Mode detection via neuron proximity in monitored self-organizing maps

Susana Vegas-Azcárate
Statistics and Decision Sciences Group
Rey Juan Carlos University
28933 Móstoles, Spain
susana.vegas@urjc.es

Jorge Muruzábal
Statistics and Decision Sciences Group
Rey Juan Carlos University
28933 Móstoles, Spain
jorge.muruzabal@urjc.es

Abstract - *We are living the blast of the silicon-based biology era, an era in which investigation of complete genomes is viable for the first time. We need computer-based technologies to cope with the vast quantities of data generated by genome projects, seeking not only increased facilities in data storage and access, but also assistance in computational manipulation and post-processing. Without methods that help us to analyze this deluge of data, the information it contains becomes useless. This paper reconsiders the basic toolkit for cluster analysis —based on the relative distance from each pointer to its immediate neighbours on the network— from this monitoring perspective. It is shown that the idea works nicely, that is, much useful information can be encoded and recovered via the trained map alone (ignoring any possible density estimate available). Moreover, the fact that a topographic map is not restricted to metric vector spaces makes this learning structure a perfect tool to deal with biological data, such as DNA or protein sequences of living organisms, for which only a similarity measure is readily available.*

Key words - Topographic maps, monitoring studies, neuron interdistances.

Note - This paper constitutes an extension of the one presented at the Workshop on Biosignal Processing and Classification (BPC), Barcelona, Spain, 2005, *On cluster analysis via neuron proximity in monitored self-organizing maps,* Susana Vegas-Azcárate and Jorge Muruzábal.

1 Introduction

The Self-Organizing Map (SOM) [Kohonen (2001)] is a popular neural network model for unsupervised learning that tries to 'imitate' the *self-organization* process taking place in the sensory cortex of the human brain (by which neighbouring neurons will typically be activated by similar stimuli). This model develops a mapping from a d-dimensional input space, into an equal or lower-dimensional discrete lattice with regular, fixed topology. Thanks to a simple competitive learning process — whereby only weights connected to the winner (or best matching unit) and its neighbours are updated—, the SOM structure is often organized (topologically ordered).

We are interested in applying the SOM neural structure for sound clustering of biological signals. We have a particular application in mind, namely, clustering proteins. This problem has some tradition in the literature [Ferrán and Ferrara (1991); Kohonen and Somervuo (2002)], and it prompts a number of interesting issues, beginning with the issue of the 'vectorial vs. nonvectorial' data representation. However, in this paper we consider only the vectorial case and pursue some foundational research (i) focusing on the use of interneuron proximity information *alone*, and (ii) emphasizing the need to monitor the training process. While both theory and applications have developed substantially in the SOM literature, there is probably not a wide awareness yet (among practitioners) of the technical requirements for the extracted maps, nor there seems to be a wide consensus on how to best monitor, train or even analyze the SOM structure.

For clustering purposes, a good density estimate of the sampling distribution can of course be very valuable, and certain kernel-based learning algorithms are naturally suited to yield such estimates. While conventional kernel-based estimates [Gray and Moore (2003); Davies and Kovac (2004)] seem to provide generally good results, drawbacks in higher dimensions should still occur in the multivariate case [Scott and Szewczyk (2000)]. Recent approaches to SOM training usually incorporate some statistical machinery yielding richer models and more principled fitting algorithms (the standard framework lacks a statistical model for the data and thus provides no density estimate).

As regards monitoring, we have introduced an early-stopping criterion called UDL (for uniform data load) and we have shown that it provides sensible density estimates in a wide array of cases [Muruzábal and Vegas-Azcárate (2005)]. Here we show that the UDL criterion is also useful for the usual basic proximity summaries (available in all training algorithms). To this end, four algorithms are tested. Specifically, the batch version (SOM-B) [Kohonen (2001)] and a convex adjustment (SOM-Cx) [Zheng and Greenleaf (1996)] of the standard SOM algorithm are compared to two kernel-based learning rules : the generative topographic mapping (GTM) [Bishop et al. (1997b)] and the kernel-based maximum entropy learning rule (kMER) [Van Hulle (1998)]. The latter three tend to achieve the 'equiprobabilistic state' that motivates the UDL criterion [Muruzábal and Vegas-Azcárate (2005)] ; it appears unlikely that SOM-B can achieve this state, but it is still monitored in the same way for the sake of reference.

The organization is as follows. Section 2 briefly describes the four topographic map formation algorithms considered in the paper. Section 3 spells out the particular training and testing strategies examined in the experiments reported in section 4. Some conclusions are drawn in section 5.

2 SOM training algorithms

Before we actually describe the details of the algorithms, it is appropriate to begin with a bit of background for the work presented here. As often noted, the quality of the SOM fit will be assessed in the first place by the extent to which the organization or topology preservation property holds. A substantial amount of research has been devoted to the formalization and quantification of this idea. Proposed approaches vary from the early measures based entirely on the map pointers [Bauer and Pawelzik (1992)] to increasingly sophisticated variants that also incorporate some aspects of the data into the analysis [Kaski and Lagus (1996); Villmann et al. (1997); Lampinen and Kostiainen (1999); Haese and Goodhill (2001)]. While much progress has been done in the area, there is no universally accepted methodology for practitioners to follow. Many proposed measures are not easy to interpret, and the available implementations are scarce. At any rate, empirical confirmation is needed in each case.

Determination of the most sensible criterion is also complicated by the nature of the training algorithms. Most importantly, algorithms differ in their magnification factors, that is, in the ability to reflect exactly the true density generating the data. When the match between pointer density and data density is exact (at least asymptotically) we talk of equiprobabilistic maps [Van Hulle (2000)]. These have been argued to overcome the output unit underutilization problem found in the standard SOM training algorithm. But even among theoretically equiprobabilistic algorithms there are, as we shall see, substantial differences in practical behaviour.

Finally, training algorithms vary also in the amount of modelling machinery involved. As noted above, the traditional SOM structure lacks a statistical model for the data, whereas modern training algorithms like GTM and kMER provide an explicit density estimate that can be very useful for clustering purposes. These estimates enhance the framework and raise questions about the preferred training strategy. Denote the SOM weight or pointer vectors as $w_i \in \mathrm{IR}^d, i = 1, ..., N$, and the data as $v_m \in \mathrm{IR}^d, m = 1, ..., M$. In this paper we consider 2D SOMs only ; besides, we restrict consideration to squared maps equipped with the standard topology.

2.1 The Batch SOM

The original version of the SOM is a *sequential* algorithm, which makes a separate update for each data point, taken one at a time. There is also a batch version of the SOM algorithm (SOM-batch) [Kohonen (2001)], for which each update of model parameters is based on all data points. The batch version of Kohonen' SOM algorithm resembles the Linde-Buzo-Gray algorithm for vector quantization [Linde et al. (1980)], which, if a batch computation is considered, is usually called the *k-means* algorithm. Moreover, SOM-batch corresponds to the Heskes' approach when $\beta \to \infty$ and the neighbourhood function verify $\Lambda_{ij} \in \{0, 1\}$. SOM-batch algorithm is summarized in Algorithm 1.

Since SOM-batch contains no learning rate parameter, no convergence problems arise, so more stable asymptotic values for the weight vectors are obtained [Kohonen (2001)]. As in the sequential SOM, a usual choice is a Gaussian-shaped neighbourhood function.

Algorithm 1
Self-Organizing Map, batch version (SOM-batch).

Initialize the weight vectors, $\mathbf{w}_i^0$.
repeat for each iteration, t,
  for each data point, $\mathbf{v}_n$, **do**
    Find its the best matching unit.
  end for.
  Update weight vectors by

$$\mathbf{w}_i^t = \frac{\sum_{n=1}^{N} \mathbf{v}_n \cdot \Lambda_{i_n^* i}^t}{\sum_{n=1}^{N} \Lambda_{i_n^* i}^t}. \tag{1}$$

  Decrease the width of the neighbourhood function.
until t reaches T.

2.2 Convex adjustment

The convex SOM (SOM-cx) [Zheng and Greenleaf (1996)] is a nonlinear model of weight adjustments based on the original SOM, developed to approach the probability distribution of the inputs. SOM-cx uses a convex model to adjust weights to produce equiprobabilistic maps, preserving the simplicity of the basic SOM to retain the biological plausibility. This model provides more efficient data representation, whereas the convergence rate is comparable to that of the linear model [Zheng and Greenleaf (1996)], see Algorithm 2.

Zheng and Greenleaf (1996) have focused their studies on the mapping between the 1D output density and the input probability $p(\mathbf{v})$, presenting numerical demonstrations to validate their methods, but did not discuss other properties of SOMs. They revised Ritter and Schulten (1986), Hertz et al. (1991) and Ritter (1991) techniques to prove the equiprobabilism of linear maps trained with SOM-cx. Following the derivation of output units density for the 1D SOM case the density of output units appears to be proportional to $p(\mathbf{v})^{\frac{2}{3}}$. What Zheng and Greenleaf (1996) have found is that the 1D output density is proportional to the input probability $p(\mathbf{v})$ if a convex function of $(\mathbf{v} - \mathbf{w}_i)$ is used— convex in the sense of the $\Delta\mathbf{w}_i$ axis. Therefore, the relative size of the weight adjustments of inputs having high probabilities is increased. Moreover, regardless of the probability distribution of inputs, all output units have similar winning frequencies, a desired feature for efficient data representation with a minimum number of under-utilized units.

2.3 Generative Topographic Mapping

The Generative Topographic Mapping (GTM) [Bishop et al. (1997b)] defines a non-linear parametric mapping from a low-dimensional latent space to a high-dimensional data space. Thanks to its topology preserving properties, nearby points in latent space will map to nearby points in data space. Specifically, the mapping is denoted $\mathbf{y}(\mathbf{u}, \mathbf{W})$, where $\mathbf{u} \in \mathbb{R}^l$, $\mathbf{y} \in \mathbb{R}^d$ (where $l < d$) and $\mathbf{W}$ denotes the matrix of parameters. The mapping is governed by $\mathbf{W}$, and could consist, for example, of a feed-forward neural network in which case $\mathbf{W}$ would

Algorithm 2
Self-Organizing Map, convex version (SOM-cx).

Initialize the weight vectors, $\mathbf{w}_i^0$.
repeat for each iteration, t,
Select a data point, $\mathbf{v}_n$.
Find the best matching unit for $\mathbf{v}_n$.
Update weight vectors so that

$$\mathbf{w}_i^t = \mathbf{w}_i^{t-1} + \eta^t \cdot \Lambda_{i_n^* i}^t \cdot (\mathbf{v}_n - \mathbf{w}_i^{t-1})^{\frac{1}{\kappa}}, \tag{2}$$

with κ a positive odd integer.
Decrease the value of η and the width of the neighbourhood function.
until t reaches T.

represent the weights and biases.

The transformation $\mathbf{y}(\mathbf{u}, \mathbf{W})$ maps the latent-variable space into an l-dimensional manifold S embedded within the data space. By suitably constraining the model to a grid in latent space, a posterior distribution over the latent grid is readily obtained using Bayes' theorem for each data point [Bishop et al. (1997a)]. GTM training process is based on a standard EM procedure [Dempster et al. (1977)] aimed at the standard Gaussian-mixture log-likelihood [Bishop et al. (1997b, 1996)]

$$\log \mathcal{L} = \sum_{n=1}^{N} \log \left\{ \sum_{i=1}^{M} p(\mathbf{v}_n|i) P(\mathbf{u}_i) \right\}, \tag{3}$$

where $P(\mathbf{u}_i)$ is the prior mass at each point in the latent grid, and $p(\cdot|i)$ is the Gaussian density centered at $\mathbf{y}_i = \mathbf{y}(\mathbf{u}_i, \mathbf{W})$ (equal, of course, to the weight vector $\mathbf{w}_i$) with spherical covariance and common variance $\sigma^2 = \beta^{-1}$. A generalized linear regression model is typically chosen for the embedding map, namely $\mathbf{y}(\mathbf{u}, \mathbf{W}) = \mathbf{W}\Phi(\mathbf{u})$, where $\Phi \equiv \Phi(\mathbf{u})$ is a $M \times B$ matrix containing the scores by B fixed basis functions and $\mathbf{W}$ is the free matrix to be optimized together with β. The prior distribution in latent space is a sum of equally-weighted Dirac functions allocated at the M nodes of the lattice, that is,

$$P(\mathbf{u}_i) = \frac{1}{M} \sum_{j=1}^{M} \delta_{\mathbf{u}_j}(\mathbf{u}_i). \tag{4}$$

The posterior $p(\mathbf{u}|\mathbf{v}_n)$ concentrates to a single node as training proceeds. A regularization term λ can be added to the objective function to control the topological order in the mapping. This is accomplished via a Gaussian prior

$$p(\mathbf{W}) = \left(\frac{\lambda}{2\pi} \right)^{\frac{Bd}{2}} \exp \left\{ -\frac{\lambda}{2} \parallel \mathbf{W} \parallel^2 \right\}.$$

Hence, GTM leads to an homoscedastic and homogenous mixture density model possibly penalized by excessive complexity in the embedding mapping. This way, the GTM algorithm

Algorithm 3
Generative Topographic Mapping (GTM).

Generate the grid of latent points.
Generate the grid of basis function centers.
Select the basis function width.
Compute the matrix of basis function activations.
Initialize the matrix of parameters, $\mathbf{W}$, randomly or using PCA.
Initialize the common variance, β^{-1}.
repeat
  E-step
    Compute the posterior probability that $\mathbf{v}_n$ was generated by $\mathbf{y}_i$.
    Compute the generalized least squares.
  M-step
    Update the matrix of parameters, $\mathbf{W}$.
    Update the common variance, β^{-1}.
until convergence.

has a connection with Utsugi (1997) proposal, an homoscedastic Gaussian mixture model for density estimation with equal mixing and a single Gaussian smoothing prior to reflect centroid variations in the neuron space.

GTM algorithm is mainly used to visualize and modelize high-dimensional data in a low dimensional latent variable space. In fact, the GTM is proposed as a principled dimensional reduction method. See Algorithm 3.

2.4 Kernel-based Maximum Entropy learning Rule (kMER)

The kernel-based Maximum Entropy Learning Rule (kMER) [Van Hulle (2000)] was introduced as an unsupervised competitive learning rule for non-parametric density estimation. Its main purpose is to obtain equiprobabilistic topographic maps on regular, fixed-topology lattices. Here, neurons receptive fields are radially overlapping symmetric kernels, whose radii are adapted to the local input density together with the weight vectors that define the kernel centroids. Note that, in the case of the standard SOM algorithm, the receptive fields are the standard Voronoi regions defined by the minimum Euclidean distance rule.

A sequential version of kMER is developed in Van Hulle (1998, 2000), and can be seen in Algorithm 4. As in the standard SOM algorithm, the neighbourhood function Λ decreases over time to achieve a topographically organized lattice. This function is a critical factor towards the generation of topology-preserving mappings, in as far as it achieves the cooperation and specialization of neighbouring neurons for similar input signals. Apart from this cooperative state, there is also a competitive element in the learning process of the receptive field centers, $\mathbf{w}_i$, which achieves the input space to be evenly filled with receptive fields according to the input distribution statistical properties [Van Hulle (2000)]. This competitive element is introduced by the fuzzy function Ξ, providing a convenient extension of the winner-take-all scheme. This fuzzy code membership function is quite different from those used in the fuzzy-

Algorithm 4
Kernel-based Maximum Entropy Learning Rule (kMER), sequential version.

Initialize the weight vectors $\mathbf{w}_i^0$ and the radii σ_i^0 of the receptive fields.
repeat for each iteration, t,
Select a data point, $\mathbf{v}_n$.
Find the best matching unit for $\mathbf{v}_n$.
Determine active neurons— the neurons that have a receptive field in which the current input $\mathbf{v}_n$ falls.
Update weight vectors, employing a learning rate η, a neighbourhood function Λ, a sing function Sgn and a fuzzy code membership Ξ,

$$\mathbf{w}_i^t = \mathbf{w}_i^{t-1} + \eta^t \cdot \sum_{j=1}^{M} \Lambda_{i_n^* i}^t \cdot \Xi_j(\mathbf{v}_n) \cdot Sng(\mathbf{v}_n - \mathbf{w}_i^{t-1}). \tag{5}$$

Update radii, using a scale factor ρ and a characteristic function ϑ,

$$\sigma_i^t = \sigma_i^{t-1} + \eta^t \left(\frac{\rho}{M-\rho}(1 - \vartheta_i(\mathbf{v}_n)) - \vartheta_i(\mathbf{v}_n) \right). \tag{6}$$

Decrease the value of η and the width of the neighbourhood function.
until t reaches T.

clustering literature. In fact, Ξ_i is a measure for the probability that neuron i belongs to the subset of activated neurons, rather than just the probability that neuron i is active. This way, Ξ expands the winner-take-all scheme of the original SOM algorithm, allowing receptive fields to overlap. Function Ξ takes the following form,

$$\Xi_i(\mathbf{v}) = \frac{\vartheta_i(\mathbf{v})}{\sum_{j=1}^{M} \vartheta_j(\mathbf{v})}, \tag{7}$$

where ϑ_i

$$\vartheta_i(\mathbf{v}) = \begin{cases} 1 & \text{if} \quad \mathbf{v} \in S_i \\ 0 & \text{if} \quad \mathbf{v} \notin S_i \end{cases} \tag{8}$$

is the characteristic function of the d-dimensional hyperspherical receptive field region S_i, which is defined as the cross-section of the kernel function at centroid $\mathbf{w}_i$ with a threshold that varies along learning. Thus, receptive field centers $\mathbf{w}_i$ will be updated proportionally to Ξ_j in the direction of the current input $\mathbf{v}_n$. Moreover, kMER derives a different standard deviation σ_i for each mixture Gaussian component. Specifically, the kernel radii σ_i are adjusted so as to verify, at convergence, that the probability of each neuron i to be active is given by $P(\vartheta_i \neq 0) = \frac{\rho}{M}$, that is, Equation 6, where ρ is a scale factor that controls the degree of overlap among receptive fields. Then, the radii are guaranteed to converge to a lattice whose neurons have an equal probability to be active, when the probability density is continuous and statistically stationary. It can be seen that the receptive field weight centers $\mathbf{w}_i$ and the radii σ_i are adapted to achieve a topographic map maximizing the unconditional information-theoretic entropy [Van Hulle (2000)].

Furthermore, the density estimate output by kMER can be written in terms of a mixture distribution where the kernel functions K_i represent the component Gaussian densities and the prior probabilities are all equal to $\frac{1}{M}$. This way, kMER provides an heteroscedastic and homogeneous mixture density model. Moreover, kMER log-likelihood can be computed just like in Equation 3.

3 SOM-based mode detection

Once we have trained the appropriate topographic map, suitable summaries of its structure will be extracted and analyzed to ascertain the modes' location. In particular, we here consider the Sammon's projection, the median interneuron distances, the data loads and a label matrix.

In order to check the goodness of the obtained maps, Sammon's projection [Sammon (1969)] can be employed. Indeed, to visualize high-dimensional SOM structures the use of Sammon's projection is customary. Sammon's map provides a useful global image while estimating all pairwise Euclidean distances among weight vectors and projecting them directly onto 2D space. Furthermore, by displaying the set of projections together with the connections between immediate neighbours, the degree of self-organization in the underlying topographic structure can be informally assessed in terms of the amount and nature of overcrossing connections in this image. Since it is not clear how much organization is possible for a given data set, the amount of connection overcrossing lacks an absolute scale for assessment. Sammon's map tends to provide informative and reliable images. In general, the higher the level of organization appreciated in it, the more the tendency to trust the conclusions derived from this topographic map.

Thus, since pointer concentrations in data space will tend to be maintained in the projected image, one can proceed to identify high-density regions directly on the projected SOM. This aspect of the analysis is rather important, because the main problem with the SOM structure, namely poor organization, needs to be controlled somehow. As usual, it is crucial to avoid poorly-organized structures (whereby immediate neighbours tend to be relatively distant from each other) but this goal is not so easy to reach when working with high-dimensional data [Kiviluoto and Oja (1998); Kohonen (2001)].

3.1 Median Interneuron Distances

Since we are interested in regions with higher weight vector density, the relative distance from each weight to its immediate neighbours on the network will provide a useful bit of information. Interneuron distance or proximity information has also been traditionally used for cluster detection in the SOM literature. Inspection of weight interdistances was pioneered by Ultsch [Ultsch (1993)], who defined the Unified matrix (U-matrix) to visualize Euclidean distances between weights in Kohonen's SOM. However, emphasis in the U-matrix is on cluster analysis, not on mode detection.

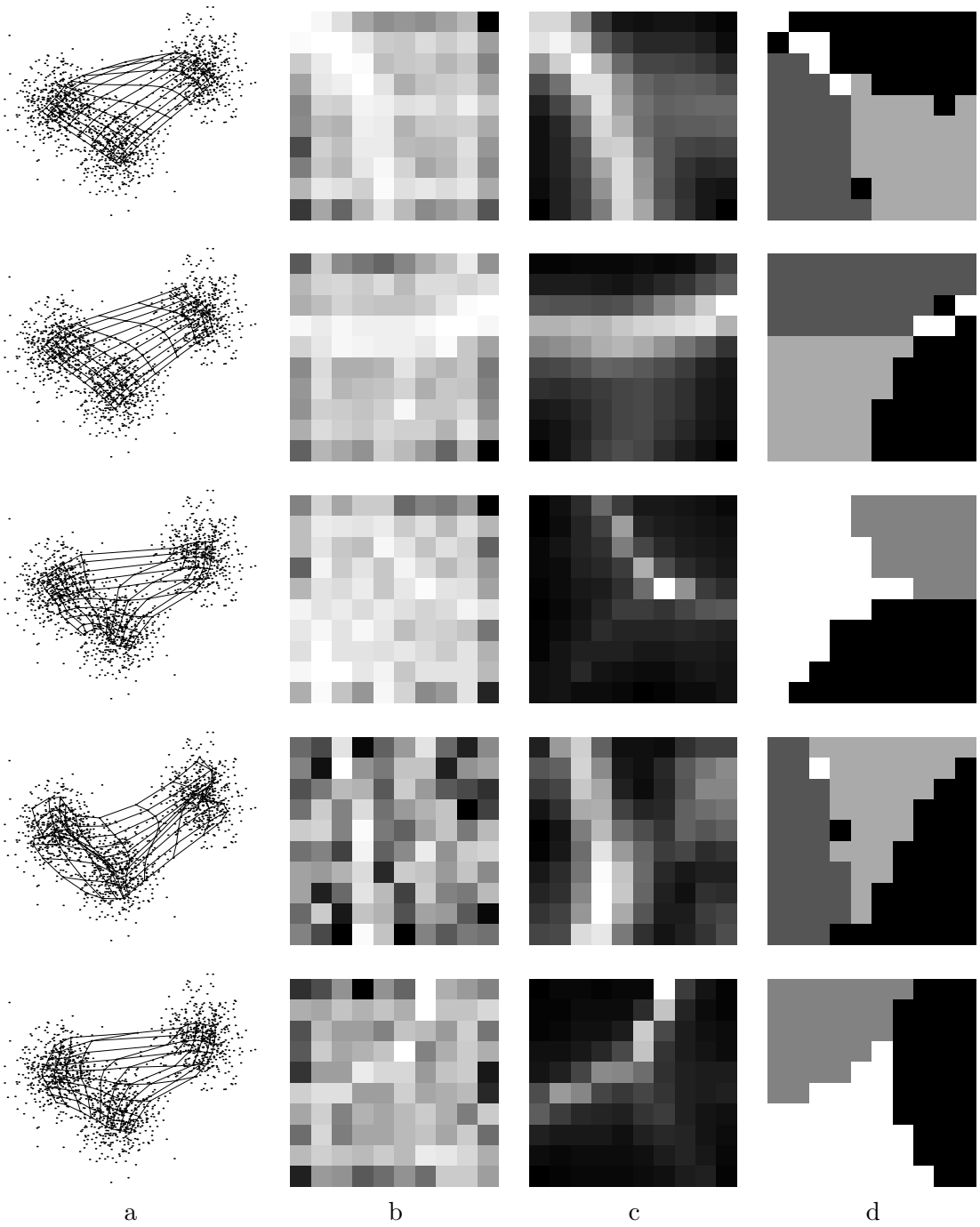

FIG. 1 – Performance on 3M-2D data set by sequential SOM (top), SOM-B (middle-up), SOM-Cx (middle) ; GTM (middle-down) and kMER (bottom) : (a) trained maps with data set highlighted ; (b) DL matrices ; (c) MID matrices ; (d) Labels matrices.

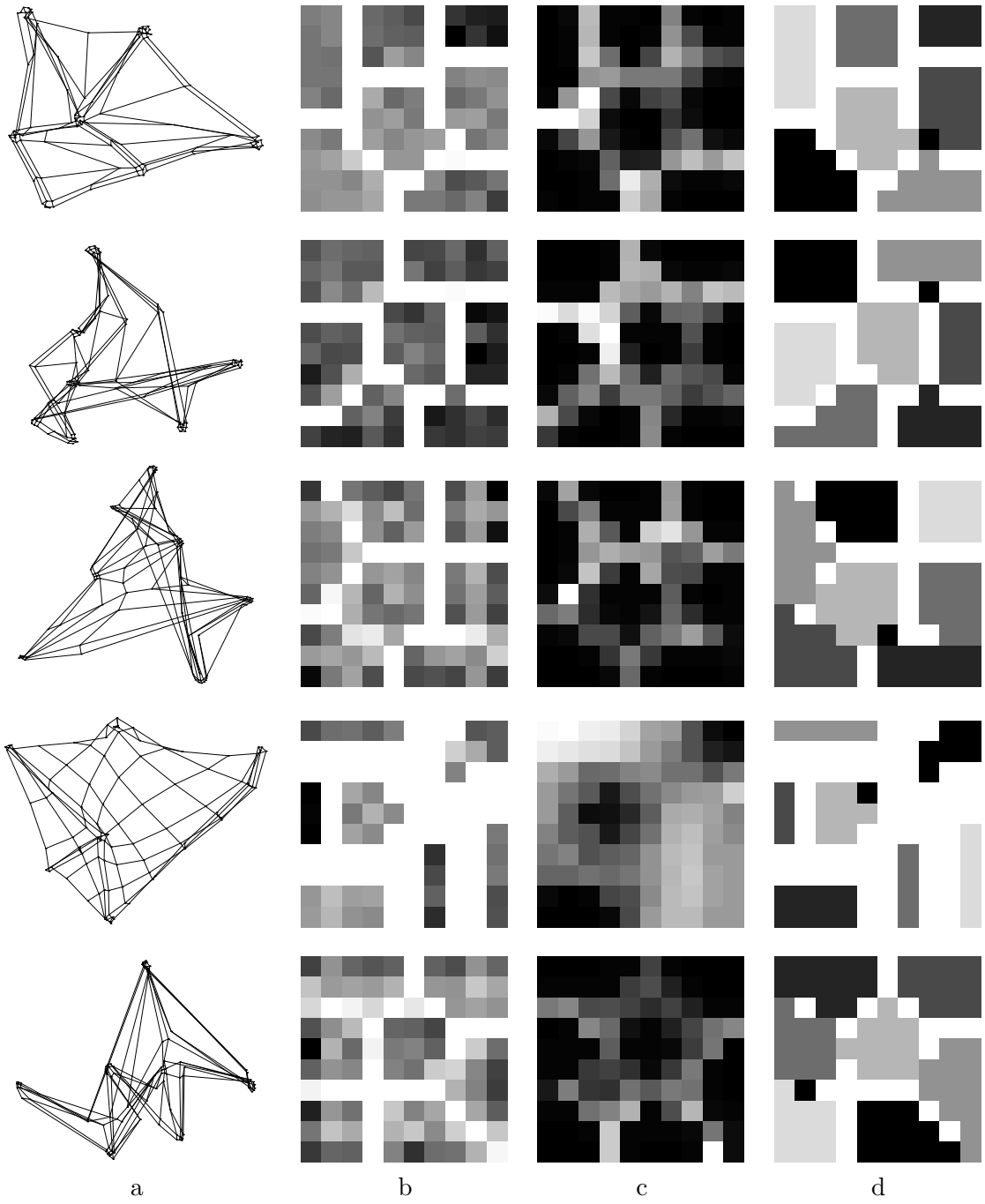

FIG. 2 – Performance on 7M-5D data set by sequential SOM (top), SOM-B (middle-up), SOM-Cx (middle) ; GTM (middle-down) and kMER (bottom) : (a) Sammon projected maps ; (b) DL matrices ; (c) MID matrices ; (d) Labels matrices.

An alternative Median Interneuron Distance matrix, say MID-matrix, was proposed in Muruzábal and Muñoz (1997) an used in Vegas-Azcárate and Muruzábal (2005) with successful results. Each MID entry is the median of the Euclidean distances between the corresponding weight vector $\mathbf{w}_i$ and all weights belonging to a star-shaped, fixed-radius neighbourhood typically containing eight units. The median can be seen as a conservative choice ; more radical options based on extremes can also be implemented. In order to facilitate the visualization of higher weight concentrations, a linear transformation onto a 256-tone gray scale is standard, where the lower the value, the darker the cell.

3.2 Dataloads

The dataloads are the natural estimate of the probability of activation, given new data are generated by the sampling distribution. Since the density of weight vectors in the trained topographic map should serve as an estimate of the density underlying the data, each neuron would cover about the same proportion of data. In this ideal case, a uniform DL-matrix should be obtained. In order to easily visualize the dataload distribution over the map, a gray image called DL-matrix is computed— in this case darker means higher.

To support the assessment, we consider a minimum Euclidean labelling scheme, in which each neuron is marked with the label that most occurs within its activation region. We denote this as the Label-matrix. Note that an extra label is needed for the case a neuron has no data projected into it.

3.3 A strategy for mode detection

The above summaries of the topographic maps structure constitute the basis of the following scheme for exploring mode estimation. To train the model we can follow the UDL criterion presented in Muruzábal and Vegas-Azcárate (2005). Special care must be taken with a well-known problem related to the SOM trained structure, namely the border effect. By this is meant that units on edges of the network do not stretch out as much as they should [Kohonen (2001)], which leads to confusing gray-scaled matrices on these map regions. Fortunately, these spurious concentrations rarely spread towards the interior of the network, although their traditional presence is somewhat annoying.

4 Synthetic examples

The experiments concern synthetic, balanced Gaussian mixtures of size $N = 1,500$ in total. The trimodal 2D data set is first analyzed. Next, a data set generated from a mixture of seven Gaussians, called 7M-5D, is studied. The maps discussed here have all been stopped early, following the ideas exposed in Muruzábal and Vegas-Azcárate (2005).

4.1 Three modes in 2D space

This data set is simulated from a mixture of three Gaussians, with two of the modes close enough to illustrate the finer detail in each algorithm. Figure 1 shows good organization and suitable DL matrices in all cases— albeit more uniform DL matrices can be seen in

the case of GTM and kMER. As a result, the three modes are correctly identified via MID analysis, yet we note that kMER and SOM-Cx provide the cleanest assignments.

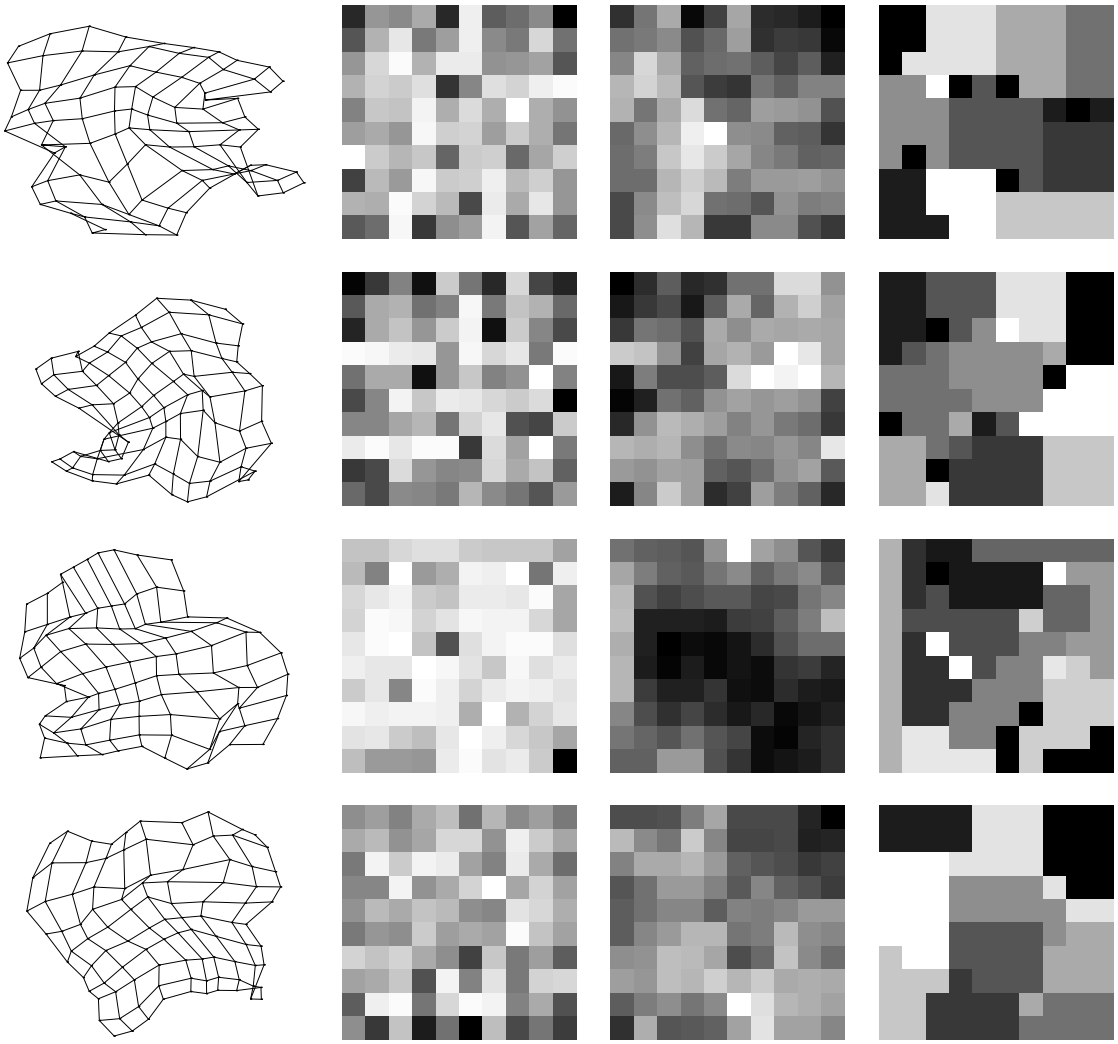

FIG. 3 – Performance on Mfeat data set by SOM-B (top), SOM (middle-up), GTM (middle-down) and kMER (bottom) : (a) Sammon projected maps ; (b) DL matrices ; (c) MID matrices ; (d) Labels matrices.

4.2 Seven modes in 5D space

Data are generated in a two-step process. First, we sample the locations of the Gaussian centroids, then we sample each Gaussian in term. All Gaussians are spherical and have the same size. The standard deviation of the centroid distribution is much larger than that of the data, so modes are well separated. Figure 2 shows a generally nice behaviour except in

the GTM case, where a relatively high number of dead units is unexpectedly observed. In all other cases, the seven modes are very clearly exhibited by the MID matrices.

5 A real-world example

We finally examine how UDL criterion does with a highly dimensional real-world example. We consider the Multiple Features database from the UCI repository [Blake and Merz (1998)] to validate the whole UDL approach. This data set (referred to as Mfeat) consists of features of handwritten numerals $(0, \ldots, 9)$, with 200 patterns per class for a total of 2,000 patterns. Hence, in this case $d = 649$, $N = 2000$ and we look for 10 decision classes. As Figure 3 shows, interestingly organized maps are obtained in all cases. These maps involve rather uniform DL matrices and result in pretty good mode predictions. GTM ranks worst in terms of cluster separation.

6 Summary and conclusions

We have revisited a universal approach to cluster detection based on the SOM structure and neuron proximity. We have shown that good results are obtained most of the time when maps are monitored (stopped early) via the UDL criterion introduced in Muruzábal and Vegas-Azcárate (2005). Early stopping seems indeed almost a requirement if the ultimate goal of the analysis depends on having a faithful approximation to the data-generating distribution. Other stopping criteria can be seen in Villmann et al. (1997).

Perhaps the Gaussian kernels in GTM are too constrained by the transformation from lattice to input space, for it appears that these kernels cannot move freely when needed at some point along the training process. We have also seen that many of the previous drawbacks are avoided by kMER, which produces more flexible and more effective maps, yet SOM-B and SOM-Cx seem also very well behaved and quite useful in the cases studied.

The scope of the above ideas for SOM-based biosignal clustering is important in as much as vectorial data keep on being worked out by researchers. Our results should be reassuring for practitioners following strategies based on neuron proximities, but they should also be recalled of the need to monitor map formation closely.

Références

Bauer, H.-U. and Pawelzik, K. (1992). Quantifying the neighborhood preservation of self-organizing feature maps, *IEEE Trans. Neural Networks*, **3** : 570–579.

Bishop, C. M., Svensén, M. and Williams, C. K. I. (1996). A fast EM algorithm for latent variables density models, in *Advances in Neural Information Processing Systems*, edited by In Touretzky, D., Mozer, M. C. and Hasselmo, M. E. E., vol. 8, pp. 465–471, MIT Press.

Bishop, C. M., Svensén, M. and Williams, C. K. I. (1997a). GTM : A principled alternative to the self-organizing map., in *Advances in Neural Information Processing Systems*, vol. 9, MIT Press.

Bishop, C. M., Svensén, M. and Williams, C. K. I. (1997b). GTM : The generative topographic mapping, *Neural Computation*, **10** : 215–235.

Blake, C. and Merz, C. (1998). UCI repository of machine learning databases, `http ://www.ics.uci.edu/mlearn/MLRepository.html`.

Davies, P. L. and Kovac, A. (2004). Densities, spectral densities and modality, *Annals of Statistics*, **32** : 1093–1136.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistics Society*, **B 39 (1)** : 1–38.

Ferrán, E. A. and Ferrara, P. (1991). Topological maps of protein sequences, *Biological Cybernetics*, **65** : 451–458.

Gray, A. G. and Moore, A. W. (2003). Nonparametric density estimation : Toward computational tractability, in *Proceedings of the Third International Conference on Data Mining, San Francisco, USA*, edited by Barbará, D. and Kamath, C.

Haese, K. and Goodhill, G. J. (2001). Auto-SOM : Recursive parameter estimation for guidance of self-organizing feature maps, *Neural Computation*, **13** : 595–619.

Hertz, J., Krogh, A. and Palmer, R. (1991). *Introduction to the Theory of Neural Computation, ch. 9*, Redwood City, CA. : Addison-Wesley.

Kaski, S. and Lagus, K. (1996). Comparing self-organizing maps, in *Proceedings of ICANN'96, International Conference on Artificial Neural Networks, Lecture Notes in Computer Science*, edited by von der Malsburg, C. and Sendhoff, B., vol. 1112, pp. 809–814, Springer, Berlin.

Kiviluoto, K. and Oja, E. (1998). S-map : A network with a simple self-organization algorithm for generative topographic mappings, in *Advances in Neural Information Processing Systems*, edited by Jordan, M. I., K. M. J. and Solla, S. A., vol. 10, pp. 549–555, MIT Press.

Kohonen, T. (2001). *Self-Organizing Maps*, Berlin : Springer-Verlag, 3rd extended ed.

Kohonen, T. and Somervuo, P. (2002). How to make large self-organizing maps for nonvectorial data, *Neural Networks*, **15** : 945–952.

Lampinen, J. and Kostiainen, T. (1999). Overtraining and model selection with the self-organizing map, in *Proc. IJCNN'99, Washington, DC, USA*.

Linde, Y., Buzo, A. and Gray, R. (1980). An algorithm for vector quantizer design, *IEEE Trans. Comunication*, **COM-28** : 84–95.

Muruzábal, J. and Muñoz, A. (1997). On the visualization of outliers via self-organizing maps, *Journal of Computational and Graphical Statistics*, **6(4)** : 355–382.

Muruzábal, J. and Vegas-Azcárate, S. (2005). On equiprobabilistic maps and plausible density estimation, in *5th Workshop On Self-Organizing Maps, Paris*.

Ritter, H. (1991). Asymptotic level density for a class of vector quantization processes, *IEEE Trans. Neural Networks*, **2(1)** : 173–175.

Ritter, H. and Schulten, K. (1986). On the stationary state of Kohonen's self-organizing sensory mapping, *Biological Cybernetics*, **54** : 99–106.

Sammon, J. (1969). A nonlinear mapping for data structure analysis, *IEEE Transactions on Computers*, **18(5)** : 401–409.

Scott, D. W. and Szewczyk, W. F. (2000). The stochastic mode tree and clustering, *Journal of Computational and Graphical Statistics*.

Ultsch, A. (1993). Self-organizing neural networks for visualization and classification, in *Information and Classification*, edited by In Opitz, O., L. B. and Klar, R. e., pp. 307–313, Berlin : Springer-Verlag.

Utsugi, A. (1997). Hyperparameter selection for self-organizing maps, *Neural Computation*, **9(3)** : 623–635.

Van Hulle, M. M. (1998). Kernel-based equiprobabilistic topographic map formation, *Neural Computation*, **10(7)** : 1847–1871.

Van Hulle, M. M. (2000). *Faithful representations and topographic maps : From distortion- to information-based self-organization*, New York : Wiley.

Vegas-Azcárate, S. and Muruzábal, J. (2005). On cluster analysis via neuron proximity in monitored self-organizing maps, in *Workshop on Biosignal Processing and Classification, Barcelona, Spain*.

Villmann, T., Der, R., Herrmann, M. and Martinetz, T. (1997). Topology preservation in self-organizing feature maps : Exact definition and measurement, *IEEE Trans. Neural Networks*, **8(2)** : 256–266.

Zheng, Y. and Greenleaf, J. F. (1996). The effect of concave and convex weight adjustements on self-organizing maps, *IEEE Transactions on Neural Networks*, **7(1)** : 87–96.