Probabilistic Self-Organizing Maps for Multivariate Sequences

Rakia Jaziri Mustapha Lebbah Nicoleta Rogovschi Younès Bennani

Abstract—This paper describes a new algorithm to learn a new probabilistic Self-Organizing Map for not independent and not identically distributed data set. This new paradigm probabilistic self-organizing map uses HMM (Hidden Markov Models) formalism and introduces relationships between the states of the map. The map structure is integrated in the parameter estimation of Markov model using a neighborhood function to learn a topographic clustering. We have applied this novel model to cluster and to reconstruct the data captured using a WACOM tablet.

I. INTRODUCTION

Data visualization is an important step in the exploratory phase of data analysis, which becomes more difficult when it involves sequential data. A popular way to treat sequential data is to simply ignore the sequential aspects and treat the observations as independent and identically distributed (i.i.d) in the first stage. For many applications, the i.i.d assumption will be a poor one. Often in many applications the treatment is decomposed in two steps: the first one is the clustering task, and in the second stage, the result of clustering is used to learn a probabilistic model by relaxing the i.i.d. assumption, and one of the simplest ways to do this is to consider a Markov model. There are many possible probabilistic structures that can be constructed according to the needs of particular applications. Graphical models provide a general formalism for motivating, describing and analyzing such structures. Self-organizing map (SOM) is a very intuitive and powerful tool for mining high dimensional data sets [1]. The computational results essentially depend on two priorly chosen ingredients: the metric or the probability distribution, and the grid topology of the self-organizing map. Unfortunately, the paradigms of self-organization cannot be easily transferred to non i.i.d data. Different approaches have been developed to embed temporal information in the Self-organizing map. The most direct way is to include time-delayed versions as input or add preprocessing to capture spatial dynamic [2], [3]. A variety of models exists for recurrent self-organizing maps: the temporal Kohonen map (TKM), the recurrent SOM (RSOM), recursive SOM (RecSOM), and SOM for structured data (SOMSD), [4], [5], [6], [7], [8]. These methods differ in their way of internally representing the temporal context. Another model, often presented as the probabilistic version of the self-organizing map is the Generative Topographic Map (GTM) that has been extended to model time series [9] and structured data [10]. However, the manner in which GTM achieves the topographic organization is quite different from those used in the SOM algorithms and our model. In this paper we propose the extension of the probabilistic self-organizing map algorithm for multivariate sequential data, which we call probabilistic self-organizing sequential data (PrSOMS), by assuming that the sequential data is generated in the same way as HMM (Hidden Markov Models).

HMM is a state transition probabilistic model which is able to model a pattern of time series signals. HMM is very popular in speech recognition field, and widely applied in other fields [11], [12], [13]. However, the organization process is not integrated in their approach. In order to overcome the limitations of HMMs, in [14] the author proposes a novel and an original machine learning paradigm, which is titled topological HMM, that embeds the nodes of an HMM state transition graph in an Euclidian space. This approach models the local structure of HMM and extracts their shape by defining a unit of information as a shape formed by a group of symbols of a sequence.

Other attempts have been made for combining HMMs and SOM (Self-Organizing Map of Kohonen) to built hybrid models that combine the clustering power of SOM with the sequential time series aspect of HMMs [15]. In these hybrid architectures, SOM models are used as front-end processors for vector quantization, and HMMs are then used in higher processing stages [16], [17]. In [18], [19] the authors propose an original combined model which is the offspring of a crossover between the SOM algorithm and the HMM theory. The model’s core consists in a novel hybrid SOM-HMM algorithm where each cell of SOM map presents an HMM. The model is coupled with a sequence data training method, that blends together the unsupervised learning (SOM) and the HMM dynamic programming algorithms. Therefore, it will be very important to have probabilistic SOM algorithms able to infer from a data set of sequences not only the probability distributions but also the topological structure of the model. Unfortunately, this task is very difficult and only partial solutions are today available.

Recently, in [20] the author proposes the extension of the SOMM algorithm (Self-Organizing Mixture Models, [21]) for multivariate time series. However, the manner in which PrSOMS achieves the topographic organization is different from those used in the SOMM models. Another recent and original work, is the Markov random field which defines discrete lattice among the states. Usually the lattice is a regular 2-dimensional grid as in the case of the SOM [22]. Markov random fields appear naturally in problems such as image segmentation.

The aim of this paper is to built a new probabilistic
model for automating and self-organizing the construction of a statistical generative model of not i.i.d data set. The proposed model is based on the probabilistic formalism of the SOM used for i.i.d data set [23, 24, 25]. Therefore, it consists of estimating the parameters of the model by maximizing the likelihood of the sequential data set. The learning algorithm that we propose is an application of the Expectation-Maximization (EM) standard algorithm. In our model, we consider a self-organizing map as a grid forming a discrete topology which each cell represents a state. Note that the word state or cell means the same thing. The generation of the observed variable at a given time step is conditional on the neighborhood cells at that same time step. Thus, a high proximity implies a high probability to contribute to the generation. This proximity is quantified using a neighborhood function. The formalism that we present is valid for all structure of the graph model.

The rest of this paper is organized as follows: Section II covers the mathematical description of the probabilistic self-organizing model dedicated to not i.i.d data (PrSOMS). The application and the experiments are presented in section III. Finally, section IV concludes the article and presents ongoing and future work in addition to possible extensions and applications of the architecture.

II. SELF-ORGANIZING AND GENERATIVE MODEL

A. Problem Formulation

For a better understanding we have used a similar notations to those in the book [26, chap. 13]. Assume an observed vector sequence \( X = \{x_1, x_2, ..., x_n, ..., x_N\} \), where \( x_n \) is an element of sequence, and \( N \) is the length of a sequence. The learning problem is to estimate the parameters of PrSOM model. \( \theta = \{S, \pi, A, \phi\} \) denotes the set of parameters governing the model:

- We assume that the PrSOMS architecture is the lattice \( S \) called a map, which has a discrete topology (discrete output latent space) defined by an undirected graph. We denote the number of cells (states) in \( S \) as \( K \). For each pair of states \((s,k)\) on the map, the distance \( \delta(s,k) \) is defined as the length of the shortest chain linking cells \( k \) and \( s \).
- \( \pi = \{\pi_1, ..., \pi_K\} \) is a set of initial probability distribution;
- \( A \) is the state transition probability distribution;
- \( \phi \) is a set of parameters governing the distribution, which is known as emission probabilities.

Let us introduce a \( K \)-dimensional binary random variable as latent variable \( z^*_n \) and \( z_n \) having a 1-of-\( K \) representation in which a particular element \( z^*_n \) and \( z_n \) is equal to 1 and all other elements are equal to 0. Each component \( z^*_n \) and \( z_n \) indicate a couple of neighborhood cells responsible of generation of an element of sequence. Each element of sequence \( x_n \) is generated by a couple of neighborhood states (cells) \( z^*_n \) and \( z_n \). The \( z^*_n \) indicates the first cell \( s \) selected and \( z_n \) (without star) indicates neighborhood cell \( k \). For each element \( x_n \), we start picking a state \( z_n \) from a map \( S \) according to the prior probability \( p(z^*_n) \). Next, we select an associated neighborhood state \( z_n \) following the conditional probability \( p(z_n/z^*_n) \). All states \( z_n \in S \) contribute to the generation of an element \( x_n \) with \( p(x_n/z^*_n) \) according to the proximity to \( z_n \) described by the probability \( p(z_n/z^*_n) \). Thus, a high proximity to state \( z^*_n \) implies a high probability, and therefore the contribution of state \( z_n \) to the generation of \( x_n \) is high (\( p(z_n/z^*_n)=1 \)). Therefore, the model parameters are completed by defining neighborhood probability \( p(z_n=1/z^*_n=1) = p(z_n/z^*_n) \equiv p(x_n/z^*_n) \) between a couple of states or cells responsible of generation of an element of the observation. To introduce the self-organizing process in the learning PrSOMS model, we assume that \( p(z_n/z^*_n) \) is known as follows:

\[
p(z_n/z^*_n) = p(z_n=1/z^*_n=1) = \frac{\mathcal{K}^T(\delta(k, s))}{\sum_{r \in \mathcal{C}} \mathcal{K}^T(\delta(r, s))},
\]

where \( \mathcal{K}^T \) is a neighborhood function depending on the parameter \( T \) (called temperature); \( \mathcal{K}^T(\delta) = \mathcal{K}(\delta/T), \) where \( \mathcal{K} \) is a particular kernel function which is positive and symmetric \( \lim_{|y| \to \infty} \mathcal{K}(y) = 0 \). Thus \( \mathcal{K} \) defines for each state \( z_n \) a neighborhood region in the map \( S \). The parameter \( T \) allows to control the size of the neighborhood influencing a given state (cell) on the map \( S \). \( T \) is decreased between two values \( T_{max} \) and \( T_{min} \). Using these notations, we can define the parameters of PrSOMS model as follows:

- The conditional distributions of the observed variables \( p(x_n/z^*_n=1) = p(x_n/z_n; \phi) \), where \( \phi \) is a set of parameters governing the distribution which is known as emission probabilities in HMM model. We can represent the emission probabilities in the form

\[
p(x_n/z_n; \phi) = \prod_{k=1}^{K} p(x_n/z_k; \phi_k)^{z_{nk}}
\]

In the case of spherical Gaussian emission densities we have \( p(x_n/z_k; \phi_k) = N(x_n; w_k, \sigma_k^2 I) \), defined by its mean \( w_k = (w_{k1}^1, ..., w_{k1}^{w_R}) \) and its covariance matrix, defined by \( \sigma_k^2 I \) where \( \sigma_k \) is the standard deviation and \( I \) is the identity matrix, \( N(x_n; w_k, \sigma_k) = \frac{1}{(2\pi\sigma_k^2)^{w_R}} \exp \left[ -\frac{1}{2 \sigma_k^2} ||x_n-w_k||^2 \right] \).

- The initial latent state \( z^*_1 \), which has a marginal distribution \( p(z^*_1) \) (prior probability) represented by a vector of probabilities \( \pi \) with elements \( \pi_k = p(z^*_1=k) = 1 \), so that \( p(z^*_1=1) = \prod_{k=1}^{K} \pi_{z^*_1} \), where \( \sum_k \pi_k = 1 \).

- Transition probability \( A \) : we allow the probability distribution of \( z^*_n \) to depend on the state of the previous latent variable \( z^*_n-1 \) through a conditional distribution \( p(z^*_n|z^*_n-1) \). Because the latent variables are \( K \)-dimensional binary variables, these conditional distributions correspond to a table of probabilities that we denote by \( A \). The elements of \( A \) are known as transition probabilities denoted by \( A_{jk} = p(z^*_n=k|z^*_n-1=j) = 1, \) with \( \sum_k A_{jk} = 1 \). In our case the number of transitions are limited by the grid (map). We can then
write the conditional distribution explicitly in the form:

\[ p(z_n^*/z_{n-1}, A) = \sum_{k=1}^{K} \sum_{j=1}^{K} A_{jk} z_{n-1,j} z_{nk} \]

All of the conditional distributions governing the latent variables share the same parameters \( A \).

### B. Cost function and optimization

We denote the set of all latent variables by \( Z^* \) and \( Z \), with a corresponding row \( z_n^* \) and \( z_n \) associated to each sequence element \( x_n \). Now assume that, for each sequence of observations in \( X \) corresponds the latent variables \( Z \) and \( Z^* \). We denote by \( \{ X, Z, Z^* \} \) the complete data set, and we refer to the observed data \( X \) as incomplete. The likelihood function is obtained from the joint distribution by marginalizing over the latent variables \( Z^* \) and \( Z \)

\[ p(X; \theta) = \sum_{Z^*} \sum_{Z} p(X, Z, Z^*; \theta) \]  

An important concept for probability distributions over multiple variables is the conditional independence [27]. Thus, we assume that the conditional distribution of \( X \), given \( Z^* \) and \( Z \), is such that it does not depend on the value of \( Z^* \). Often used for graphical model, so that \( p(X/Z, Z^*) = p(X/Z) \). Thus the joint distribution of the sequence is equal to:

\[ p(X, Z^*, Z) = p(Z^*) p(Z|Z^*) p(X/Z) \]

and we can rewrite the marginal distribution as

\[ p(X; \theta) = \sum_{Z^*} p(Z^*) \sum_{Z} p(Z|Z^*) p(X/Z) \]

We note that \( p(X/Z^*) \) is

\[ p(X/Z^*) = \sum_{Z} p(Z|Z^*) p(X/Z) \]

The generation is not dependent only on the current state but also on the previous state (i.e. on the transitions) and on the neighborhood cells at that same time step \( p(Z^*/Z^*_\text{old}) \). Thus, a high proximity implies a high probability to contribute to generation.

The joint probability distribution over a sequence of observed variables and both latent \( Z \) and \( Z^* \) is then given by

\[ p(X, Z^*, Z; \theta) = \left[ p(z_1^* | \pi) \prod_{n=2}^{N} p(z_n^*/z_{n-1}; A) \right] \times \prod_{i=1}^{N} p(z_i/z_i^*) \times \prod_{m=1}^{N} p(x_m/z_m; \phi) \]  

\[ \theta = \{ \pi, A, \phi \} \] denotes the set of parameters governing the model. It’s not obvious to maximize the likelihood function, because we obtain complex expressions with no closed-form solutions. Hence, we use the expectation maximization algorithm to find parameters for maximizing the likelihood function. EM algorithm starts with some initial selection for the model parameters, which we denote by \( \theta^{old} \). In the E step, we take these parameter values and find the posterior distribution of the latent variables \( p(Z^*, Z/X, \theta^{old}) \). We then use this posterior distribution to evaluate the expectation of the logarithm of the complete-sequence data likelihood function (4), as a function of the parameters \( \theta \), to give the function \( Q(\theta, \theta^{old}) \) defined by:

\[ Q(\theta, \theta^{old}) = \sum_{Z^*} \sum_{Z} p(Z^*, Z/X, \theta^{old}) \ln p(X, Z^*, Z; \theta) \]

The maximization of the function \( Q(\phi, \theta^{old}) \) then gives

\[ \mathbf{w}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})} \]  

\[ \sigma_k^2 = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) |x_n - \mathbf{w}_k|^2}{d \sum_{n=1}^{N} \gamma(z_{nk})} \]

where \( d \) is the dimension of the element \( x \).

\[ \pi_k = \frac{\gamma(z_{nk})}{\sum_{j=1}^{K} \gamma(z_{nk})} \]

\[ A_{jk} = \frac{\sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^{K} \sum_{n=2}^{N} \xi(z_{n-1,j}, z_{nl})} \]

where

\[ \xi(z^*_{n-1,j}, z_{nk}) = \mathbb{E}[z^*_{n-1,j} z_{nk}] = \sum_{z^*} \gamma(z^*) z^*_{n-1,j} z_{nk} \]

We seek an efficient procedure for evaluating the quantities \( \gamma(z^*_{nk}), \gamma(z_{nk}) \) and \( \xi(z^*_{n-1,j}, z_{nk}) \), corresponding to the E step of the EM algorithm. As the hidden Markov model, this is known as the forward-backward algorithm [28], or the Baum-Welch algorithm [29], [30]. We use the notations \( \alpha(z_{nk}) \) and \( \alpha(z_{nk}) \) to denote the values of \( \alpha(z^*) \) and \( \alpha(z) \) when \( z^*_{nk} = 1, z_{nk} = 1 \) with an analogous notation of \( \beta \). Using Bayes theorem, we have

\[ \gamma(z^*_{nk}) = \frac{\alpha(z^*_{nk}) \beta(z_{nk})}{p(X)} \]

Using the similar decomposition we obtain

\[ \gamma(z_{nk}) = \frac{\alpha(z_{nk}) \beta(z_{nk})}{p(X)} \]

The values \( \alpha(z^*_{nk}) \) and \( \alpha(z_{nk}) \) represent respectively the joint probability of observing all of the given data up to time \( n \), the
values of \( z_n^* \) and \( z_n \): \( \beta(z_n^*) \) and \( \beta(z_n) \) represent respectively the conditional probability of all future data from time \( n + 1 \) up to \( N \) given the values of \( z_n^* \) and \( z_n \). The values of \( \alpha(z_n^*) \) and \( \alpha(z_n) \) are calculated by forward recursion as follows:

\[
\alpha(z_n^*) = \left[ \sum_x p(x_n/z_n^*) p(z_n^*/z_n^*) \right] \times \sum_{z_n^*} \alpha(z_n^*) p(z_n^*/z_n^*)
\]

(9)

and

\[
\alpha(z_n) = p(x_n/z_n) \sum_{z_n^*} p(z_n^*/z_n)
\]

(10)

The value of \( \beta(z_n^*) \), are calculated by backward recursion as follows:

\[
\beta(z_n^*) = \sum_{z_n^+} \beta(z_n^+1) \left[ \sum_x p(x_n+1/z_n^+1) p(z_n^+1/z_n^+1) \right] \times p(z_n^+1/z_n^*)
\]

\[
\beta(z_n) = \frac{1}{p(z_n)} \sum_{z_n^*} p(z_n^*) p(z_n^*/z_n) \sum_{z_n^+} \sum_{z_n^+1} p(z_n+1/z_n^+1) \beta(z_n^+1) p(z_n^+1/z_n^+1) p(z_n^*/z_n^*)
\]

(11)

where

\[
p(x_n+1/z_n^+1) = \left[ \sum_x p(x_n+1/z_n^+1) p(z_n+1/z_n^+1) \right]
\]

\[
p(z_n) = \sum_{z_n^*} p(z_n^*) p(z_n^*/z_n^*)
\]

It is clear that the neighborhood function is independent of time:

\[
p(z_n+1/z_n^*) = p(z_n^*)
\]

\[
p(z_n+1/z_n^*) = p(z_n+1, k = 1/z_n^+1, s = 1) = \frac{\phi}{\sum_{r,s} \phi(r,s)}
\]

Next we consider the evaluation of the quantities \( \xi(z_n^* \rightarrow 1, z_n^*) \) which correspond to the values of the conditional probabilities \( p(z_n^* \rightarrow 1, z_n^*/X) \) for each of the \( K \times K \) settings for \( (z_n^*, z_n) \). Using the applying Bayes theorem, we obtain:

\[
\xi(z_n^* \rightarrow 1, z_n^*) = \frac{\alpha(z_n^*) \left[ \sum_x p(x_n/z_n^*) p(z_n^*/z_n^*) \right]}{p(X)} \times \frac{\beta(z_n^*)}{p(X)}
\]

If we sum both sides of \( \alpha(z^*) \) over \( z_N \), we obtain

\[
p(X) = \sum_{z_N^*} \alpha(z_N^*)
\]

Then we compute the forward \( \alpha \) recursion and the backward \( \beta \) recursion and use the results to evaluate \( \gamma \) and \( \xi(z_n^* \rightarrow 1, z_n^*) \). We use these results to compute a new parameter model \( \theta(z, \lambda, \phi) \).

Let us summarize the steps required to train a PrSOM model using the EM algorithm. We first make an initial selection of the parameters \( \theta^{old} \) where \( \theta(z, \lambda, \phi) \). As the traditional HMM the parameters \( A \) and \( \pi \) are initialized uniformly from a uniform distribution. The parameters \( \phi \) can be initialized by applying the clustering algorithm on data with i.i.d hypothesis.

### III. Experimentations

The dataset here were used for a PhD study on primitive extraction using HMM based models. The data consists of 2858 character samples, contained in the cell array 'mixout' [31], [32]. The data was captured using a WACOM tablet, where 3 dimensions were kept - \( x, y \), and pen tip force. The data has been numerically differentiated and Gaussian smoothed, with a sigma value of 2. Data was captured at 200Hz and was normalized. 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 experiments were performed in Matlab using Bayes Net Toolbox written by Kevin Murphy, 1997–2002 and SOM Toolbox. The single best state sequence is obtained by using the Viterbi algorithm [28]. Selecting only the best sequence corresponds to the winner-take-all principle of the PrSOMS. 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. In order to show the advantages of self-organization of PrSOMS, we run separately PrSOMS model on 4 data sets: \( p \)-data set, \( q \)-data set, \( pq \)-data set, \( abc \)-data set.

Figures 1 and 2 show respectively the PCA projection of the \( p \)-data set and \( q \)-data set visualized in the latent-variable (maps projection). The blue points present the element of original sequence. We observe a clear topological organization of the PrSOMS map. These projections provide a topographical visualization of sequential data set. We can clearly observe that the topological order of state respects the topological order of data set: close samples are represented by close states. This figures show that PrSOMS is useful for visualizing low-dimensional views of high-dimensional sequence.

In order to show that learning phase is well done, we calculate Viterbi path for each sample. Figures 3 and 4 show for one example, that Viterbi algorithm indicates an organized path indicated in red line (figures: 3.a, 4.a): neighborhood components of sequences are represented by neighborhood cells. The followed figures 3.b and 4.b indicate the original (blue line, left figure) and reconstruction (red line, left figure) of all samples used in the learning phase.

To show the generative model power, we reconstructed...
the character \( p \) using the \( q \)-PrSOMS map learned with the character \( q \) and vice versa. Figures 5.a and 5.b show in both case, that each model provides a good reconstruction of the common part of the character \( p \) and \( q \).

On the other hand, in order to show again that PrSOMS model has ability to distinguish common features, we trained PrSOMS model with \( pq \)-data set. The PrSOMS map plotted in figure 6 shows a clear organization. Figure 6 shows training samples \( pq \)-data set and \( pq \)-PrSOMS followed by original and reconstructions \( p \) and \( q \) samples. Note that the map of states share states across characters. One of the characteristic of feature distinguishing PrSOMS from other HMM paradigms, is the topographic preservation using the neighborhood function. Figure 7 shows the reference maps (profiles) corresponding respectively to \( p \)-PrSOMS map, \( q \)-PrSOMS map and \( pq \)-PrSOMS map. Each cell plot the 3 variables : \( x \) velocity, \( y \) velocity and differential pressure.

The topological maps in our PrSOMS model allowed us to visualize the partition. So, the experts could use the plot to analyze the organization of the samples and detects the relevant features. PrSOMS reveals an important implication to the neighborhood function to organize PrSOMS states. A clear organization of prototype is observed in each map. Various model selection techniques could be employed to automate the selection of the appropriate features and states. We can detect the relevance features associated to each state or character. Hence, a common area and features is observed looking at the same time \( p \)-PrSOMS map, \( q \)-PrSOMS map and \( pq \)-PrSOMS map.

A. Discussion about self-organization process : From PrSOMS to traditional HMM

The PrSOMS model allows us to estimate the parameters maximizing the log-likelihood function \( Q^T \) for a fixed \( T \), which is decreased between \( T_{max} \) and \( T_{min} \). As a more challenging problem we used \( abc \)-data set. The PrSOMS model is trained on multi-character data sets. Figure 8 shows
3 different characters drawn using the same \( abc \)-PrSOMS map. To make a quick comparison with the conventional SOM we run a learning with the same size and considering i.i.d data. For this map we use the classical assignment function to reconstruct characters. We see clearly in figure 9 that the reconstructed character are of low quality in the case of SOM. Although, given the specificity of the data set, SOM managed to rebuild the character \( c \).

In order to study the self-organization process we used in this case random initialization. We used 12\( \times \)12 map in a two-dimensional latent space. In figure 11 we show the means of the Gaussians, connected according to the PrSOMS map in the latent space. It can be clearly seen how the PrSOMS spreads out over the data as the neighborhood function is shrunk. Figure 10 shows the same projection of the means of the Gaussians, connected according a regular intervals in a 1D latent space. Here, we show the configuration after each 5 iterations and the latest iteration. We can observe that the PrSOMS spreads out over the data as the neighborhood function width decreases. When decreasing \( T \), the model of PrSOMS will be defined in the following way:

- The first step corresponds to high \( T \) values. In this case, the influencing neighborhood of each state \( z_{ns}^* \) on the PrSOMS map is important and corresponds to higher values of \( K_T(\delta(k,s)) \). Each state \( s \in S \) uses a high number of neighborhood states to generate an element of sequence. This step provides the topological order.
- The second step corresponds to small \( T \) values. The neighborhood is small and limited. Therefore, the adaptation is very local and the parameters are accurately computed from the local density. In this case we consider that PrSOMS converge to traditional HMM.

It is clear that the computational cost of our PrSOMS model is more expensive than traditional HMM. However, by restricting the neighborhood of \( z^* \) to limit a number of pairs of states \( (z, z^*) \), we can obtain a low computational cost.

B. Empirical evaluation of classification

We used the \( K \)-fold cross validation technique, with \( k=5 \), to estimate the performance of PrSOMS. In this case we used data set with \( \{ a, b, c, d, e, g, h, l, m, n, o, p, q \} \). For each run, the data set was split into five disjoint groups. We used four 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. Empirical results were obtained using PrSOMS, and HMM. Our purpose is not to show that our approach is better or not, but to show that our approach can be used as classifier. In table I, we observe that the use of topological maps improved the performance of PrSOMS algorithm and reduced the deviation of the results. Each value indicates the rate of good classification (accuracy).
Fig. 8. Sample reconstructions using abc-PrSOMS map $12 \times 12$. Blue characters show the original samples (left subfigure).

Fig. 9. Sample reconstructions using abc-SOM map $12 \times 12$. Blue characters show the original samples (left subfigure).

Fig. 10. Configuration of the abc-PrSOMS $1 \times 20$ training after 5; 10; ...; 20 iterations.

Fig. 11. Configuration of the abc-PrSOMS $12 \times 12$ training after 5; 15; 20 iterations.
followed by the associated deviation. We remind here that PrSOMS model provides more informations as topographic visualizations.

<table>
<thead>
<tr>
<th>model</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrSOMS std</td>
<td>100</td>
<td>99.16</td>
<td>99.28</td>
<td>100</td>
<td>99.47</td>
<td>100</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>HMM std</td>
<td>100</td>
<td>99.28</td>
<td>99.28</td>
<td>99.35</td>
<td>98.92</td>
<td>100</td>
<td>85.66</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>model</th>
<th>l</th>
<th>m</th>
<th>n</th>
<th>o</th>
<th>p</th>
<th>q</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrSOMS std</td>
<td>93.13</td>
<td>68</td>
<td>80</td>
<td>100</td>
<td>97.41</td>
<td>100</td>
</tr>
<tr>
<td>HMM std</td>
<td>3.92</td>
<td>12.82</td>
<td>3.7</td>
<td>0</td>
<td>2.34</td>
<td>0</td>
</tr>
</tbody>
</table>

TABLE I

CROSS-VALIDATION WITH \{a, b, c, d, e, g, h, l, m, n, o, p, q\}-DATA SET. GOOD CLASSIFICATION RATE.

IV. CONCLUSION

We have presented a novel mathematical paradigm that extends the traditional probabilistic self-organizing map in order to capture and model topographic information present in the not i.i.d data. PrSOMS generalizes HMM formalism and therefore provides self-organization map within a single probabilistic learning scheme. The proposed PrSOMS model provides also a novel topographical visualization tool for sequential data set. PrSOMS model is very well adapted for multidimensional sequences (not i.i.d observations), and it maintains a low computational cost. As future work, PrSOMS can be applied for categorical data and binary data. Also providing an application of PrSOMS dealing with mixed data should be an interesting task.

REFERENCES