresponding plate diagram is detailed in Fig. 3.

- generate a point $\boldsymbol{x}$ in the latent space from the prior density $p(\boldsymbol{x})$. (This point is sequence-specific.)
- for each time point till the length of the sequence, $t = 1, \ldots, T_n$.
  - generate a *situated* point $\boldsymbol{y}_k$ in the second latent space, conditioned on $\boldsymbol{x}$, with the 'channel noise' probability $P(k|\boldsymbol{x}) = \phi_k(\boldsymbol{x})$. (This point is symbol-specific.)
  - generate the next symbol s_t from the k-th component generator model, i.e. with probability $P_{s_{t-1}}(.|k)$.

A somewhat similar generative process has been proposed in (Keller and Bengio 2004) in the context of text document modelling, where the sequence-specific hidden variable is associated with higher level themes whereas the symbol-specific latent variables are meant to signify topics within a theme. The difference is that unlike (Keller and Bengio 2004), we require that topics descending from a theme must be in the topographic neighbourhood of that theme with high probability. Thus, the sharing of topics is constrained, ensuring that each theme will encompass a different distribution of topics.

We may perform the model identification starting from the above alternative formulation of the model. To maximise (17), the posterior probabilities r_{mn} and r_{kmij} are both computed in the E-step. Using Bayes theorem and marginalising over $\boldsymbol{y}$ the E-step equation for computing r_{mn} is identical to (6), and using Bayes theorem once more to compute r_{kmij}, we have

$$r_{kmij} = \frac{P_j(i|k)\phi_k(\boldsymbol{x}_m)}{\sum_{k'} P_j(i|k')\phi_{k'}(\boldsymbol{x}_m)}. \tag{18}$$

Now, maximising (17) in the M-step, w.r.t. to the parameters and subject to the required constraints amounts to maximising the following Lagrangian

$$\tilde{Q}' = E[\log \mathcal{L}'^{C}] - \sum_j \sum_k u_{jk} \left(\sum_i P_j(i|k) - 1 \right) - v \left(\sum_m P(\boldsymbol{x}_m) - 1 \right) \tag{19}$$

where, as before, u_{jk} and v are Lagrange multipliers. The stationary equations for the parameters $P_j(i|k)$ follow as

$$\frac{\delta Q'}{\delta P_j(i|k)} = \sum_n \sum_m r_{mn} r_{kmij} N_{ij}^n \frac{1}{P_j(i|k')} - u_{jk} = 0. \tag{20}$$

Multiplying both sides by $P_j(i|k)$ we obtain a closed form solution

$$P_j(i|k) = \frac{1}{u_{jk}} \sum_n \sum_m r_{mn} r_{kmij} N_{ij}^n \tag{21}$$

and by summing both sides over i, the normalisation constant is now

$$u_{jk} = \sum_{i'} \sum_n \sum_m r_{mn} r_{kmi'j} N_{i'j}^n. \tag{22}$$

From the theory of the EM algorithm we know that both (5) and (17) are so-called auxiliary functions (McLachlan and Krishnan 1997) to the same data likelihood (3). That is, each E and M step is guaranteed not to decrease the data likelihood (3) and a local maximum of either (5) or (17) is also a local maximum of (3).

It is interesting to note that by replacing (18) into (21), and rearranging, the fixed point update (9) is recovered. Hence (as already anticipated in Sect. 2) each fixed point iteration of the form (9) is also guaranteed not to decrease the data likelihood—which in turn completes the arguments used for the convergence claims made for the algorithm given in the previous section (see Sect. 2). It is also obvious that from an efficient implementation point of view, the algorithmic form given previously is more convenient since the posterior probabilities of $\boldsymbol{y}$ need not be explicitly computed and stored.

However, the alternative view presented here not only offers a hierarchical interpretation of the model as an insight, but it will also be used to shed light on the observed low-entropy characteristic of the model parameters $P_j(i|k)$, as well as on the topographic organisation ability of the model. This is what we analyse in the sequel. To begin with, let us rewrite the expected complete log likelihood (17), with the use of the M-step equations (21) and (10).

$$\begin{aligned} E = & \sum_k \sum_j \left\{ u_{jk} \sum_i P_j(i|k) \log P_j(i|k) \right\} \\ & + \sum_m \left\{ \sum_k \left[\sum_n \sum_{i,j} r_{mn} r_{kmij} N_{ij}^n \right] \log \phi_k(\boldsymbol{x}_m) \right\} \\ & + N \sum_m P(\boldsymbol{x}_m) \log P(\boldsymbol{x}_m) = \text{Term1} + \text{Term2} + \text{Term3}. \end{aligned} \tag{23}$$

In (23), u_{jk}, $P_j(i|k)$ and $P(\boldsymbol{x}_m)$ are all functions of r_{mn} and r_{kmij}. Naturally, when $P(\boldsymbol{x}_m)$ is fixed to uniform, then the last term becomes a constant.

3.1 Low Entropy Components

Now let us observe that the first of the above terms is the negative of a weighted sum of entropies. Using the shorthand $a_{ik}^j \equiv P_j(i|k)$, and $\boldsymbol{a}_k^j \equiv P_j(.|k)$, the first term reads as

$$\text{Term1} = -\sum_k \sum_j u_{jk} H[\boldsymbol{a}_k^j] \tag{24}$$

where the entropy (Cover and Thomas 1991) is defined as $H(\boldsymbol{a}_k^j) = -\sum_i a_{ik}^j \log a_{ik}^j$ and the dependencies on r_{mn} and r_{kmij} are implicit. From (22), we also have the meaning of the weighting factors: each u_{jk} represents the expected number of symbols 'explained' by the k-th generator in the context of j. A maximisation of this term would signify that the generators that contribute more, must have lower entropies. Since by definition the entropy of a distribution is a measure of uncertainty, low entropy component models are interesting, because they tend to be both more interpretable and more predictable.

3.2 Topographic Organisation

The second term is concerned with the distribution of the K symbol-level generators, for each latent point $\boldsymbol{x}_m$. Rearranging this term by denoting $p_{km} \equiv \frac{\sum_{nij} r_{mn} r_{kmij} N_{ij}^n}{c_m}$ where $c_m \equiv \sum_{k'} \sum_{nij} r_{mn} r_{k'mij} N_{ij}^n = \sum_{nij} r_{mn} N_{ij}^n = \sum_n r_{mn} T_n$, we obtain:

$$\begin{aligned} \text{Term2} &= \sum_m c_m \left\{ \sum_k p_{km} \log \phi_k(\boldsymbol{x}_m) \right\} \\ &= \sum_m c_m \{-KL(\boldsymbol{p}_{.m} || \phi \cdot (\boldsymbol{x}_m)) - H(\boldsymbol{p}_{.m})\} \end{aligned} \tag{25}$$

where $KL(.||\cdot)$ is the Kullback–Leibler divergence (Cover and Thomas 1991) between the distribution of the expected probability of symbols explained by the various K basis functions, $\boldsymbol{p}_{.m}$, and the pre-defined Euclidean neighbourhood probability distribution $\phi \cdot (\boldsymbol{x}_m)$ associated with $\boldsymbol{x}_m$, defined as $KL(\boldsymbol{p}_{.m} || \phi \cdot (\boldsymbol{x}_m)) = \sum_k p_{km} \log \frac{p_{km}}{\phi_k(\boldsymbol{x}_m)}$. The last term of (25) is the entropy of the distribution of symbol-level generators associated with a latent point $\boldsymbol{x}_m$. The weighting factor c_m again represents a notion of importance of the latent point $\boldsymbol{x}_m$ in terms of the expected total number of symbols of sequences 'explained'. Naturally, both p_{km} and c_m are functions of r_{mn} and r_{kmij}, and the explicit dependence on these quantities was omitted in the notation for brevity.

From (25) we can see that a larger value of Term2 implies that for each $\boldsymbol{x}_m$ the assignments of generators $k = 1, \ldots, K$ are such that: (i) Even if the symbols of each sequence may be generated from different generators $k = 1, \ldots, K$ (shared by other sequences too), those which are situated around the sequence-specific latent point $\boldsymbol{x}_m$ must be more probable; (ii) Each $\boldsymbol{x}_m$ should have a small number of active generators associated with it. The former property is a key difference from generative aspect models (Blei et al. 2003; Girolami and Kabán 2005), where the generators are allowed to interleave without constraints. In turn, this constraint has the effect of a topographic ordering of sequence histories that is useful both for its regularising effect and for explanatory data organisation. These issues will be demonstrated in more detail in the experimental section.

3.3 The 2D Latent Density

The last term of (23) is again a negative entropy, weighted by the total number of data instances N.

$$\text{Term3} = -H[\boldsymbol{\alpha}] \times N \tag{26}$$

where $\boldsymbol{\alpha}$ is now a function of r_{mn}. A higher value of this term implies the creation of regions of varying density in the latent 2D space.

3.4 Putting all Together: A Multi-Objective Optimisation View

We may view the above three terms together as a tradeoff of multiple objectives being maximised simultaneously during the EM iterations. Indeed, if we were to start off from maximising the objective (23) as a function of r_{mn} and r_{kmij}, subject to these having to be the expectations of some discrete assignments and additionally, subject to the two constraint definitions (21) and (10), then solving the associated discrete search problem for the pairs $\boldsymbol{x}_m$ and $\boldsymbol{y}_k$ over the finite set when $m = 1, \ldots, M$ and $k = 1, \ldots, K$ by the mean field trick (Peterson and Söderberg 1989) (given in the Appendix for completeness), yields, after some straightforward algebra, exactly the EM algorithm (6) & (18) & (21) & (10). By implication, the generalised EM solution (11–13) implicitly optimises the same objective.

In conclusion, in this section we have shown that (1) the minimisation of the component parameter entropies is implicitly part of the objective, thus the representation will tend to create low entropy parameters. (2) At the same time, it will try to organise the latent point assignments in a locally topography preserving manner. (3) Finally, if the uniformity of the latent distribution is not imposed, then rather than using the entire latent space uniformly, the latent point assignments will have the flexibility to create regions of high density, that may reflect a clustered structure in the data.

4 Relation to Previous Topographic Models

4.1 Natural Parameter Based Modelling: Relation to Multinomial GTM

As already mentioned, the proposed approach, in its form presented in Sect. 2, is conceptually related to the multinomial latent trait GTM model (Bishop et al. 1998b; Kabán and Girolami 2001). However, there is an important structural difference in that the latter proceeds at modelling the natural parameters of the multinomials

$$\boldsymbol{\mu}_m = g(\boldsymbol{A}\boldsymbol{\phi}(\boldsymbol{x}_m)) \tag{27}$$

and so the mapping from the Euclidean space to the space of probabilities is achieved through the inverse-link function $g(\theta_{im}) = \exp(\theta_{im})/\sum_i \exp(\theta_{im})$. Consequently, the parameter matrix of the multinomial GTM is still in an Euclidean space and therefore it is not interpretable. By contrary, as we have seen, the proposed approach models the mean parameter directly (15). Thus, our model parameters live in the space of probabilities and as such, they can readily be interpreted as probabilities and low-entropy components of the data.

Secondly, this difference also implies that unlike the positively constrained parameter space in SGTM, the parameter space of multinomial GTM is unconstrained. Such extent

of flexibility may not be required for the case of high dimensional sparse data (which will typically exhibit a high degree of redundancy). In turn, the unconstrained parameter space is more difficult to search. Specifically, due to the nonlinear inverse link, in each M-step, a nonlinear equation of the following form needs to be solved (Kabán and Girolami 2001).

$$\boldsymbol{D}\boldsymbol{R}^T\boldsymbol{\Phi}^T = g(\boldsymbol{A}\boldsymbol{\Phi})\boldsymbol{G}\boldsymbol{\Phi}^T \tag{28}$$

where the matrix notations used are as before, and the matrix $\boldsymbol{G}$ is diagonal with elements $\sum_n r_{mn}$ (Kabán and Girolami 2001). Clearly, there is no closed form solution to this equation. As discussed in Sect. 2.3, existing developments are somewhat lacking in terms of computational efficiency. Care must also be taken about the tradeoff between the convergence guarantee of the optimisation and a possibly long convergence time. Such problems are not encountered with our SGTM model definition.

4.2 Dealing with Very High Dimensions: Relation to ProbMap

In this subsection we develop a simple extension of the proposed model for the case of excessively large state spaces. The main purpose here is to highlight a close relation with the ProbMap model of (Hofmann 2000). It should be mentioned, however that neither our extension nor ProbMap are fully generative.

We note that although due to its constrained nature, our model is able to deal with fairly high dimensions—as our early results on text modelling over a dictionary size of nearly ten thousand words have demonstrated (Kabán 2005)—when excessively increasing the data dimensionality (size of the state space), the converged posteriors may become very sharp. This is due to the fact that in the absence of sufficient amounts of data (compared to the dimensionality of the problem), the mixture of Dirac deltas prior dominates.

Retaining the discretisation of the latent space, which is desirable for tractability reasons, in this subsection we will adopt a simple form of modular mixture (Attias 2001; Blei et al. 2003) approximation in order to make our model able to deal with very high dimensional problems. This essentially introduces a convex linear interpolation among the centres of the Dirac deltas with the aid of an additional random variable that defines a distribution over the latent space samples $\boldsymbol{\pi} = P(\{\boldsymbol{x}_{1:M}\})$. The data distribution conditioned on $\boldsymbol{\pi}$, where $\sum_m \pi_m = 1$, will be defined as follows.

$$p(S_n|\boldsymbol{\pi}) = \prod_{i,j}\left\{\sum_k P_j(i|k)\varphi_k(\boldsymbol{\pi}, \{\boldsymbol{x}_{1:M}\})\right\}^{N_{ij}^n} \tag{29}$$

with

$$\varphi_k(\boldsymbol{\pi}, \{\boldsymbol{x}_{1:M}\}) = \sum_m \pi_m \phi_k(\boldsymbol{x}_m). \tag{30}$$

The posterior expectations for visualisation and prediction, as per Sects. 2.4. and 2.5), will now both depend on the posterior expectations of $\boldsymbol{\pi}$:

$$E[\boldsymbol{x}|S_n] = \sum_m \boldsymbol{x}_m P(\boldsymbol{x}_m|S_n) = \sum_m \boldsymbol{x}_m \int P(\boldsymbol{x}_m|\boldsymbol{\pi})q(\boldsymbol{\pi}|S_n)d\boldsymbol{\pi} \tag{31}$$

$$= \sum_m \boldsymbol{x}_m E_{q(\boldsymbol{\pi}|S_n)}[\pi_m] = \sum_m \boldsymbol{x}_m E[\pi_m|S_n] \tag{32}$$

where $q(\boldsymbol{\pi}|S_n)$ denotes the (approximate) posterior of $\boldsymbol{\pi}$ employed for the inference of $\boldsymbol{\pi}$ and $P(\boldsymbol{x}_m|\boldsymbol{\pi}) = \pi_m$. Clearly, each observation will have its own posterior over the mixing coefficients, $q(\boldsymbol{\pi}|S_n)$, and consequently its own posterior expectation $E[\pi_m|S_n]$. Analogously,

$$E[\phi_k(\boldsymbol{x})|S_n] = \sum_m \phi_k(\boldsymbol{x}_m) E[\pi_m|S_n]. \tag{33}$$

Defining an appropriate prior distribution for the convex coefficients $p(\boldsymbol{\pi})$ analytically is, however, not straightforward. Due to the special structure induced by the neighbourhood probabilities and the mapping from an Euclidean space to the space of probabilities, we found in our experiments that a Dirichlet is inappropriate in this case. For this reason, as well as since our scope in this section is mainly to arrive at a connection with the ProbMap method of (Hofmann 2000), here we will adopt a noninformative uniform prior for $\boldsymbol{\pi}$ and perform a simple Maximum a Posteriori/Maximum Likelihood point-estimation. That is, we approximate $q(\boldsymbol{\pi}|S_n) \approx \delta(\boldsymbol{\pi} - \boldsymbol{\pi}_n^{ML})$ and so $E_{\delta(\boldsymbol{\pi}-\boldsymbol{\pi}_n^{ML})}[\pi_k] = \pi_{kn}^{ML}$. Now, for each S_n, $\boldsymbol{\pi}_n^{ML}$, needs to be estimated.

As before, we organise the parameters $\{P_j(i|k)\}$ into a matrix $\boldsymbol{A}$ and in addition we also organise the point estimates π_{mn}^{ML} into an $M \times N$ matrix $\boldsymbol{\Upsilon}$. Solving for all stationary equations for both sets of variables, subject to the required constraints ($\sum_i P_j(i|k) = 1$ and $\sum_m \pi_{mn}^{ML} = 1$) we arrive at the convergent alternating iterative algorithm below, written in matrix notation.

$$\boldsymbol{\Upsilon}^{(\text{new})} \propto \boldsymbol{\Upsilon}^{(\text{old})} \odot \{\boldsymbol{A}^T[\boldsymbol{D} \oslash (\boldsymbol{A}\boldsymbol{\Phi}\boldsymbol{\Upsilon}^{(\text{old})})]\}, \tag{34}$$

$$\boldsymbol{A}^{(\text{new})} \propto \boldsymbol{A}^{(\text{old})} \odot \{[\boldsymbol{D} \oslash (\boldsymbol{A}^{(\text{old})}\boldsymbol{\Phi}\boldsymbol{\Upsilon})]\boldsymbol{\Upsilon}^T\}. \tag{35}$$

The posterior statistics required in (32) for visualisation in this case reduce to computing $E[\boldsymbol{x}|S_n] = \sum_m \boldsymbol{x}_m E[\pi_m|S_n] \approx \sum_m \boldsymbol{x}_m \pi_{mn}^{ML}$, and $E[\phi_k(\boldsymbol{x})|S_n] \approx \sum_m \phi(\boldsymbol{x}_m)\pi_{mn}^{ML}$, where π_{mn}^{ML} are newly estimated for previously unseen data instances, while maintaining the parameters $P_j(i|k)$ fixed. This is essentially the empirical Bayes methodology (Bernardo and Smith 2001; Blei et al. 2003), i.e. the distribution of $\boldsymbol{\pi}$ is defined by the set of samples estimated from the data. The sample estimates obtained from the training set may also be used for computing the likelihood of new data points under the model.

Now let us observe that making abstraction from the continuous latent variables, and inspecting (34–35) formally, these are identical to Hofmann's ProbMap (Hofmann 2000) (written in matrix form). ProbMap (Hofmann 2000) is not a generative model, therefore its functionality is restricted to organising data for exploratory purposes. The way to assess its generalisation abilities is not well defined. However, from the analysis made in this section, the empirical Bayes procedure may in principle be used to extend its functionality.

4.3 Joint Clustering and Visualisation: Relation to Parametric Embedding

Parametric Embedding (PE) (Iwata et al. 2005) is a recently proposed technique that takes class membership probabilities obtained e.g. from a clustering algorithm and visualises them in 2D. It is not a generative model, so it has no predictive abilities, instead, it was devised for the sole purpose of visualisation. By contrary, in our approach, the visualisation function is built into the model on the grounds of a predictive model with generative semantics. However there are some structural analogies that could be followed between these two approaches. For this section, the models are assumed to be zeroth order

i.e. $P(s_{tn}|s_{t-1,n}, k) = P(s_{tn}|k)$. Let us inspect the log of the likelihood term in our model formulation (1),

$$\sum_n \log P(S_n|\boldsymbol{x}) = \sum_n \sum_t \log \sum_k P(s_{tn}|k)\phi_k(\boldsymbol{x}) \tag{36}$$

where t is a data feature and $P(s_{kt}|k)$ is now a multinomial probability parameter, however an arbitrary distributional form may be defined instead, if needed. Further, k may be thought of as a cluster variable. To see the connection with PE more formally, let us decouple (36) into two separate objectives, by replacing $\phi_k(\boldsymbol{x})$ by a 'dummy' probabilistic variable $P(k|S_n)$, which we then require to be close in Kullback–Leibler sense to $\phi_k(\boldsymbol{x})$. The modified expression is now:

$$\sum_n \sum_t \log \sum_k P(s_{tn}|k)P(k|S_n) - \sum_n \sum_k KL(P(k|S_n)||\phi_k(\boldsymbol{x})) \tag{37}$$

where as before,

$$\phi_k(\boldsymbol{x}) = \frac{\exp(-\frac{1}{2\sigma^2}|\boldsymbol{y}_k - \boldsymbol{x}|^2)}{\sum_{k'} \exp(-\frac{1}{2\sigma^2}|\boldsymbol{y}_k - \boldsymbol{x}|^2)}.$$

Now the first term is clearly a clustering objective, identical to the log of the likelihood term in of aspect-style of models, e.g. Latent Dirichlet Allocation (LDA) (Blei et al. 2003) or Multinomial PCA (MPCA) (Buntine 2002), whereas the second term is identical to the PE objective (Iwata et al. 2005). However, PE proposes to fully decouple these two tasks, by completing the optimisation of the first objective before optimising the second. This has the advantage of a full modularity, at the expense of sub-optimality due to accumulating errors. For instance, if a data set has no clear clusters or simply the cluster membership estimates $P(k|S_n)$ happen to be a poor summary of the data, then the subsequent PE visualisation is compromised. In turn, our approach implicitly optimises for both the above objectives simultaneously.

A further difference may be followed on the algorithmic level. The estimation of PE proceeds by considering both $\boldsymbol{y}_k$ and $\boldsymbol{x}$ as parameters and optimising. Instead, we obtain posterior distributions over the discrete samples from our generative latent variables, which may then used both for making inferences or predictions about previously unseen data instances, and to produce data summaries for visualisation. The former is not possible with PE, in its existing form.

The experimental section will provide detailed assessment of the generalisation performance of our approach and its robustness against small sample sizes of various kind and comparisons with existing models of multiple sequences will be made on this ground. In addition, as a byproduct of our model design, we also obtain interpretable parameters and intuitively meaningful visual representations, which, necessarily provide explanations of the data that are reflecting its predictive encoding. Experimental comparisons with other visualisation methods would, however, not be straightforwardly fair and are therefore outside of our scope. This is primarily either or both because the existing visualisation methods were not devised for multiple sequence data sets, or because they were not devised as predictive models. We believe there would be little basis for objective comparison in this sense. Moreover, there is no universally valid objective criterion for comparing visualisation plots, while there are well defined criteria for measuring the predictive performance, that we can use by exploiting the generative nature of our model.

5 Numerical Simulations: Prediction and Explanatory Representation

A toy experiment will illustrate the working of our method first. The data was generated from a 10×10 uniform grid of points in the 2D latent space $\boldsymbol{x}_m \in [-1, 1]^2$, passed through a 4×4 set of functions $\phi_k(\boldsymbol{x}_m), k = 1, \ldots, 16$ and mixed with some (4×4) randomly generated and sufficiently low entropy parameters $P_j(i|k)$ over a common state space of seven symbols. From each of the resulting 100 transition matrices $P_\cdot(.|\boldsymbol{x}_m), m = 1, \ldots, 100$ from this manifold of transition matrices, one sufficiently long sequence was drawn (we used random lengths $T_n \in [5000, 9000]$). These 100 sequences were then fed into our algorithm. Figure 4 shows examples of the obtained posterior expectations corresponding to local optima obtained in two independent runs with fixed uniform priors and a run with estimated priors. We can see the local topology has been well recovered. As discussed earlier, we have no uniqueness guarantees when both the latent points and the component transitions (the model parameters) are unknown, and so (similarly to GTM) different local optima will produce slightly different results. However, the local topology will be preserved due to the smoothness of the generative mapping, which is a useful property for data visualisation. Moreover, we now turn to demonstrate and objectively asses its beneficial role for prediction.

The next set of experiments studies the predictive capabilities of our algorithm and the effects of finite sample sizes on the performance. We vary both the average sequence lengths and the number of sequences available for training.

For each experimental setting, we generated 1500 sequences from a set of 3 generator processes over a symbol dictionary of 5 symbols. The actual generator models are shown on Fig. 5.

In order not to favour our model over competing approaches, the following two extreme generation procedures have been employed: (1) a mixture of Markov Chains (MMC) (Cadez et al. 2003), i.e. a model having a mixture of delta prior over K different Markov chain generators, and (2) a simplicial mixture of Markov chains (SMMC) (Girolami and Kabán 2005),

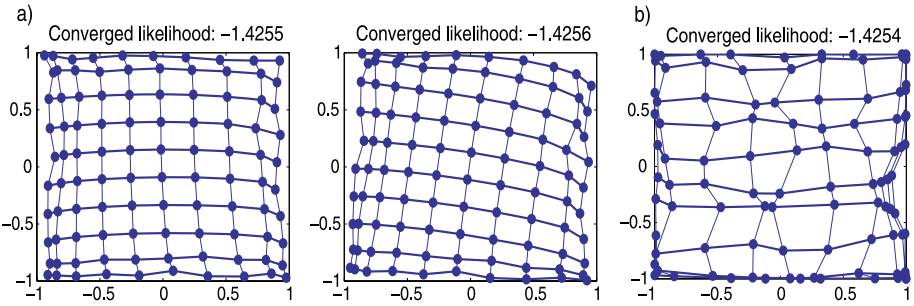

Fig. 4 Examples of recovered posterior mean mapping from 10×10 toy sequences generated from the model: **a** fixed uniform prior; **b** estimated priors. In all runs, the local topology is well preserved

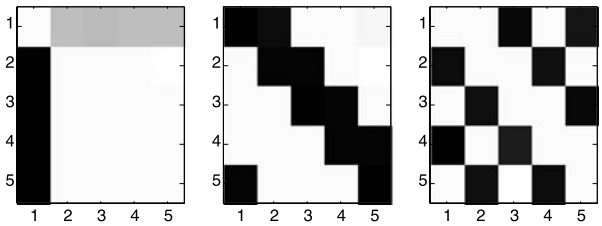

Fig. 5 The three generator Markov transitions that were used to create the synthetic data sets. Darker encodes higher probability, and each row sums to one

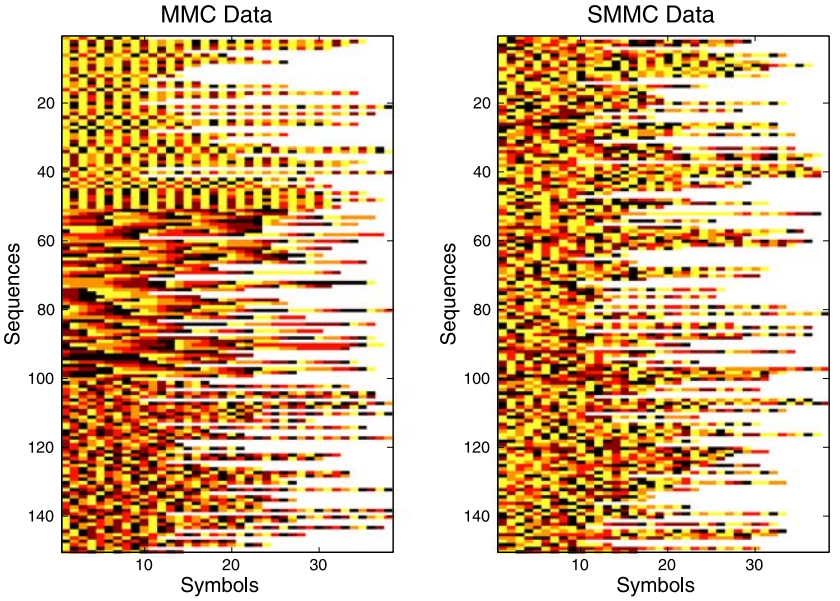

Fig. 6 Examples of sequences generated over a state space of five symbols. Each row is a sequence of varying length and each symbol is represented by a unique colour

employing a uniform Dirichlet prior over K different Markov chain generators. A few instances of the resulted data sequences are shown in Fig. 6 for illustration. It can easily be observed that in the case of MMC sequences, one can visually tell apart the trajectories of the three groups of sequences. Indeed, since each sequence is entirely generated by one of the generator models, all sequences generated from the same generator model have the similar characteristic dynamic patterns. By contrary, the SMMC trajectories result from interleaving all three generators in uniform instance-specific proportions, and so there is no natural grouping among the set of trajectories.

Six data sets have been generated from the above two models: (i) Long sequences (relative to the dictionary size), with lengths evenly distributed between a minimum length of 18 symbols up to a maximum length of 400 symbols; (ii) Medium length sequences, having lengths between 10 to 40 symbols; (iii) Short sequences, with lengths between 4 to 15 symbols—all these from both the MMC and SMMC generation procedures respectively, totalling six data sets. Further, in order to also assess the issue of sensitivity w.r.t. the training set size, each of these data sets have been considered in two different instances: In a first instance, 90% of the data has been employed for estimating the model, the remaining data being utilised as out of sample sequences for testing. In a second instance, only 10% of the data has been used for training and the remainder 90% was used for testing.

From each of these data settings, two variants of our topographic model—with fixed uniform prior, and with estimated prior—were estimated, along with MMC (Cadez et al. 2003) and SMMC (Girolami and Kabán 2005) models. For the latter, we used the variational estimation procedure described in (Girolami and Kabán 2005; Blei et al. 2003) (since the SMMC model is known to be intractable). For each experiment, 15 independent, randomly initialised parameter estimation runs, across 10 disjoint folds have been performed and the number of components tested ranged on a quasi logarithmic scale between 2 and 50. The

results are summarised in Fig. 7, in terms of out-of-sample log likelihoods, over a range of model orders listed on the log 10 scale. In the case of SGTM, the out of sample log likelihood is computed as the log of (3), from held out subsets of sequences. For MMC and SMMC, the associated expressions are those in (Cadez et al. 2003) and (Girolami and Kabán 2005) respectively. The performance of Global Markov chains (Sarukkai 2000) of various order are also shown for comparison. The latter uses a single Markov chain (of some order) to modelling the entire collection of sequences in a data set, and thus the variability of individual sequences can only be accounted for by increasing the order of the Markov chain.

Since no detailed and controlled assessment of the sample size requirements of the earlier methods (MMC and SMMC) are found in the literature, it is useful to first summarise the relevant findings regarding these two algorithms before using them as a basis for comparisons to our SGTM.

- MMC is more prone to overfitting due to small number of sequences in the training set, while being more robust against the issue of short sequences. SMMC, in turn, is more prone to overfitting due to shortness of sequences, however it is more robust against the issue of small number of training sequences. The reasons for this may be traced back to the definition of these two models and the algorithmic details of the associated estimation procedures.
- Unsurprisingly, from the last two columns of plots, in the case of mixture data (i.e. well separated clusters) the performance of both MMC and SMMC algorithms behave similarly and there was no statistically significant difference between the best results at the 5% level as tested by the Wilcoxon rank-sum test, except in the case of very short sequences where SMMC overfits earlier than MMC.
- While global Markov chains of sufficiently large order are able to outperform both MMC and SMMC of first order models on the mixture data, this is not the case on simplicial mixture data. Of course, this observation may be data dependent (e.g. in the case of a large state space, the high order global model may overfit more easily).

Remember both these existing state-of-the-art predictive models of multiple sequences are linear. Therefore their flexibility is limited and their abilities are rather complementary. Our topographic model is in turn nonlinear, which is expected to be an advantage. From the comparison the main empirical findings regarding our SGTM are summarised as follows.

- Firstly, the predictive performance of SGTM with estimated prior is comparable with the best out of the MMC and SMMC estimates. This is because unlike the latter linear models, the correlation structure between latent points is part of the modelling. So, for example in the case of a short sequence, SGTM is able to automatically complement the information with that coming from correlated other sequences. Further, in the case of a small number of sequences in the training set, the model is able to populate the clusters with partial memberships from neighboring clusters.
- Secondly, SGTM with a fixed uniform prior significantly outperforms all other models on simplicial mixture data while it is significantly suboptimal on mixture data.

Naturally, if the mixing is ‘diverse’, then the uniform latent density is the best suited and having this fixed rather than estimating it provides an advantage. In turn, if there are clearly distinct clusters in the data set, then assuming a uniform spread is suboptimal from the predictive density modelling point of view since it suffers from underfitting. In this latter case, SGTM is still well suited for visualisation purposes but less suited as a predictive model. To illustrate this point, Fig. 8 shows the posterior mean visualisation of one of the mixture

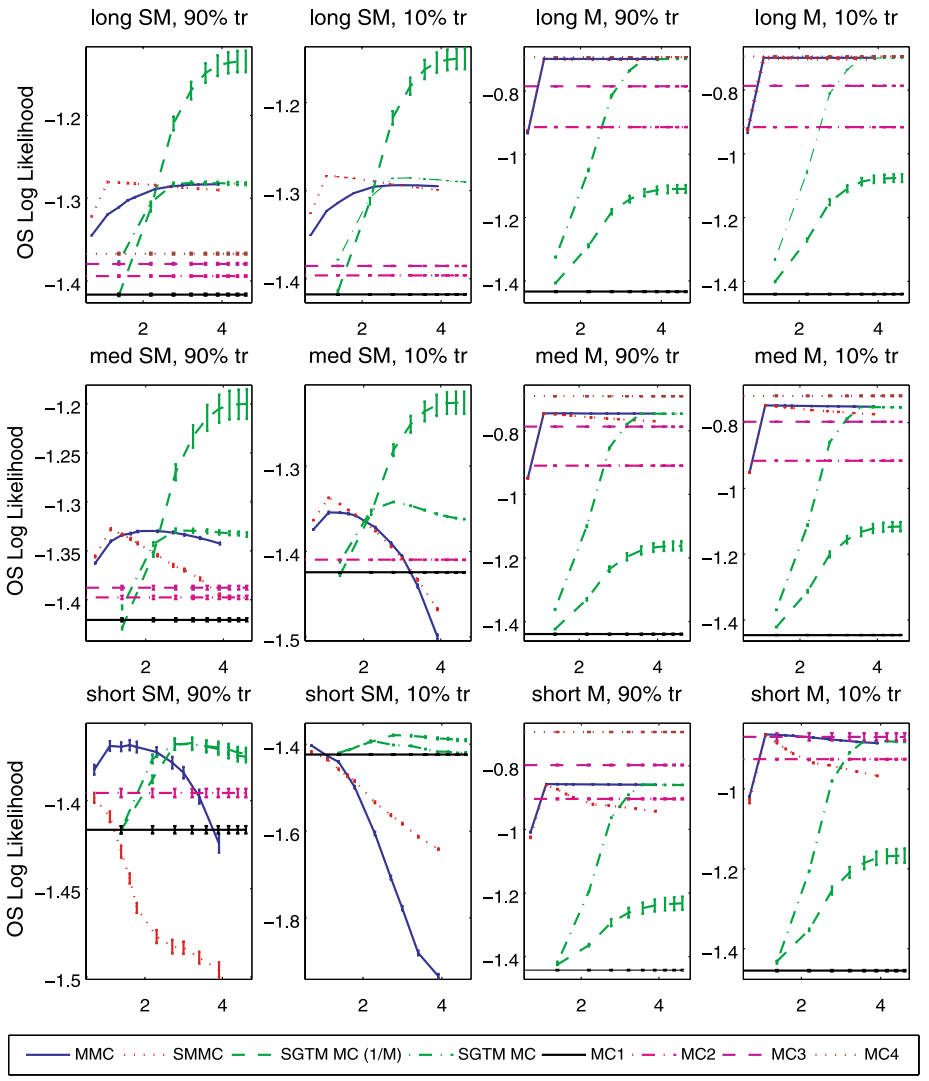

Fig. 7 Comparative results of the predictive generalisation performance as evaluated on generated data sets. On all plots, the out-of-sample log likelihood is given (*on the vertical axis*) versus the number of components on the log 10 scale (*on the horizontal axis*)—higher value indicates better generalisation. The error bars give one standard error over ten non-overlapping folds. The acronyms are the following: 'SM': data generated from a simplicial mixture (*leftmost two columns*); 'M': data generated from a mixture (*rightmost two columns*); 'n% tr': the percentage of sequences used for training (the remainder were used for testing). The average length of the sequences decreases from the *top row* to the *bottom*. MMC, SMMC and the two versions of SGTM (with fixed uniform prior (1/M) and with estimated prior mixing coefficients respectively) all utilise first-order Markovian components. The *straight lines* stand for global Markov models of varying order on each plot, these are given up to the order that still improves over the previous one. (The competing models are made with different line styles and also in different colors for enhanced clarity in case of colour viewing)

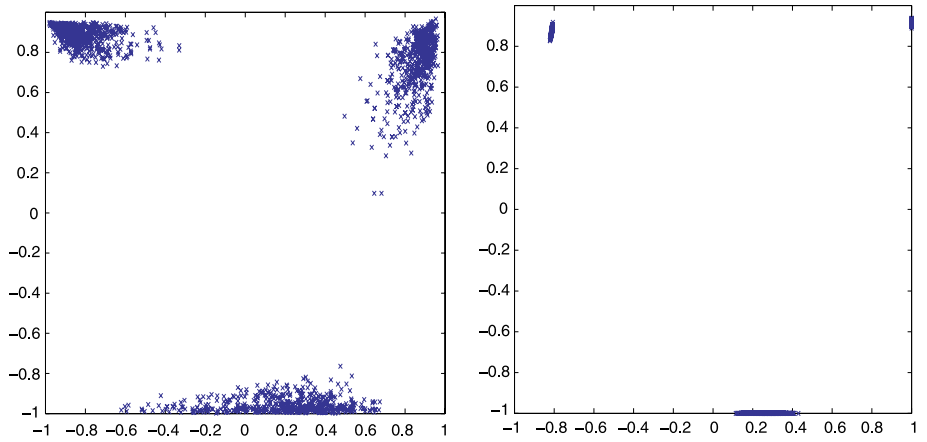

Fig. 8 *Right*: Posterior mean visualisation obtained by SGTM with fixed uniform prior, on the set of medium length sequences generated from a mixture of three Markov chains. *Left*: Posterior mean visualisation of the same data set, with SGTM with estimated mixing coefficients

data sets (the medium length sequences) obtained from a SGTM with fixed uniform prior. The cluster structure is well displayed. However, if SGTM with estimated prior probabilities is employed, then the spread of the three clusters shrink—so we see less details within the clusters but the obtained summary correctly reflects the generative distribution of the data. As we have seen, this is seconded by the increased predictive performance.

The above considerations regarding the influence of the latent prior are useful from the methodological point of view, since knowing the implications of our modelling choices enables us to employ them in an appropriate way and in accordance with the applications purposes. It should also be highlighted, however, that in the case of real-world data, we may mostly expect intermediate cases rather than clearly distinct clusters or uniformly diverse data—for example, it is unlikely that a population of web users will produce well separated clusters of homogeneous behaviour. It is also unlikely that the spread or variation is exactly uniform. Therefore it is not surprising to find that on such data the predictive performance of the two versions of SGTM (with fixed uniform prior or with estimated prior) is not very different from each other. However, in the light of the results above, we expect to achieve equal or higher accuracy using SGTM than the best out of the existing linear models (MMC and SMMC) on data collections of multiple sequences.

6 Application to Preference Prediction and Exploratory Analysis of Web Navigation Sequences

The organisation and exploratory analysis of the dynamic behaviour of individuals in the context of web environments is a major challenge for automated data analysis research. Such investigations are quite recent (Cadez et al. 2003; Girolami and Kabán 2005) and motivated by the availability of vast quantities of user traces and the opportunity for creating predictive profiles as well as creating tools that allow e.g. a site administrator to explore large sets of navigation sequences. The possibility of visual exploration in this context has been proposed in (Cadez et al. 2003), where an approach employing mixture based clustering of first order Markov chains has been explored.

However, in a mixture model, the relation between clusters is not modelled, and in the case of a large site collection, with several thousands of browsing users, it would be impractical to expect the site administrator to examine all clusters individually in order to obtain an overview of the ongoing activity or to locate behaviours of interest. In addition, browsing behaviours that are common to all clusters of users, and are therefore less interesting, will end up being present on all cluster prototypes, making the visual analysis difficult. Indeed, in the mentioned work, such problems have been noticed and the ad-hoc constraint of fixing the initial state in each cluster has been employed in order to aid visual inspection. This of course came at the expense of a suboptimal predictive model as reflected by the out of sample log likelihood. On other hand, previous work reported in (Girolami and Kabán 2005) suggests that a distributed model may in fact describe a collection of heterogeneous behaviours more realistically.

Here we investigate our approach for organising the same set of web navigation sequences, `msnbc.com`,[4] used in (Cadez et al. 2003), the subset investigated in (Girolami and Kabán 2005), as well as a number of comprehensive intermediate settings. The data set comprises over a million of sequences that share a common state space of the following 17 page categories: 'frontpg', 'news', 'tech', 'local', 'opinion', 'onair', 'misc', 'weather', 'msnnews', 'health', 'living', 'business', 'msnsport', 'sports', 'summary', 'bbs' and 'travel'. The vast majority of these sequences is very short—the average sequence length was found as 8.056. So in the light of the controlled empirical study of the previous section, we may expect with a random sample to find ourselves in the situation of a 'large' number 'short' sequences, in principle.

We begin with assessing the prediction and generalisation performance of our model. To this end, we constructed both selected subsets of rich sequences and random subsets of various sizes. These are summarised in Table 1.

We report 10-fold cross-validated predictive perplexity results on the first five data sets listed in Table 1, since these contain a relatively small sample size. The predictive perplexity measures the uncertainty of predictions and is computed as $\exp\{-\frac{1}{N_{\text{test}}}\sum_{r=1}^{N_{\text{test}}}\log P(s_{\text{next}}|S_r)\}$ (lower values are better). The results of MMC, SMMC and first-order global Markov chains are shown on Fig. 9. The graph for WEB9 is identical to that reported in (Girolami and Kabán 2005), and usefully serves as a basis for the comparisons. From Fig. 9 we can see that by increasing the training set size, all methods improve their performance to some extent. However, MMC benefits more from increasing the number of sequences available for training whereas, SMMC benefits more from the length (richness) of the sequences. The

Table 1 Data sets constructed from `msnbc.com`, used in the reported experiments: WEB9 and WEB7 are selected 'rich' sequences, which includes only users who visited at least 9 (or 7 respectively) out of the overall 17 different page categories. The remainder are randomly chosen subsets. $\text{WEB}_{\text{train}}$ and WEB_{test} are training and independent test sets of the size used by (Cadez et al. 2003)

Name	Nr sequences	Nr transactions	Avg length
WEB9	1480	119 667	80.856
WEB7	5800	246 360	42.476
WEB-1500	1500	20 799	13.866
WEB-5000	5000	51 665	10.333
WEB-10000	10 000	96 384	9.638
$\text{WEB}_{\text{train}}$	100 000	801 745	8.018
WEB_{test}	88 181	714 280	8.1

[4] http://kdd.ics.uci.edu/databases/msnbc/msnbc.html.

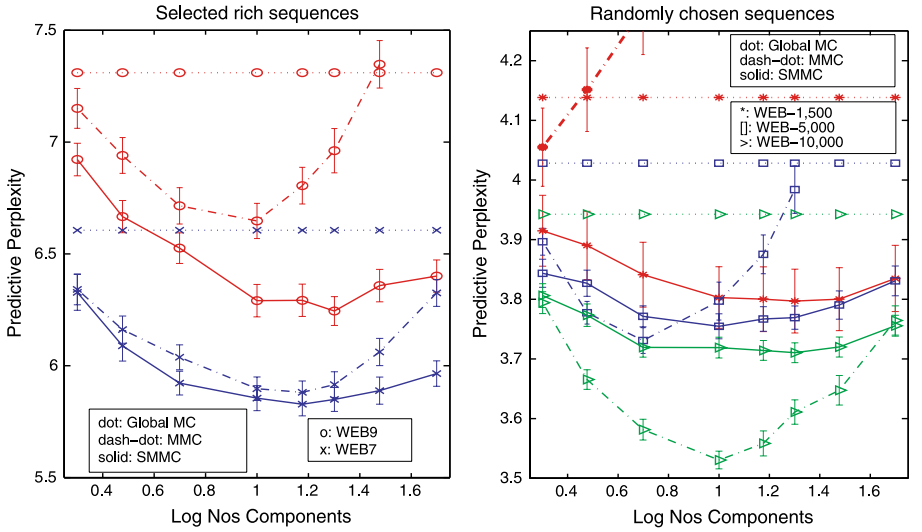

Fig. 9 Comparative predictive perplexity results of MMC and SMMC on weblog sequence collections from `msnbc.com`, of various sizes. SMMC benefits more from 'richer' sequences whereas MMC benefits more from a larger number of sequences. For each data collection under study, the best performing method is taken over to Fig. 10 to be compared with SGTM

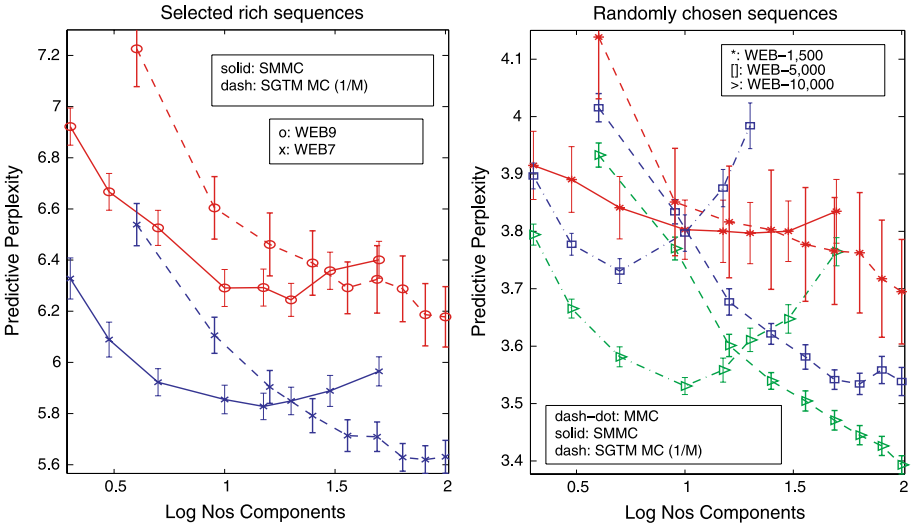

Fig. 10 SGTM compared with the best performing previous method, for each of the five sequence collections tested. SGTM significantly outperforms the previous winner on each of these data sets

differences were found statistically significant at the 5% level based on the nonparametric rank sum test, and these results are in accordance with our findings on the synthetic simulations, presented in the previous section. More interestingly, Fig. 10 shows the predictive perplexity results of SGTM in comparison with the best performing method retained from Fig. 9, for each data set under consideration. Clearly, SGTM outperforms the previ-

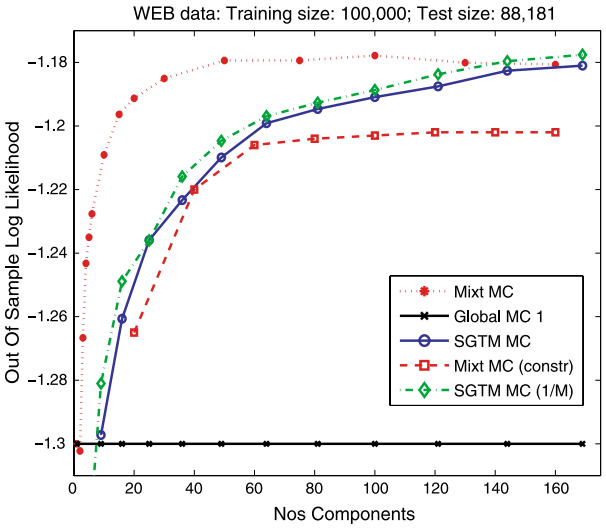

Fig. 11 Out of sample log likelihood against model order obtained by SGTM estimated from $\text{WEB}_{\text{train}}$ and tested on WEB_{test}. Both versions of SGTM outperform the constrained mixture employed in (Cadez et al. 2003) and are at least as good as an unconstrained mixture

ous winners in all situations tested, in a statistically significant manner. The two variants of SGTM—employing a fixed uniform prior or estimating the mixing coefficients—have performed similarly on these data (no statistically significant differences), and the former is shown on the plots.

Finally to compare against the results of (Cadez et al. 2003), we have taken the training set of 100 000 sequences $\text{WEB}_{\text{train}}$, drawn at random from the entire data set, totalling 801 745 page requests and the independent test set WEB_{test}, of 88 181 sequences totalling 714 280 page requests. Since the vast majority of these sequences is very short, the SMMC overfits immediately and is not shown on the figure. Figure 11 depicts the out of sample log likelihood as obtained on the independent test set comparatively for SGTM, the constrained mixture of (Cadez et al. 2003) (fixing the initial state in each cluster to aid visualisation), an unconstrained mixture and a baseline global first order Markov model. Clearly, our method outperforms the mixture with constrained initial states of (Cadez et al. 2003) and approaches an unconstrained mixture in terms of predictive performance. We can thus be confident that the advantages of our model in terms of visualisation and parameter interpretability do not produce a limitation of its predictive power on this data. Our model requires more components than MMC, though, since we have seen it is a constrained mixture. This is because unlike unconstrained mixtures, it allows us to create detailed topographically ordered summary mappings of the data collection and these may be used for exploratory analysis.

We now demonstrate that our model also creates a meaningful, topographically ordered visual summary of the recorded dynamic activity, and is therefore more convenient to use by e.g. a site administrator, against an unconstrained mixture. Figure 12 shows the full map of sequences created from the 100 000 sized data set. For equidistant points on the latent space, the 15 highest probability sequences are shown in colour-coding, where each colour stands for one page category. The topographical principle that is induced, originally proposed by Kohonen (Kaski et al. 1998), provides a proximity constraint that has proved intuitive and useful in hundreds of applications in the past (Kaski et al. 1998). Indeed, our eyes are sensitive not just to individual colours but also to reasonably low-entropy patterns or textures, therefore our hope is that visualising temporal activity in terms of proximity structures may be useful.

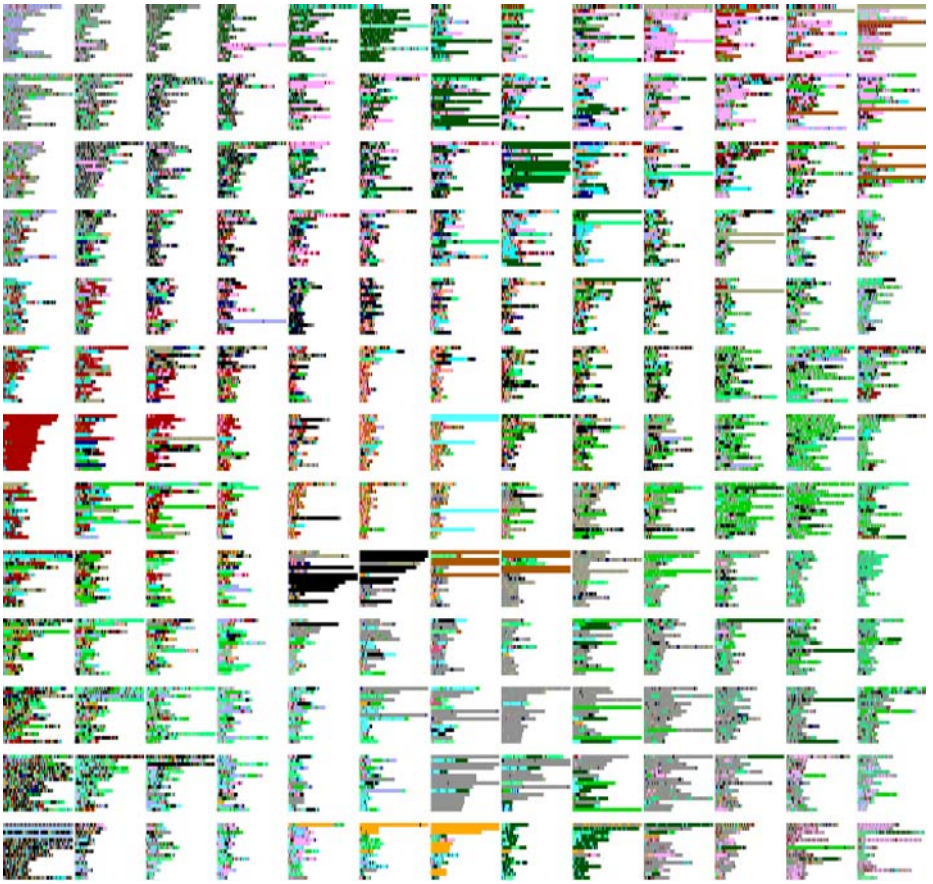

Fig. 12 Topographical display obtained from a random sample of 100 000 sequences from the `msnbc.com` web navigation data set. The top matching sequences are shown at equidistant points of the latent space of $\boldsymbol{x}$. for each prototype. Sequences are in *rows*, and the *colors* encode symbols (page categories). *Blank lines* separate the sequences of neighboring latent point locations

In addition, Fig. 13 shows a fragment (three columns) of the component-level representation created. On the right, the probability transitions associated with the model parameter components are shown. As discussed in Sect. 3, these are low-entropy components of behaviour, different from cluster prototypes (the latter being local averages). On the left, the top matching 15 actual user sequences are listed for each aspect. Using these, we can follow a gradual shift of interest on this fragment of the representation. E.g. from top toward the bottom, the strong interest in the 'frontpage' of the site (1-st page category) shifts through a repetitive user behaviour toward a pronounced interest in 'news' (2-nd page category), corroborated with a more dynamic browsing activity. On the horizontal axis in the first row the interest shifts from 'frontpage' to 'sports' and 'health'. These trends can be more easily followed by looking at the transition plots. However, from the listing of the actual sequences we can see how represented, how homogeneous or inhomogeneous these behavioural components are, and we can recognise groups of similar behaviours by the specific combinations of patterns and colours. The topographic organisation is most apparent in both views.

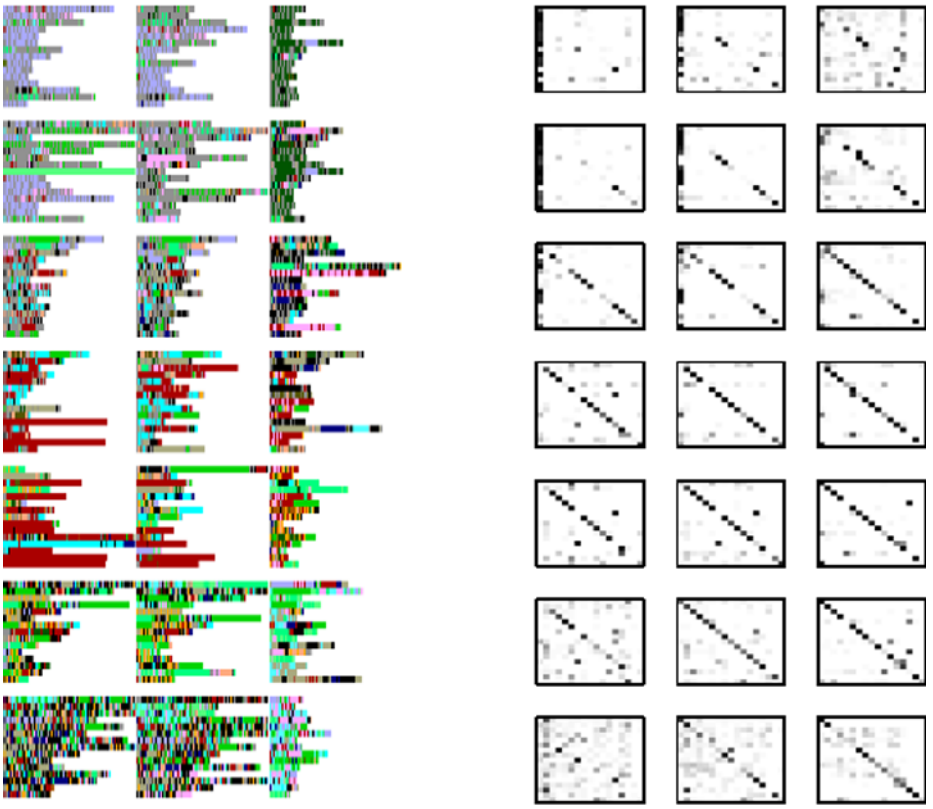

Fig. 13 A fragment from the component-level topographical display of the same `msnbc.com` web navigation data set. The top matching sequences are shown for each component on the *left*. On the *right*, *grey levels* encode the associated transition probabilities between the 17 page categories. *Darker* stands for higher probability

It should also be noted that, as expected, the prototype-level transition behaviours are far not as informative. This is simply because behavioural patterns that are common to all clusters appear on all prototypes, making it difficult to distinguish the distinctive features. This is a problem for the mixture-based visualisation method of (Cadez et al. 2003). To illustrate this, a fragment of the mixture-level transitions is shown on Fig. 14.

The habit of repeating the previous page category request is present everywhere. We thus conclude that the entropy-minimising characteristic in our model is quite important for parameter interpretability. It is a unique feature of the proposed model that it is able to produce such low-entropy components of the data, simultaneously with a 2D nonlinear compression, while additionally also being a well-performing predictive model, able to generalise to new, previously unseen data—as demonstrated by both the out of sample likelihood values and the predictive perplexity measures.

Finally, our model also induces probabilistic profiles for each individual sequence, in the form of two posterior distributions (over the latent space of cluster prototypes and over the space of low entropy components respectively). These may be used to understand some of the relationships between users. Figure 15 shows three examples. In the first example, a fairly long activity sequence induces a relatively sharp cluster-posterior. In the second

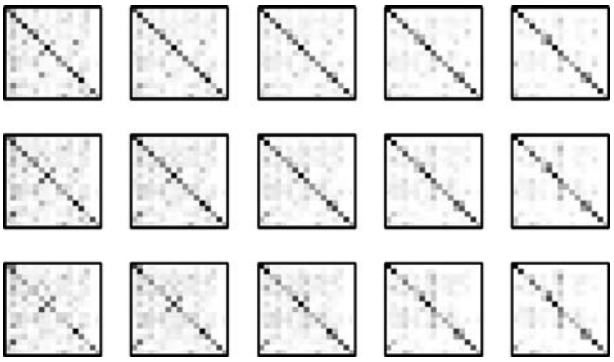

Fig. 14 Fragment from the prototype-level transitions

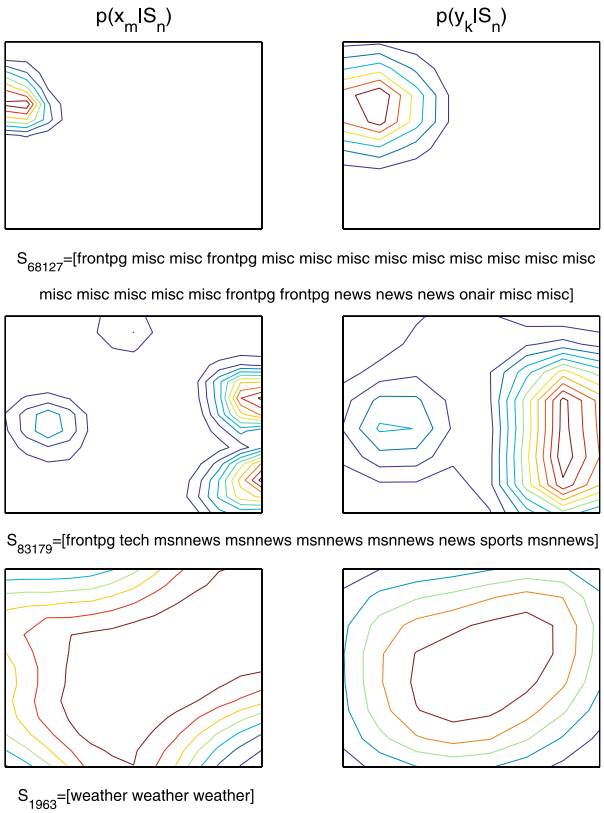

Fig. 15 Three individual user profiles over the two latent spaces

example we have a multi-modal profile that encodes more interesting relationships based on multiple interests. In both these examples, the component-level posterior produces smoothed versions of the cluster-level posterior. In the third example in turn we have a very common activity, therefore the cluster-posterior is very broad. By contrary, the component-posterior shrinks—this is because the model parameters tend to separate common behaviours into fewer components.

The computation time required to convergence is now assessed experimentally. Subsets of data with increasing size have been created and the CPU time has been measured against

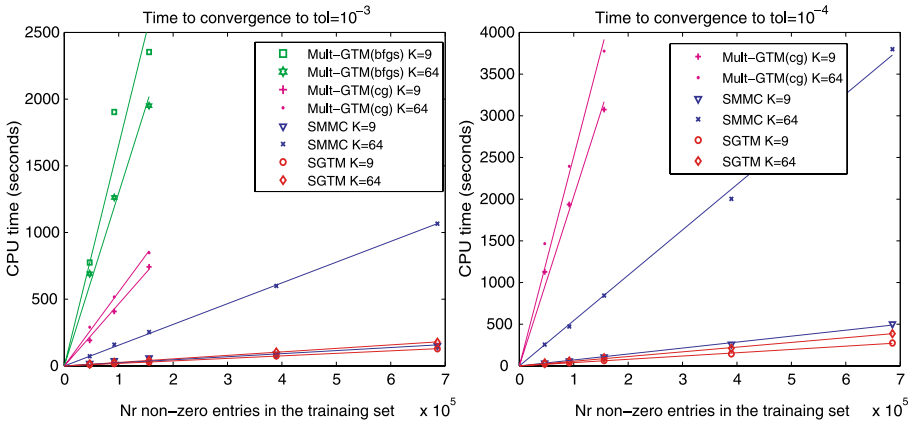

Fig. 16 Experimental comparison of the computation time to convergence to a tolerance of 10^{-3} (*left*) and 10^{-4} (*right*), in varying data size conditions

the number of non-zero entries in the data, on a standard desktop computer (Intel Pentium 2 GHz). The tolerance criterion used was the difference between consecutive log likelihood values. Figure 16 shows the results comparatively, for the tolerance of 10^{-3} and 10^{-4} respectively. The competing methods in this comparison are SGTM, SMMC and multinomial Latent Trait GTM. For the latter, the efficiency of the M-steps depends on the nonlinear optimisation used and their parameters. (The E-steps enjoy the same scaling as in SGTM.) We skip showing results from IRLS and gradient ascent, for poor efficiency. Instead, to be as fair as possible, for this comparison we implemented more efficient versions: A version uses partial M-steps employing Broyden–Fletcher–Goldfarb–Shanno (BFGS) optimisation (Kelley 1995) with polynomial line search,[5] where the number of inner iterations was set to 5 (determined empirically). A second version employs partial M-steps computed by conjugate gradients together with a fairly sophisticated set of searches;[6] again the number of line searches was set to 5. In both implementations, we also took advantage of that the product $\boldsymbol{DR}^T$ in (28) only needs evaluated once before entering into the numerical optimisation routine. Still, the time taken to convergence is far longer than for the competing methods.

The results are shown in Fig. 16. In all cases, two model orders were tested ($K = 9$ and $K = 64$) and the number of 2D latent samples was fixed to 100 throughout. The markers show CPU times averaged over 10 independent random restarts, and the straight lines represent the regression lines fitted to the set of outcomes (and passing through the origin of course), for each method. As expected, our SGTM method, similarly to SMMC, scales indeed linearly with the number of non-zero entries in the data. The plot also shows that there is little increase in computation time when increasing the model order K in our model. This is partly because there is little variation in the level of sparsity of the matrix $\boldsymbol{\Phi}$ when K is varied. We also observed that the average number of iterations required to convergence stayed roughly the same when varying either the model order or the data set size, for SGTM. (For the largest data set, SGTM needed 25.1 EM loops in average for achieving the tolerance

[5]Matlab routine by Kelley, T.C., available from: http://www.siam.org/books/kelley/fr18/matlabcode.php.

[6]Matlab routine by Rasmussen, C.E., available from: http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize/.

of 10^{-3} and 72.3 EM loops for the tolerance of 10^{-4}.) Overall, at both model orders considered, and with both tolerance thresholds, the scaling of our method was comparable to that of a SMMC with 9 components. SMMC in turn requires more time to converge when the model order increases. Clearly, the multinomial GTM has been the most computationally demanding method in this comparison (despite the optimisation method being chosen with care)—which means, this model is essentially limited to relatively small or medium size data sets in practice. This is partly because the M-step computations are longer due to the search for step sizes, and partly because the number of EM loops needed for convergence to a given tolerance was significantly larger—in average, the EM loops needed were almost 4 times larger than those needed by SGTM for the tolerance of 10^{-3}, and more than 5 times larger for the tolerance of 10^{-4}. More work would be needed to develop more efficient algorithms for this model before we could make realistic comparisons, in terms of generalisation and prediction ability, between the natural parameter based modelling of multinomial GTM with that of the mean parameter based modelling of SGTM.

7 Conclusions

In this paper we presented a theoretically principled, computationally efficient and intuitively simple topographic generative model for sparse symbolic sequence collections. Besides being stable and scalable to large and sparse data sets, the proposed approach is robust against finite sample sizes, and improves prediction in comparison with the state of the art, on both synthetic and real-world data. It is also able to create a compact compression of the histories, in a locally topology-preserving manner, which is useful for visualisation and exploratory analysis. In addition, the model parameters are also interpretable probabilities, which may be understood as low-complexity component models of the data collection.

We have discussed the relationship of our model with a number of related topographic approaches, we have analysed its representation tendency towards low-entropy parameters, we have empirically assessed its predictive performance in comprehensive comparisons and we have demonstrated an application of our approach to the prediction and exploratory analysis of large real-world web navigation sequence collections. Our nonlinear model has been able to outperform the state of the art in all experimental settings in terms of predictive modelling, and at the same time has revealed simple intuitive structures behind the apparently high-entropy activity recordings. Further work may include a formal analysis of the sample size requirement properties, a more detailed analysis of the convergence speed e.g. following (Salakhutdinov et al. 2003; Celeux et al. 2001), algorithmic extensions to models with deeper memory, as well as possibly applications to other areas, such as for example, model based multi-task reinforcement learning.

Appendix

This appendix provides the details of obtaining the discrete assignment of latent points associated with the objective (23) of Sect. 3. As highlighted in the main text, the first and third of the terms of this expression are negative weighted sums of entropies, and the second term contains a sum of divergences between the distribution over the symbol-level latent generators and the pre-wired neighbourhood probability distribution of these generators, relative

to sequence-specific latent points. The discrete optimisation objective associated with (23) is the following.

$$E(\delta_{mn}, \delta_{kmij}) = \sum_{k,j}\left(\sum_{i',n,m} \delta_{mn}\delta_{kmi'j} N_{ij}^n\right) \sum_i a_{ik}^j \log a_{ik}^j + \sum_m \left\{\sum_k \left[\sum_{n,i,j} \delta_{mn}\delta_{kmij} N_{ij}^n\right] \log \phi_k(\boldsymbol{x}_m)\right\} + N \sum_m \alpha_m \log \alpha_m$$

subject to:

$$\sum_m \delta_{mn} = 1, \quad \forall n; \qquad \sum_k \delta_{kmij} = 1, \quad \forall m, i, j,$$

$$\delta_{mn}, \delta_{kmij} \in \{0, 1\}, \quad \forall m, n, k, i, j,$$

$$a_{ik}^j = \frac{\sum_n \sum_m \delta_{mn}\delta_{kmij} N_{ij}^n}{\sum_{i'} \sum_n \sum_m \delta_{mn}\delta_{kmi'j} N_{i'j}^n}, \quad \forall i, k,$$

$$\alpha_m = \frac{1}{N} \sum_n \delta_{mn}, \quad \forall m.$$

To solve this by the mean field approach, the following objective needs to be maximised (called the free energy function) (Wu and Chiu 2001; Peterson and Söderberg 1989).

$$\begin{aligned}\psi(r_{mn}, r_{kmij}, e_{mn}, e_{kmij}) &= -E(r_{mn}, r_{kmij}) + \sum_{mn} r_{mn} e_{mn} - \frac{1}{\beta} \sum_n \log \sum_m \exp(e_{mn}) \\ &\quad + \sum_{kmij} r_{kmij} e_{kmij} - \frac{1}{\beta} \sum_{mij} \log \sum_k \exp(e_{kmij}) \end{aligned} \tag{38}$$

where $r_{mn} = P(\delta_m n = 1)$ and $r_{kmij} = P(\delta_{kmij} = 1)$ are the probabilities of the discrete assignments and $\{e_{mn}\}$ and $\{e_{kmij}\}$ are two sets of dummy auxiliary variables used for carrying out the mean field optimisation, and β is the so-called inverse temperature, which may be increased during the iterations as in simulated annealing. We disregarded this parameter here, setting it to one, since our scope for now is mainly to show the connection to our EM algorithm previously derived.

The stationary point of (38) gives the following equations:

$$\frac{\partial \psi(r_{mn}, r_{kmij}, e_{mn}, e_{kmij})}{\partial r_{mn}} = 0 \Rightarrow e_{mn} = \frac{\partial E(r_{mn}, r_{kmij})}{\partial r_{mn}}, \tag{39}$$

$$\frac{\partial \psi(r_{mn}, r_{kmij}, e_{mn}, e_{kmij})}{\partial e_{mn}} = 0 \Rightarrow r_{mn} = \frac{\exp(e_{mn})}{\sum_{m'} \exp(e_{m'n})} \tag{40}$$

and similarly,

$$\frac{\partial \psi(r_{mn}, r_{kmij}, e_{mn}, e_{kmij})}{\partial r_{kmij}} = 0 \Rightarrow e_{kmij} = \frac{\partial E(r_{mn}, r_{kmij})}{\partial r_{kmij}}, \tag{41}$$

$$\frac{\partial \psi(r_{mn}, r_{kmij}, e_{mn}, e_{kmij})}{\partial e_{kmij}} = 0 \Rightarrow r_{kmij} = \frac{\exp(e_{kmij})}{\sum_{k'} \exp(e_{k'mij})} \tag{42}$$

where the constant terms were ignored from (39) and (41), since they cancel in (40) and (42) respectively. Detailing the above equations we obtain (6) & (18) identical to the E-step of the EM algorithm derived in Sect. 3 of the main text. Further, using the constraints amounts exactly to the M-step equations (21) & (10). In consequence, the iterative mean-filed solution of the above constrained discrete optimisation problem yields exactly the same algorithmic solution as the EM iterations (6) & (18) & (21) & (10) given in the text.

References

Attias, H. (2001). Learning in high dimensions: Modular mixture models. In *Proceedings of the 8th International Conference on Artificial Intelligence and Statistics* (pp. 144–148).

Bengio, Y., Paiement, J.-F., & Vincent, P. (2004). *Neural information processing systems (NIPS): Vol. 16. Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering*. Cambridge: MIT Press.

Bernardo, J. M., & Smith, A. F. M. (2001). *Bayesian theory*. Cambridge: Wiley.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford: Oxford University Press. Chap. 7

Bishop, C. M., Svensen, M., & & Williams, C. K. I. (1998a). The generative topographic mapping. *Neural Computation*, *10*(1), 215–234.

Bishop, C. M., Svensen, M., & Williams, C. K. I. (1998b). Developments of the generative topographic mapping. *Neurocomputing*, *21*, 203–224.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*(5), 993–1022.

Buntine, W. (2002). Variational extensions to EM and multinomial PCA. In *Proc. of the 13-th European Conference on Machine Learning (ECML)*.

Cadez, I., Heckerman, D., Meek, C., Smyth, P., & White, S. (2003). Model-based clustering and visualisation of navigation patterns on a web site. *Data Mining and Knowledge Discovery*, *7*(4), 499–242.

Carreira-Perpiñán, M. Á., & Renals S. (1998). Experimental evaluation of latent variable models for dimensionality reduction. In *Proc. IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing (NNSP'98)* (pp. 165–173).

Celeux, G., Chrétien, S., Forbes, F., & Mkhadri, A. (2001). A component-wise EM algorithm for mixtures. *Journal of Computational & Graphical Statistics*, *10*(4), 697–712(16).

Cover, T., & Thomas, J. (1991). *Elements of information theory*. New York: Wiley.

Girolami, M., & Kabán, A. (2005). Sequential activity profiling: Latent Dirichlet allocation of Markov chains. *Data Mining and Knowledge Discovery*, *10*, 175–196.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York: Springer.

Hofmann, T. (2000). Probmap—A probabilistic approach for mapping large document collections. *Journal for Intelligent Data Analysis*, *4*, 149–164.

Hofmann, T., & Buhmann, J. (1998). Competitive learning algorithms for robust vector quantization. *IEEE Transactions on Signal Processing*, *46*(6), 1665–1675.

Hollmén, J., Tresp, V., & Simula, O. (1999). A self-organizing map algorithm for clustering probabilistic models. In *Proc. of the 9-th International Conference on Artificial Neural Networks (ICANN):, Vol. 2* (pp. 946–951).

Iwata, T., Saito, K., Ueda, N., Stromsten, S., Griffiths, T. L., & Tennenbaum, J. B. (2005). *Neural information processing systems (NIPS): Vol. 17. Parametric embedding for class visualisation*. Cambridge: MIT Press.

Kabán, A. (2005). A scalable generative topographic mapping for sparse data sequences. In *Proc. IEEE International Conference on Information Technology: Coding and Computing (ITCC)* (pp. 51–56).

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

Kaski, S., Kangas, J., & Kohonen, T. (1998). Bibliography of self-organizing map (SOM) papers: 1981–1997. In *Neural computing surveys: Vol. 1* (pp. 102–350).

Keller, M., & Bengio, S. (2004). TTMM: a graphical model for document representation. In *PASCAL Workshop on Text Mining and Understanding*, Grenoble, France

Kelley, T. C. (1995). *Iterative methods for optimization. Frontiers in Applied Mathematics*, Philadelphia: SIAM.

Kohonen, T. (1999). *Self-organising maps*. Berlin: Springer.
McLachlan, G., & Krishnan, T. (1997). *The EM algorithm and extensions*. New York: Wiley.
Peterson, C., & Söderberg, B. (1989). A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, *1*(1), 3–22.
Ramakrishnan, N., & Grama, A. (2001). Mining scientific data. *Advances in Computers*, *55*, 119–169.
Roweis, S., Saul, L. K., & Hinton, G. (2002). *Neural information processing systems (NIPS): Vol. 14*. *Global coordination of local linear models* (pp. 889–896). Cambridge: MIT Press.
Salakhutdinov, R., Roweis, S., & Ghahramani, Z. (2003). Optimization with EM and expectation-conjugate-gradient. In *Proc. of the 20-th International Conference on Machine Learning (ICML)* (pp. 672–679).
Sarukkai, R. (2000). Link prediction and path analysis using Markov chains. *Computer Networks*, *33*(1–6), 377–386.
Tiňo, P., Kabán, A., & Sun, Y. (2004). A generative probabilistic approach to visualising sets of symbolic sequences. In *Proc. of the 10-th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 701–706). New York: ACM.
Wu, J. M., & Chiu, S. J. (2001). Independent component analysis using Potts models. *IEEE Transactions on Neural Networks*, *12*(2), 202–211.