GTM Mixture through time for sequential data

Rakia Jaziri, Faicel Chamroukhi, Mustapha Lebbah, Younès Bennani

***Abstract*—Generative Topographic Mapping (GTM) is a popular probabilistic framework for modeling non-linear relationships in high-dimensional data as well as for unsupervised learning and visualization of such data. It is also known as to provide a principled probabilistic alternative to the well-known Self-Organizing Map (SOM) in the neural networks community, thanks to its flexible mixture model formulation and the desirable properties of the expectation-maximization (EM) algorithm. However, much attention has been focused on the use of GTM for multivariate data, in general assumed to be independent and identically distributed (i.i.d) and the problem of modeling sequences using GTM is less investigated. In this paper, we focus on GTM for unsupervised modeling and visualization of sequential data. We consider modeling sequences of continuous multidimensional observations and we propose a GTM through time (GTM-TT) approach based on hidden Markov models (HMM) where the observations are a sent of independent sequences, rather than a signle sequence. We further extend the model to the clustering of multiple sequences by proposing a GTM-TT mixture model. The model parameters are estimated by maximum likelihood via the EM algorithm. The proposed approach is evaluated using simulated data and real-world data.**

I. Introduction

Unsupervised modeling of non-linear relationships in high-dimensional data, as well as dimensionality reduction and visualization of such data is an important topic in machine learning and data analysis. Topographic approaches, in particular the self-organizing map (SOM) [10], are popular in the neural networks community thanks to their well-established bio-inspired framework. The SOM idea consists in an unsupervised learning approach based on artificial neural networks and was inspired from the competitive learning. Competitive learning [11] is an unsupervised adaptive process during which the neurons of a neural network (units) compete for the right to respond to a subset of the input data and each unit of the network gradually becomes specialized of a subset of the data. When an input is presented, the neuron that is best to represent it in the sense of a chosen similarity measure, typically an Euclidean distance, wins the competition and is allowed to learn from it; this is the well-known “winner-take-all" rule. In the standard formulation of competitive learning, only the winner neuron (also called the best matching unit (BMU)) is updated and the approach does not consider the order between the neurons.

The SOM [10], which is an unsupervised neural approach for the exploration and visualization of high-dimensional data, generalizes the competitive learning by allowing also the neighbors of the winner to be updated, which can be seen as a cooperation phase after the competition phase, and for which the neurons become ordered on a map lattice. The BMU or prototype, that is, the winner of the competition, is updated, as well as its neighbors. The neighbors of the BMU are also updated in a weighted manner in the cooperation phase according to a chosen neighborhood function around the winner unit. The neighborhood function can for example be a Gaussian centered at the BMU and having a neighborhood width σ which decreases monotonically as the learning proceeds. Due to the neighborhood function, the units which are closer to the BMU will be more affected than the others.

The SOM thus derives an orderly mapping of multidimensional data onto a regular typically 2-dimensional map and coverts complex nonlinear relationships in the high-dimensional space into simpler relationships in the plan (map). The important topological and metric relationships are conveyed in this non-linear projection and the data are organized on the map in such a way that observations that are close together in the high-dimensional data space are also closer to each other on the map (projection space). After the training step, we have the set of prototypes over the 2-d coordinates on the map. For a clustering purpose, a partition of the data can be computed by running a standard clustering algorithm (e.g., K-means) on the obtained prototypes.For visualizing the resulting map and prototypes, one can cite for example the U-matrix [16] that is often used.Another non-linear dimensionality reduction method, also used for unsupervised exploratory data analysis and data visualization is the Multidimensional Scaling [7]. We note that while the SOM can be seen as an unsupervised learning algorithm that (stochastically) minimizes a cost function (e.g., [9]), at the origin, the algorithm is based on heuristics and is not derived from the optimization of an objective function. In addition, the preservation of the neighborhood structure is not guaranteed by the SOM method. The SOM does not define a density model, the choice of how the neighborhood function should shrink during the learning process is also sensitive.

The Generative Topographic Mapping (GTM) [3], which is a popular probabilistic framework for modeling non-linear relationships in high-dimensional data as well as for unsupervised learning and visualization of such data, provides a principled probabilistic alternative to the SOM. The GTM was indeed inspired by the SOM and attempts to overcome its limitations through a probabilistic formulation. More specifically, the GTM is described in terms of a latent variable (or space) model with dimensionality L [3] where the goal is to find a representation for the distribution $p(\mathbf{y})$ of

Rakia Jaziri is with the Data Science Department - LINCOLN Consulting group. Faicel Chamroukhi is with the Lab of mathematics Paul Painlevé - UMR CNRS 8524 and the Information Sciences and Systems Lab - UMR CNRS 7296. Mustapha Lebbah and Younès Bennani are with The Computer Science Lab of Paris Nord - UMR CNRS 7030

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

d-dimensional data, in terms of a smaller number of L latent variables where $L < d$, typically $L = 2$ for visualization in the 2-d space. Additionally, the model parameters learning is performed by monotically maximizing the observed-data likelihood by using the expectation-maximization (EM) algorithm [8], [12]. Thus, both convergence and topographic ordering are guaranteed with the GTM. In addition, the GTM performs soft clustering in contrary to the SOM which performs hard assignments of the data to the clusters. Comparisons of the GTM to the SOM can be found in [3].

In this paper, we focus on the GTM model due to its well established statistical background and the well-known desirable properties of the EM algorithm [8], namely its stability, convergence and the monotonic increase of the likelihood.

The approach proposed in this paper, which consists in a topographic inspired hidden Markov model for modeling and visualizing sequences of multidimensional continuous observations, is an extension of the GTM Through time model for sequential data [4][13] to a mixture framework. The proposed generative approach assumes therefore that the observation sequence is generated according to a HMM model for which the emission probability density function for each state is a mixture density, each component of the mixture density being a GTM model. This allows to provide a more flexible data modeling, thanks to the flexibility of the mixture modeling framework.

The reminder of this paper is organized as follows. Section II provides an account of the GTM Through Time approach. Section III introduces the proposed generative topographic model for sequential data and its parameter estimation via the EM algorithm. Finally, section IV deals with the experimental study carried out real-world data from a character recognition problem to illustrate the performance of the the proposed approach.

Let $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ be an observation sequence of n multidimensional data vectors $\mathbf{y}_t = (y_{t1}, \ldots, y_{td}) \in \mathbb{R}^d$ regularly observed at the time points $(1, \ldots, n)$. Assume that the observation sequence $\mathbf{Y}$ is generated by a K-state hidden Markov model (HMM) and let $\mathbf{z} = (z_1, \ldots, z_T)$ be the associated unknown (hidden) states with $z_i \in \{1, \ldots, K\}$. Now consider a two-dimensional latent space (the map) $\mathbf{x} = (x_1, x_2)$ on which we aim to visualize the data. In the following we will give an overview of the GTM model for sequential data.

II. The Generative Topographic Mapping (GTM) and GTM Through Time (GTM-TT)

Here we consider the GTM through time (GTM-TT) model proposed in [4] for a single sequence of observations. The GTM-TT model extends the standard GTM model [6][5], which is dedicated to i.i.d data, to learn from sequential data by relaxing the independence assumption. More specifically, the GTM-TT model incorporates the standard GTM model as the emission density of a hidden Markov model (HMM). Let us first recall that the GTM model is a latent data (space) model. The aim of a latent data model is to represent the distribution of the observed data $p(\mathbf{y}_t)$ in the data space $\mathbb{R}^d$ in terms of a number of L-dimensional latent variables $\mathbf{x}$ with distribution $p(\mathbf{x})$. The distribution $p(\mathbf{y})$ is then obtained by integrating over the distribution of $\mathbf{x}$ and by considering a given conditional density $p(\mathbf{y}|\mathbf{x})$, that is, for the ith observation we have

$$p(\mathbf{y}_i) = \int_{\mathcal{X}} p(\mathbf{y}_i|\mathbf{x}_i)p(\mathbf{x}_i)d\mathbf{x}_i. \tag{1}$$

For computational tractability of this integral, the GTM model assumes that the distribution over the latent variables $\mathbf{x}$ (representing the latent space) is a mixture of Dirac distributions with uniform mixing proportions and is given by

$$p(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K}\delta(\mathbf{x} - \mathbf{x}_k) \tag{2}$$

where $\mathbf{x}_k$ represents the coordinates of the Dirac placement on the latent space. In the case of GTM, the latent variables $\mathbf{x}$ generally lie in a two-dimensional space ($L = 2$).

To specify the conditional density of the observations $\mathbf{y}$ on the latent variables $\mathbf{x}$, for the GTM, this is achieved by considering a parametric non-linear mapping function $\mathbf{f}(\mathbf{x};\mathbf{W})$ that maps the latent data $\mathbf{x}$ from the latent space to corresponding projection in the data space. Then, the conditional density of the observations is given as a Gaussian density centered at the projected mean points $\mathbf{f}(\mathbf{x};\mathbf{W})$ with a noise variance β^{-1} (isotropic spherical model):

$$\begin{aligned} p(\mathbf{y}_i|\mathbf{x}_k) &= \mathcal{N}(\mathbf{y}_i; \mathbf{f}(\mathbf{x}_k;\mathbf{W}), \beta^{-1}\mathbf{I}_d) \\ &= \left(\frac{\beta}{2\pi}\right)^{d/2} \exp\left\{-\frac{\beta}{2} \parallel \mathbf{y}_i - \mathbf{f}(\mathbf{x}_k;\mathbf{W}) \parallel^2\right\} \end{aligned} \tag{3}$$

where $\mathbf{f}(\mathbf{x}_k;\mathbf{W}) = \mathbf{W}\mathbf{\Phi}(\mathbf{x}_k)$ is a d-dimensional point in the manifold embedded in data space with $\mathbf{W}$ is a $d \times M$ matrix of parameters that govern the mapping, $\mathbf{\Phi}(\mathbf{x}_k) = (\Phi_1(\mathbf{x}_k), \ldots, \Phi_M(\mathbf{x}_k))$ consists of M non-linear basis functions. In the standard model $\phi_m(\mathbf{x}_k)$ can be a Gaussian given by $\Phi_m(\mathbf{x}_k) = \exp\left\{-\frac{\|\mathbf{x}_k - \mu_m\|^2}{2\sigma^2}\right\}$.

The GTM density (1) therefore is finally given by the following mixture density

$$\begin{aligned} p(\mathbf{y}_i;\mathbf{W},\beta) &= \int_{\mathcal{X}} p(\mathbf{y}_i|\mathbf{x}_i)p(\mathbf{x}_i)d\mathbf{x}_i \\ &= \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(\mathbf{y}_i; \mathbf{f}(\mathbf{x}_k;\mathbf{W}), \beta^{-1}\mathbf{I}_d) \end{aligned} \tag{4}$$

described by the parameters $(\mathbf{W}, \beta)$. The estimation of the GTM parameters $(\mathbf{W}, \beta)$ from and i.i.d data sample is performed by maximizing the observed-data likelihood

$$p(\mathbf{Y};\mathbf{\Psi}) = \prod_{t=1}^{T}\frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(\mathbf{y}_i; \mathbf{f}(\mathbf{x}_k;\mathbf{W}), \beta^{-1}\mathbf{I}_d). \tag{5}$$

The usual tool in this case is the EM algorithm (see [6], [5]).

The independence assumption becomes however very restricting when the data are organized in sequences. The GTM Through Time (GTM-TT) model [4] relaxes this independence assumption by considering a hidden Markov model (HMM), which is the usual framework to take into account sequential aspect in the data, for which the GTM model is taken as the emission density for the observed data sequence. We describe this model and then consider it for a set of sequences. We assume that the heterogeneous sequence of observed data $(\mathbf{y}_1,\dots,\mathbf{y}_n)$ is governed by the latent space $\mathbf{x}$, in which we possibly aim to visualize the observed multidimensional sequential data, and a latent structure described by variables $(z_1,\dots,z_n)$ which characterises the heterogeneity in the data, in the form of states. Formally, in this case of sequential data, the hidden state sequence $(z_1,\dots,z_n)$ is a Markov chain with initial distribution π and a transition matrix $\mathbf{A}$ where $\pi_k = p(z_1 = k)$ and $\mathbf{A}_{\ell k} = p(z_t = k|z_{t-1} = \ell)$. The conditional emission density function is the one of a GTM model, that is, given the state z_t, the conditional distribution of the tth observation, for example the observation at time t in the case of time series, is given by the following Gaussian as in the case of the GTM

$$p(\mathbf{y}_t|\mathbf{x}_{z_t}) = \left(\frac{\beta}{2\pi}\right)^{d/2} \exp\left\{-\frac{\beta}{2} \parallel \mathbf{y}_t - \mathbf{f}(\mathbf{x}_{z_t};\mathbf{W}) \parallel^2\right\}$$

where z_t denotes the state at time t. The model parameters $\mathbf{\Psi} = (\boldsymbol{\pi}, \mathbf{A}, \beta, \mathbf{W})$ are estimated by maximizing the observed data likelihood, which is expressed as the one of a standard HMM:

$$p(\mathbf{Y};\mathbf{\Psi})=\sum_{z_1}\dots\sum_{z_n} p(z_1)p(\mathbf{y}_1|\mathbf{x}_{z_1})\prod_{t=2}^{T} p(z_t|z_{t-1})p(\mathbf{y}_t|\mathbf{x}_{z_t}). \tag{6}$$

The maximization is performed by the EM (Baum-Welch) algorithm [4][8][2] where the E-step includes a forward-backward recursion to evaluate the posterior state distribution and to compute the likelihood. The GTM-TT model have also been considered more recently in [13].

Note that the models in [4][13] are derived for a single sequence and can be easily extended to the case of an independent set of sequences, which is also considered here. In this case, on may maximized the following observed data log-likelihood $\log\prod_{i=1}^{n} p(\mathbf{Y}_i;\mathbf{\Psi})$, n being the number of sequences. Then the EM algorithm can be directly used to maximized the resulting log-likehood, in a very similar way as the EM algorithm in [4][13].

III. A GTM Mixture Through Time (GTMM-TT)

In this section, we model the observation density by a GTM mixture rather then a single GTM to allow capturing more complex data distributions. For this GTM Mixture Through Time approach, we therefore assume that the observed data, at each time step, are generated according to the following GTM mixture model:

$$\begin{aligned}p(\mathbf{y}_t|\mathbf{x}_k)=&\sum_{m=1}^{M} p(h_t = m|\mathbf{x}_k)p(\mathbf{y}_t|\mathbf{x}_k, h_t = m)\\ =&\sum_{m=1}^{M} \alpha_{km}\,\mathcal{N}\left(\mathbf{y}_t;\mathbf{f}(\mathbf{x}_k;\mathbf{W}_m),\beta_m^{-1}\mathbf{I}_d\right)\\ =&\sum_{m=1}^{M} \alpha_{km}\left(\frac{\beta_m}{2\pi}\right)^{d/2} \exp\left\{-\frac{\beta_m}{2} \parallel \mathbf{y}_i - \mathbf{f}(\mathbf{x}_k;\mathbf{W}_m) \parallel^2\right\}\end{aligned} \tag{7}$$

where the α_{km}'s represent the non-negative mixture proportions that sum to one of GTM component m for state k. The model parameters given by

$$\mathbf{\Psi} = (\boldsymbol{\pi}, \mathbf{A}, \alpha_1,\dots,\alpha_{KM}, \mathbf{W}_1,\dots,\mathbf{W}_M, \beta_1,\dots,\beta_M)$$

.

A. Maximum likelihood parameter estimation

are estimated by maximizing the observed data likelihood

$$p(\mathbf{Y};\mathbf{\Psi})=\sum_{z_1}\dots\sum_{z_n} p(z_1)p(\mathbf{y}_1|\mathbf{x}_{z_1})\prod_{t=2}^{T} p(z_t|z_{t-1})p(\mathbf{y}_t|\mathbf{x}_{z_t}) \tag{8}$$

by using the EM algorithm. To specify the EM scheme, the complete-data likelihood for the proposed model is stated as:

$$\begin{aligned}p(\mathbf{Y},\mathbf{z};\mathbf{\Psi})=&p(\mathbf{z};\boldsymbol{\pi},\mathbf{A})p(\mathbf{Y}|\mathbf{z};\mathbf{\Psi})\\ =&p(z_1;\boldsymbol{\pi})\prod_{t=2}^{T} p(z_t|z_{t-1};\mathbf{A})\prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_{z_t};\mathbf{\Psi}_{z_t})\end{aligned} \tag{9}$$

By introducing the binary indicator variables z_{tk} such that $z_{tk} = 1$ if $z_t = k$ (i.e $\mathbf{y}_t$ originates from the kth state at time t) and $z_{tk} = 0$ otherwise, and binary indicator variables h_{tm} such that $h_{tm} = 1$ if $h_t = m$ (i.e $\mathbf{y}_t$ originates from the mth GTM mixture component of the kth state at time t) and $h_{tk} = 0$ otherwise, we get the complete-data log-likelihood:

$$\begin{aligned}\mathcal{L}_c(\mathbf{\Psi})=&\sum_{k=1}^{K} z_{1k}\log\pi_k + \sum_{t=2}^{T}\sum_{k=1}^{K}\sum_{\ell=1}^{K} z_{tk}z_{t-1,\ell}\log\mathbf{A}_{\ell k}\\ &+\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{m=1}^{M} z_{tk}h_{tm}\log[\alpha_{km}\mathcal{N}(\mathbf{y}_t;\mathbf{f}(\mathbf{x}_k;\mathbf{W}_m),\beta_m^{-1}\mathbf{I}_d)].\end{aligned} \tag{10}$$

The EM algorithm starts with an initial parameter $\mathbf{\Psi}^{(0)}$ and alternates between the following two steps until convergence:

a) E-step: This step consists of computing the expected complete-data log-likelihood given the observed sequence and a current parameter estimation $\mathbf{\Psi}^{(q)}$, q being the current iteration. This is the well-known Q-function, which corresponds here to:

$$\begin{aligned}Q(\mathbf{\Psi},\mathbf{\Psi}^{(q)})=&\mathbb{E}\Big[\mathcal{L}_c(\mathbf{\Psi};\mathbf{Y},\mathbf{z})|\mathbf{Y};\mathbf{\Psi}^{(q)}\Big] = Q_\pi(\pi,\mathbf{\Psi}^{(q)}) + Q_{\mathbf{A}}(\mathbf{A},\mathbf{\Psi}^{(q)})\\ &+\sum_{k=1}^{K}\big[Q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha},\mathbf{\Psi}^{(q)}) + \sum_{m=1}^{M} Q_{\mathbf{\Psi}_{km}}(\mathbf{\Psi}_{km},\mathbf{\Psi}^{(q)})\big]\end{aligned} \tag{11}$$

with

$$Q_{\pi}(\boldsymbol{\pi}, \boldsymbol{\Psi}^{(q)}) = \sum_{k=1}^{K} \gamma_{1k}^{(q)} \log \pi_k, \tag{12}$$

$$Q_{\mathbf{A}}(\mathbf{A}, \boldsymbol{\Psi}^{(q)}) = \sum_{t=2}^{T} \sum_{k=1}^{K} \sum_{\ell=1}^{K} \xi_{t\ell k}^{(q)} \log \mathbf{A}_{\ell k}, \tag{13}$$

$$Q_{\alpha_{km}}(\boldsymbol{\alpha}, \boldsymbol{\Psi}^{(q)}) = \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_{tk}^{(q)} \tau_{tkm}^{(q)} \log \alpha_m, \tag{14}$$

$$Q_{\boldsymbol{\Psi}_{km}}(\boldsymbol{\Psi}_{km}, \boldsymbol{\Psi}^{(q)}) = \sum_{t=1}^{T} \gamma_{tk}^{(q)} \tau_{tkm}^{(q)} \log \mathcal{N}(\mathbf{y}_t; \mathbf{f}(\mathbf{x}_k; \mathbf{W}_m), \beta_m^{-1} \mathbf{I}_d). \tag{15}$$

which requires the calculation of the following posterior probabilites;

- $\gamma_{tk}^{(q)} = p(z_t = k | \mathbf{Y}; \boldsymbol{\Psi}^{(q)})$ $\forall t = 1, \ldots, T$ and $k = 1, \ldots, K$ is the posterior probability of the state k at time t given the whole observation sequence and the current parameter estimation $\boldsymbol{\Psi}^{(q)}$,
- $\tau_{tkm}^{(q)} = p(h_t = m | \mathbf{y}_t, \mathbf{x}_{z_t=k}; \boldsymbol{\Psi}^{(q)})$ $\forall t = 1, \ldots, T$, $k = 1, \ldots, K$ and $m = 1, \ldots, M$ is the posterior probability of the mixture component m of state k at time t given observation $\mathbf{y}_t$ and the current parameter estimation $\boldsymbol{\Psi}^{(q)}$,
- $\xi_{t\ell k}^{(q)} = p(z_t = k, z_{t-1} = \ell | \mathbf{Y}; \boldsymbol{\Psi}^{(q)})$ $\forall t = 2, \ldots, T$ and $k, \ell = 1, \ldots, K$ is the joint posterior probability of the state k at time t and the state ℓ at time $t-1$ given the whole observation sequence and the current parameter estimation $\boldsymbol{\Psi}^{(q)}$.

The probabilities γ's and ξ's are computed by the forward-backward procedures as in HMMs. The forward procedure computes recursively the probabilities

$$a_{tk} = p(\mathbf{y}_1, \ldots, \mathbf{y}_t, z_t = k; \boldsymbol{\Psi}), \tag{16}$$

where a_{tk} is the probability of observing the partial sequence $(\mathbf{y}_1, \ldots, \mathbf{y}_t)$ and ending with the state k at time t. It can be seen that the likelihood (8) can be computed after the forward pass as: $p(\mathbf{Y}; \boldsymbol{\Psi}) = \sum_{k=1}^{K} a_{nk}$. The backward procedure computes the probabilities

$$b_{tk} = p(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_n | z_t = k; \boldsymbol{\Psi}) \tag{17}$$

b_{tk} being the probability of observing the rest of the sequence $(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_1)$ knowing that we start with the k at time t. The forward and backward probabilities are computed recursively by the so-called Forward-Backward algorithm ([2]). The posterior probabilities are then expressed in function of the forward backward probabilities as follows [15]:

$$\gamma_{tk}^{(q)} = \frac{a_{tk}^{(q)} b_{tk}^{(q)}}{\sum_{k=1}^{K} a_{tk}^{(q)} b_{tk}^{(q)}} \tag{18}$$

and

$$\xi_{t\ell k}^{(q)} = \frac{a_{t-1,\ell}^{(q)} p(\mathbf{y}_t | \mathbf{x}_{z_t=k}; \boldsymbol{\Psi}^{(q)}) b_{tk}^{(q)} \mathbf{A}_{\ell k}^{(q)}}{\sum_{\ell=1}^{K} \sum_{k=1}^{K} a_{t-1,\ell}^{(q)} p(\mathbf{y}_t | \mathbf{x}_{z_t=k}; \boldsymbol{\Psi}^{(q)}) b_{tk}^{(q)} \mathbf{A}_{\ell k}^{(q)}}. \tag{19}$$

The posterior probabilities τ's are computed as in standard mixture as

$$\tau_{tkm}^{(q)} = \frac{\alpha_{km}^{(q)} \mathcal{N}\left(\mathbf{y}_t; \mathbf{f}(\mathbf{x}_k; \mathbf{W}_m^{(q)}), \beta_m^{-1(q)} \mathbf{I}_d\right)}{\sum_{m\prime=1}^{M} \alpha_{km\prime}^{(q)} \mathcal{N}\left(\mathbf{y}_t; \mathbf{f}(\mathbf{x}_k; \mathbf{W}_{m\prime}^{(q)}), \beta_{m\prime}^{-1(q)} \mathbf{I}_d\right)} \cdot \tag{20}$$

b) M-step: In this step, the value of the parameter $\boldsymbol{\Psi}$ is updated by computing the parameter $\boldsymbol{\Psi}^{(q+1)}$ maximizing the expectation Q with respect to $\boldsymbol{\Psi}$.

The maximization of $Q(\boldsymbol{\Psi}, \boldsymbol{\Psi}^{(q)})$ with respect to $\boldsymbol{\Psi}$ is then performed by separately maximizing $Q_{\boldsymbol{\pi}}(\pi, \boldsymbol{\Psi}^{(q)})$, $Q_{\mathbf{A}}(\mathbf{A}, \boldsymbol{\Psi}^{(q)})$, $Q_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}, \boldsymbol{\Psi}^{(q)})$ and $Q_{\boldsymbol{\Psi}_k}(\boldsymbol{\Psi}, \boldsymbol{\Psi}^{(q)})$ $(k = 1, \ldots, K)$.

Maximizing Q_π with respect to $\boldsymbol{\pi}$ and $Q_{\boldsymbol{\alpha}}$ w.r.t α respectively subject to $\sum_k \pi_k = 1$ and $\sum_m \alpha_m = 1$ consist of constrained optimization problem which is solved using Lagrange multipliers. The values maximizing $Q_{\mathbf{A}}$ corresponds to the expected number of transitions from state ℓ to state k relative to the expected total number of transitions away from state ℓ (see [14]).

Finally, the maximization of $Q_{\boldsymbol{\Psi}_{km}}$ w.r.t $\mathbf{W}_m$ and β_m for $m = 1, \ldots, M$ consists of a weighted variant of the problem of estimating a standard GTM through time model. The updating formulas of the M-Step are given by:

$$\pi_k^{(q+1)} = \gamma_{1k}^{(q)} \tag{21}$$

$$\mathbf{A}_{\ell k}^{(q+1)} = \frac{\sum_{t=2}^{T} \xi_{tk\ell}^{(q)}}{\sum_{t=2}^{T} \gamma_{tk}^{(q)}} \tag{22}$$

$$\alpha_m^{(q+1)} = \frac{1}{\sum_{t=1}^{T} \gamma_{tk}^{(q)} \tau_{tkm}^{(q)}} \sum_{t=1}^{T} \gamma_{tk}^{(q)} \tau_{tkm}^{(q)} \tag{23}$$

$$\mathbf{W}_m^{T(q+1)} = \boldsymbol{\Phi}^T \mathbf{G}_{km}^{(q)} \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{\Gamma}_{km}^{(q)} \mathbf{Y} \tag{24}$$

$$\left(\frac{1}{\beta_m}\right)^{(q+1)} = \frac{\sum_{t=1}^{T} \sum_{k=1}^{K} \gamma_{tk}^{(q)} \tau_{tkm}^{(q)} \parallel \mathbf{y}_t - \mathbf{f}(\mathbf{x}_k; \mathbf{W}_m^{(q+1)}) \parallel^2}{d \times \sum_{t=1}^{T} \sum_{k=1}^{K} \gamma_{tk}^{(q)} \tau_{tkm}^{(q)}} \tag{25}$$

where $\boldsymbol{\Phi}$ is the matrix with elements $\boldsymbol{\Phi}_r(\mathbf{x}_k)$, $\mathbf{G}_{km}^{(q)}$ is the $K \times K$ diagonal matrix whose diagonal elements are the weights $\mathbf{G}_{km}^{(q)} = \sum_{t=1}^{T} \gamma_{tk}^{(q)} \tau_{tkm}^{(q)}$, $\boldsymbol{\Gamma}_{km}^{(q)}$ is the matrix with elements $\gamma_{tk}^{(q)} \tau_{tkm}^{(q)}$ and $\mathbf{Y}$ is the data matrix.

B. Data visualization

For data visualization, as in GTM the posterior state probabilities γ_{tk} and the posterior mixture component probabilities τ_{tkm} for each mixture component m can be used to provide a visualization of the posterior responsibility map for individual data points in the two-dimensional latent space. Another way is to visualize the posterior mode. In this case, since the data are sequential in this context of HMM-based modeling, the optimal state sequence can be obtained with the Viterbi algorithm [**?**].

IV. Experiments

In this section we perform experiments to assess the model for the exploratory analysis of a set of multi-dimensional sequences. The algorithms were written in Matlab. We consider real-world data issued from a real-world character recognition problem.

The data consists of 2858 character samples [17], [1]. captured using a WACOM tablet, where 3 dimensions were kept - x, y, and pen tip force at a frequency of 200Hz.
The data were normalized. Then, only characters with a single 'PEN-DOWN' segment were considered. Character segmentation was performed using a pen tip force cut-off point. The characters have also been shifted so that their velocity profiles best match the mean of the set. Each character sample is a 3-dimensional pen tip velocity trajectory.
The single best state sequence is obtained by using the Viterbi algorithm [15].

Selecting only the best sequence corresponds to the winner-take-all principle of the model. The Viterbi algorithm is used in the present work for choosing the best-matching model sequence. Note that the result should be analyzed in color mode.

Figure 1 shows overlay plot of 131 character samples of the letter "p", and 124 of the letter "q" demonstrating the variation in the data.

A. Data visualisation

We consider a model an 12 x 12 latent states on a map lattice. In order to show topological organisation advantage, we run separately the model on 4 data sets : a-data set, d-data set and g-data set.

Figure 2 shows reconstruction of all samples used in the learning phase.

We observe a clear topological organization of the model map. These projections provide a topographical visualization of sequential data set. This figures show that model is useful for visualizing low-dimensional views of high-dimensional sequence.

B. Empirical evaluation of classification

We used the K-fold cross validation technique, with k=3, to estimate the performance of our model. In this case we used data set with {a,b,c,p,q} characters. For each run, the data set was split into three disjoint groups. We used two subsets for training and then tested the model on the remaining subset. The labels generated were compared to the real labels of the test set for each run. We note that here, even the global problem is supervised, for each validation set, the problem is stil unsupervised and the proposed model is used to approximate the density of each class of these sequences.

We then test our model as a classifier and compare it to the HMM. In this case we learn for each run five models, where each GTM model is learned from each set of a given character. We observe that our approach can learn in one map multiple characters and multidimensional sequence using a single map. In table I, we observe that using our model provides more information as topographic visualizations.

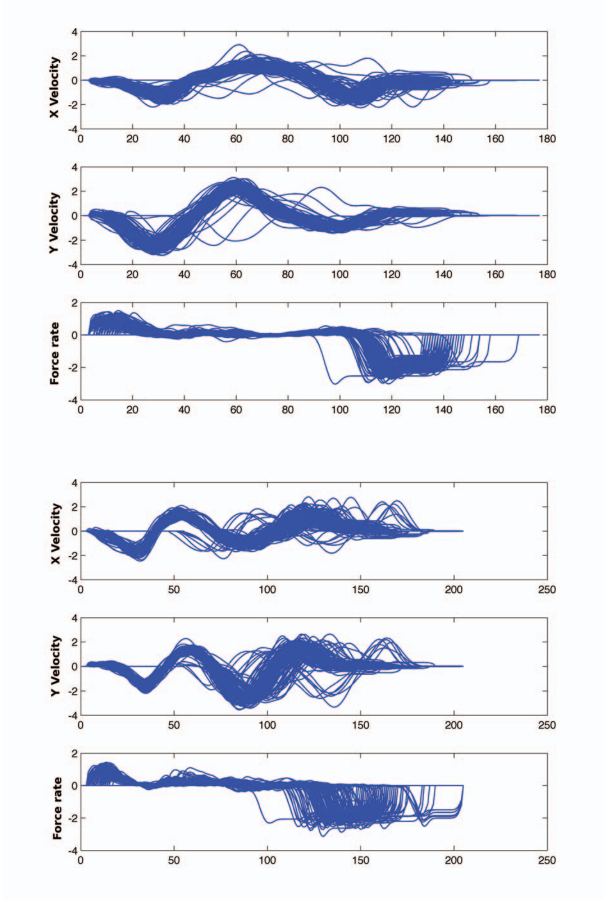

Fig. 1. overlay plot of 131 character samples of the letter "p", and 124 of the letter "q" demonstrating the variation in the data. (a) letter "p", (b) letter "q"

$model$	a	b	c	p	q
$GTM-TT$	100	99.86	99.28	98.41	100
HMM	100	99.28	99.28	97.46	100

TABLE I
CROSS-VALIDATION WITH {a, b,c, p, q} DATA SET. THE VALUE INDICATES A GOOD CLASSIFICATION RATE.

The second experiment is done by learning our model using all characters a, b, c, p, q for each run. Thus we assign the sequence data tests using the Viterbi algorithm and compute the quantization error. Table II shows the rate of quantization error after Viterbi assignment. We observe that our model improves the performance and provides better result than HMM.

V. CONCLUSION AND FUTURE WORK

In this paper we considered the problem of unsupervised topographic modeling of sequences. We proposed an extension of the GTM through model to a mixture framework by considering an HMM with Gaussian mixture states. The model learning is performed by the EM algorithm. First experimental results on a real-world character recognition problem highlight the interest of the proposed approach. Comparison with HMMs, the usual tool for modeling sequences, show that the proposed model is able to provide copetetive results.

Future work will consist in performing further experiments on additional real data and comparisons with standard well-known clustering and visualisation techniques such as the SOM or the standard GTM for non-sequential data.

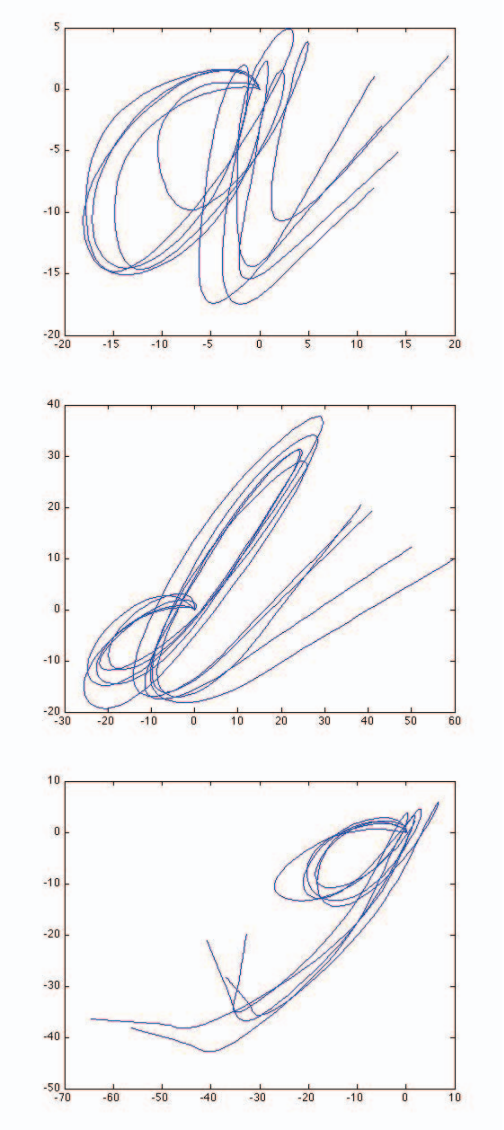

Fig. 2. Reconstruction of $\{a,d,g\}$-data set. Blue characters show the the reconstruction samples

$model$	a	b	c	p	q
$GTM-TT$	2,49	2,46	3,28	2,58	2,99
HMM	2,91	3,15	3,80	2,73	3,22

TABLE II

CROSS-VALIDATION RESULTS FOR THE DATASET OF THE CHARACTERS $\{a, b, c, p, q\}$. THE VALUES INDICATE QUANTIZATION ERROR.

REFERENCES

[1] A. Asuncion and D. Newman, "UCI machine learning repository," http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.

[2] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *Annals of Mathematical Statistics*, vol. 41, pp. 164–171, 1970.

[3] C. M. Bishop and C. K. I. Williams, "Gtm: The generative topographic mapping," *Neural Computation*, vol. 10, pp. 215–234, 1998.

[4] C. M. Bishop, G. E. Hinton, and I. G. D. Strachan, "Gtm through time," in *IEE Fifth International Conference on Artificial Neural Networks*, 1997, pp. 111–116.

[5] C. M. Bishop, M. Svensén, and C. K. I. Williams, "Gtm: The generative topographic mapping," *Neural Computation*, vol. 10, pp. 215–234, 1998.

[6] C. M. Bishop and C. K. I. Williams, "Gtm: A principled alternative to the self-organizing map," in *Advances in Neural Information Processing Systems*. Springer-Verlag, 1997, pp. 354–360.

[7] T. F. Cox, M. A. A. Cox, and B. Raton, "Multidimensional scaling," *Technometrics*, vol. 45, no. 2, p. 182, May 2003.

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

[9] S. Kaski, "Data exploration using self-organizing maps," Ph.D. dissertation, Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering, 1997.

[10] T. Kohonen, *Self-Organizing Maps*, third edition ed., ser. Information Sciences. Springer, 2001.

[11] S. Kong and B. Kosko, "Differential competitive learning for centroid estimation and phoneme recognition," *IEEE Transactions on Neural Networks*, vol. 2, no. 1, pp. 118–124, 1991.

[12] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*. New York: Wiley, 1997.

[13] I. Olier and A. Vellido, "Advances in clustering and visualization of time series using gtm through time," *Neural networks*, vol. 21, no. 7, pp. 904–913, 09 2008.

[14] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[15] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[16] A. Ultsch and H. P. Siemon, "Kohonen's self organizing feature maps for exploratory data analysis," in *Proceedings of International Neural Networks Conference (INNC)*. Kluwer Academic Press, 1990, pp. 305–308.

[17] B. H. Williams, "Extracting motion primitives from natural handwriting data," Ph.D. dissertation, Institute for Adaptive and Neural Computation School of Informatics, 2008.