Density Estimation with Equiprobabilistic 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 - *Topographic maps have become standard tools for frequently encountered data-analytic tasks such as visualization, clustering and classification. Unfortunately, however, a complete SOM training methodology is not firmly established yet. For example, when using the standard SOM fitting algorithm, it is well-known that an appropriate choice for the end adaptation radius or final neighborhood width is crucial for obtaining useful results. This parameter controls in effect the degree of smoothness (or the amount of quantization error minimization) in the final map. Several heuristics have been suggested to guide this choice. A new monitorization idea is introduced in this paper, and a few examples are used to illustrate the scope of the approach.*

Key words - Self-organization, map selection, mode estimation, mixture models.

Acknowledgement - The authors are grateful to Drs. M. Van Hulle, T. Gautama and M. Svensén for useful discussions. Our research is funded by Spanish and European agencies. We also appreciate the support by the DMR Foundation's Decision Engineering Lab.

Note - This paper constitutes an extension of the one presented at the Workshop on Self-Organizing Maps (WSOM), Paris, France, 2005, *On Equiprobabilistic Maps and Plausible Density Estimation,* Jorge Muruzábal and Susana Vegas-Azcárate.

1 Introduction

With respect to the standard Kohonen algorithm, equiprobabilistic maps have been argued to provide a more faithful representation of the distribution generating the data [Van Hulle (1998, 2000). Indeed, these maps tend to minimize or eliminate the number of "dead" or "empty" units commonly found in standard fits. The motivating idea is that all trained neurons should have the same probability of being the best matching unit for a new data vector sampled from the same generating distribution. Here we consider two kernel-based variants of this type of map, namely, the generative topographic mapping (GTM) [Bishop et al. (1997) and the maximum entropy learning rule (kMER) [Van Hulle (1998)] algorithms. Just like in the case of the standard Kohonen algorithm, relatively organized structures are achieved during the early and intermediate stages of training, whereas more tangled-up structures (typically providing a slight improvement in quantization fit) are obtained when training continues all the way through the latter stages. Thus, an overfitting problem is also latent in equiprobabilistic maps. Van Hulle (2000)] has recognized the usefulness of certain form of monitorization and early-stopping policy to prevent damage to the organized map once it is achieved.

In this paper we investigate a novel, alternative scheme which is valid for any kind of equiprobabilistic map. Instead of looking directly at measures of self-organization [Haese and Goodhill (2001)], we care only about the extent to which equiprobabilism is achieved. Specifically, the key observation is that we can measure how close to equiprobabilistic a given map is, and we consider to stop training as soon as the map shows the first signs of equiprobabilism. This policy is seen to prevent excessive vector quantization indirectly. Further, the resulting density estimates are also seen to be rather accurate, in the sense that they recover the modes of the true generating density in many cases. We show that good results are obtained regardless of the particular training algorithm used.

The organization is as follows. Section 2 revisits the concept of equiprobabilistic map and Section 3 reviews a number of previous ideas on monitorization. The new criterion is presented in Section 4, and the experimental evidence is discussed in Section 5. Finally, Section 6 summarizes some conclusions and points out a few lines for further research.

2 Equiprobabilistic SOMs and density estimation

Inspired by a number of ordered mappings found in certain neural structures, the typical self-organizing map develops, in an unsupervised way, a mapping from a d-dimensional input space $V \subseteq \mathrm{IR}^d$, into an equal or lower-dimensional discrete lattice with regular, fixed topology [Kohonen (2001)]. For simplicity, in this paper we only consider 2D squared SOMs equipped with the standard topology. Although cluster analysis, data visualization and non-parametric regression are the most common applications, the SOM structure can also be regarded as a tool for generating non-parametric models of the sampling probability density, $p(v)$ say. Models of this sort can be termed explicit or implicit depending on whether a formal density estimate, say $\hat{p}(v)$, is available or not. When such a density is available, standard scoring measures can be readily used to assess its merit. When not, the areal *magnification factor* of a neural map can be used to measure model accuracy, see e.g. [Bauer et al. (1996)]. This is the exponent μ verifying $p(w_i) \propto p(v)^\mu$, where $p(w_i)$ is the (asymptotic) neuron weight density in input space. For the standard training algorithm, however, magnification factors

are only known in the simplest cases. For example, $\mu = \frac{2}{3}$ in the limit of an infinite density of neurons in a linear map [Ritter and Schulten (1986)]. For higher dimensional maps, the standard algorithm and its variants tend to show the same behaviour, that is, they tend to underestimate higher density regions and therefore fail to provide a *faithful* representation of the sampling distribution [Ritter and Schulten (1986); Lin et al. (1997)].

Various strategies have been explored to minimize this problem [Bauer et al. (1996)]. In this paper we focus on the idea of *equiprobabilistic* maps. These stick to the information-theoretic optimal $\mu = 1$, so that all neurons have (in the limit) the same probability to be maximally excited. While the issue was initially approached as a limit case of mutual information maximization, a more intuitive approach is perhaps to maximize the entropy of the map's outputs directly. Several unsupervised learning rules have been developed following this approach [Van Hulle (2000)]. Here we consider two specific algorithms leading to equiprobabilistic maps : GTM [Bishop et al. (1997)] and kMER [Van Hulle (1998)]. These kernel-based topographic maps can also be regarded as mixture density models, that is, the target $p(v)$ is approximated via density functions of the form

$$\hat{p}(v) = \sum_{i=1}^{M} q_i K_i(v, w_i, \sigma_i),$$

where M is the total number of neurons in the map, q_i are the mixing parameters, K_i is the kernel function allocated at the neuron weight or centroid w_i, and the radii σ_i determine the spread of the various mixture components. Most kernel-based topographic maps reported in the literature use Gaussian kernels K_i [Bishop et al. (1997); Van Hulle (1998); Yin and Allinson (2001); Heskes (2001)]. Both GTM and kMER use *homogeneous* mixtures ($q_i \equiv \frac{1}{M}$). As regards the σ_i, GTM uses a homoscedastic model ($\sigma_i \equiv \sigma$), whereas kMER allows different radii (which are adapted to the local input density together with the weight vectors or kernel centroids w_i). On the other hand, GTM incorporates an optional regularization term λ which can be added to the objective function (a standard likelihood function) to control the topological order in the mapping. This control is accomplished via a joint spherical Gaussian prior (with variance λ^{-1}) on the matrix W defining the non-linear manifold containing the centroids [Bishop et al. (1997)].

We use the fitted density estimate $\hat{p}(v)$ as the primary evaluation tool. Specifically, we look at the *modes* of this density as follows. Steepest ascent Hill-Climbing [Van Hulle (2000)] is used over the map lattice, that is, all neighbours are compared and the best is selected (until no further improvement is possible). Since the maps we handle are obtained via the UDL criterion and thus are rather smooth, we can explore the density surface in greater detail by using a star-shaped neighbourhood linking at most 8 neurons to any given unit. More conservative policies can be naturally considered too. Of course, many local maxima may be present, so that the process is restarted from all possible initial neurons. We illustrate below the (gray-valued) DENS-matrix, depicting the $\hat{p}(w_i)$ values at the lattice nodes, and the HC-matrix, highlighting all modes estimated by this procedure.

3 Previous work on map monitorization

The underlying motivations for some kind of monitorization of the training process are as follows. Firstly, the optimal map is that obtained using an "infinitely slow" training

process. Although good approximations can be derived for synthetic data with many cycles of training, this long training is not recommended in higher-dimensional real data spaces due to the enormous amount of time consumed by the algorithms and the impossibility of knowing when this "long enough" training has been achieved.

Secondly, for finer density estimation purposes (mode detection) and, more specifically, for its 2D visualization, the output lattice has to be disentangled and well-organized (just like in cluster analysis). Maps not free of topological defects could lead to contiguous high-density regions become split, whereas separated areas could be shown as a single region. Several heuristics have been suggested to guide this choice [Kaski and Lagus (1996); Villmann et al. (1997); Lampinen and Kostiainen (1999); Haese and Goodhill (2001)]. Still, Lampinen and Kostiainen (1999)] acknowledge that "the model complexity in SOM is usually chosen ad hoc by the user". In the case of the standard SOM, GTM and other algorithms, it has often been noted that the end adaptation radius or final neighborhood range has a big impact on the final maps obtained. If the final value of this range is too large, neurons will not properly span the input data set, but if it is too small, violations in topographic order will occur. Although it is commonly suggested that this radius should drop to "small" values by the end of the run, this is not appropriate in many cases [Kaski and Lagus (1996); Haese and Goodhill (2001)]. Similar considerations apply to kMER [Van Hulle (2000); Van Hulle and Gautama (2002)].

Hence, most training algorithms need to monitor somehow the degree of topology preservation achieved during training. What varies is the type of heuristic used to carry out this monitorization task. Some heuristics proceed by controlling certain amount of "overlap" among neurons, see e.g. [Van Hulle (2000)]. Others make use of some explicit topology preservation measure [Villmann et al. (1997); Haese and Goodhill (2001)]. Since the topographic product and other such measures do not involve the training data at all, they can be argued to have some basic limitations for the present task [Kaski and Lagus (1996); Villmann et al. (1997)]. Thus, Kaski and Lagus (1996)] propose a goodness of fit measure based on the first and *second* best matching units for a given input vector v ; a similar heuristic is implemented in the well-known SOM_PAK software. Going one step further, Lampinen and Kostiainen (1999) propose a *generalization* measure based on the disagreement between the projections of training and *test* data on the trained SOM. We see that these and related ideas differ mainly in the role played by the data. While the measure we introduce below is also based on the training data, it departs substantially from previous approaches. As noted later, however, approaches that consider the test data are also worth-studying in detail.

4 The UDL stopping policy

This paper explores the novel monitoring criterion based on the uniformity of the dataload vector, a criterion called UDL— Uniform DataLoad vector [Vegas-Azcárate and Muruzábal (2005); Muruzábal and Vegas-Azcárate (2005)]. Specifically, our UDL criterion looks for the moment at which the speed of decrease of the dataload standard deviation function is nearly zero. Pointer density in the trained equiprobabilistic map serves as an estimate of the density underlying the data. Thus, each neuron would cover about the same proportion of the data, leading to a uniform dataload vector. It appears that the stochastic Gaussian behaviour in the equiprobabilistic case can be approximately detected when it is first reached [Vegas-Azcárate and Muruzábal (2005); Muruzábal and Vegas-Azcárate (2005)].

Algorithm 1
UDL monitoring scheme.

Train the map with a constant neighbourhood range.
Store the obtained disentangled lattice, Q^0.
Starting from Q^0, perform one complete training with $\gamma^1 = 1$.
Determine the number of epochs, t^1_{udl}, and the corresponding range, $\sigma^1_{\Lambda,udl}$, for which the speed of decrease of SD^1 function is nearly zero.
repeat, for each monitoring run, $j > 1$,
Perform a new training, starting from Q^0, with $T^j = 2 \cdot t^{j-1}_{udl}$ and

$$\gamma^j = -\frac{\ln \frac{0.9 \cdot \sigma^{j-1}_{\Lambda,udl}}{\sigma_{\Lambda(0)}}}{2 \cdot \sigma_{\Lambda(0)}} \tag{1}$$

to cool at a slower rate, but only run the simulation as long as necessary. Determine the epoch, t^j_{udl}, and the range, $\sigma^j_{\Lambda,udl}$, for which the speed of decrease of SD^j function is nearly zero.
until $\sigma^j_{\Lambda,udl} \simeq \sigma^{j-1}_{\Lambda,udl}$.
Do a complete run with $T_{udl} = t^j_{udl}$ and $\gamma_{udl} = \gamma^j$.

Hence, the associated UDL stopping rule could be stated as follows : quit as soon as the trained map shows the first signs of having reached the reference Gaussian DL distribution—a moment referred to as the UDL stage. In other words, quite as soon as SD function reaches its stability.

Furthermore, minor gains in quantization error brought by training beyond the UDL stage seem to enforce the loss of useful organization, implying fuzzier displays for analysis. Indeed, the UDL stage also signals approximately the beginning of the fine-tuning phase in quantization error [Vegas-Azcárate and Muruzábal (2005); Muruzábal and Vegas-Azcárate (2005)]. Note that during the learning process, the best matching unit of each datum is calculated, so the overhead in obtaining the dataloads and its standard deviation is minimal.

The problem now is to obtain t_{udl} or σ^{udl}_{Λ}, since a complete long enough training has to be done. Indeed, this is not possible with real-world data where 'long enough' is not known. To solve the 'long enough' problem, a scheme similar to the one presented in [Van Hulle (2000)] is developed here. In it, the rate at which the neighbourhood function range decreases is adjusted during training, ensuring that the final range value is the one desired, σ^{udl}_{Λ}. Hence, in a finite and affordable number of cycles the same optimal map as the one achieved in an infinitely slow training is obtained. The neighbourhood cooling scheme is developed following,

$$\sigma_{\Lambda}(t) = \sigma_{\Lambda}(0) \cdot \exp\left(-2 \cdot \sigma_{\Lambda}(0) \cdot \frac{t}{T_{udl}} \cdot \gamma_{udl}\right), \tag{2}$$

where γ_{udl} and T_{udl} provide the map that has a range value of σ^{udl}_{Λ}. Algorithm 1 reflects the proposed UDL monitoring scheme.

The monitoring processes converge, since in a slow enough training the neighbourhood

range value at which SD function reaches its stability does exist [Vegas-Azcárate and Muruzábal (2005); Muruzábal and Vegas-Azcárate (2005)]. Hence, if the training algorithm employs a neighbourhood function to ensure topographic ordering, as in the case of SOM-like and kMER algorithms, it is possible to use UDL monitoring scheme.

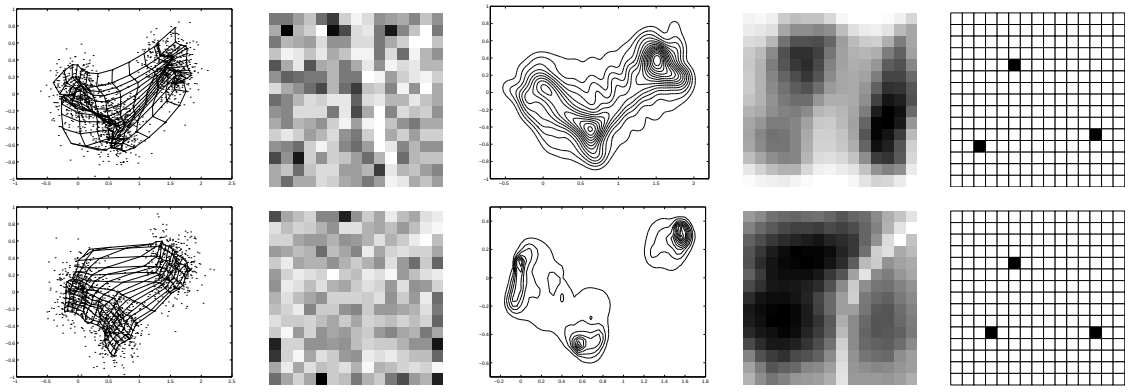
5 Density-based approach

This section shows how to use the density estimate $p(\mathbf{w}_i)$ as the primary tool to extract a structure from a UDL-monitored topographic map. Focusing on the HC-winning neurons and the highest density regions should be helpful to approach a precise mode estimation, provided that the map is appropriately organized. A gray-valued density matrix depicting $p(\mathbf{w}_i)$, the density values at the lattice nodes, is used to visualize the estimated density, namely DENS-matrix. In this matrix darker means larger.

A kernel-based density estimate can be computed by positioning equivolume Gaussian kernels at the neuron weights. For the topographic map developed by kMER, these kernels will be variable in width, proportional to the neuron radii σ_i. The standard and batch SOM will be developed with both fixed and variable kernel density estimation. The widths in the variable approach will be computed following the k^{th} nearest-neighbour method, in which a hypersphere is centered at the neuron weights with a radius chosen in such a way that it contains the k^{th} nearest-neighbour input samples. For selecting k-value, we will follow the next scheme. During the learning of kMER algorithm, kernel radii are adjusted in such a way that, at convergence, the probability for neuron i to be active is $\frac{\rho}{M}, \forall i$, with ρ the scale factor. Since kMER neuron weights 'capture', at convergence, $\frac{\rho}{M} \cdot N$ inputs, we will take $k = \frac{\rho}{M} \cdot N$ nearest-neighbour input samples for computing the variable kernel density estimation with SOM and SOM-batch. For the fixed kernel approach, all σ_i are set to 0.1. The optimal degree of smoothing, ρ_s, will be determined as that that maximize the likelihood over the data set with the current weights.

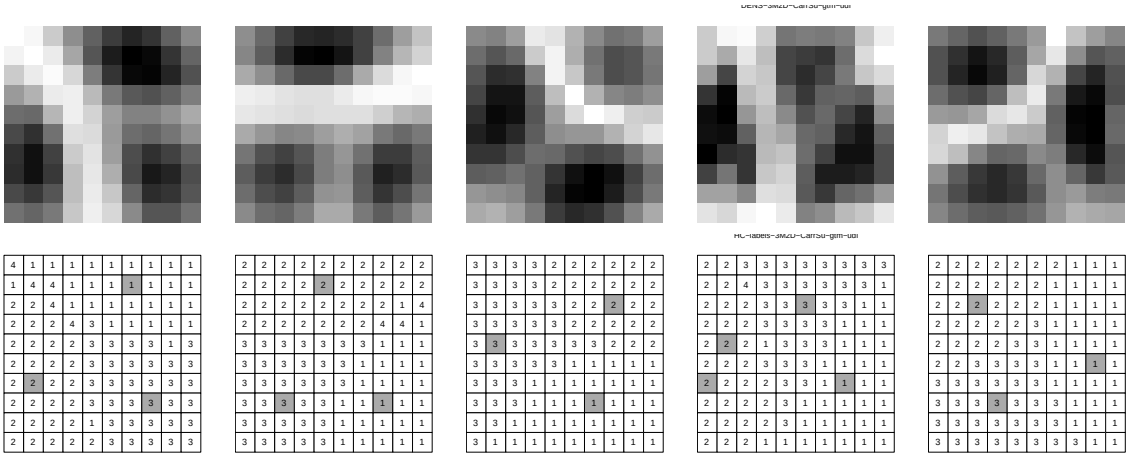
Steepest ascent Hill-Climbing (HC)— a graph search algorithm where the current path is extended with a successor node which is closer to the solution than the end of the current path [Howe (2005)]— is developed on the network structure. This way, all neighbours are compared and the best is selected, until no further improvement is possible. Since the maps we handle are obtained via the UDL criterion and thus are rather smooth, we can explore the density surface in greater detail by using a star-shaped neighbourhood linking at most 8 neurons to any given unit. More conservative policies can be naturally considered too. Of course, many local maxima may be present, so that the process is restarted from all possible initial neurons. This way all local maxima are detected, each of them corresponds in principle to a different mode.

The Hill-Climbing standard approach requires a spatially continuous function since a density estimate is needed, yielding to a high computational complexity. Van Hulle (2000) proposed to determine density estimate only at the M weight vectors, guaranteing a local maximum in a finite number of steps, always smaller than M. This procedure follows next scheme. Let dispose of the input density estimates at weight vector $\mathbf{w}_i$, allocate there a hypersphere that contains the ι nearest-neighbour weights, choosing its radius in an appropriate way [Van Hulle (2000)].



a b c d e

FIG. 1 – Performance on the trimodal data set by GTM (top) and kMER (bottom) : (a) trained map with data set highlighted ; (b) DL-matrix ; (c) 2D isolines generated from the density estimate ; (d) DENS-matrix ; (e) HC-matrix.



a b c d e

FIG. 2 – 3M-2D data set, DENS-matrix (top) and HC-Labels-matrix (bottom). (a) SOM ; (b) SOM-batch ; (c) SOM-cx ; (d) GTM ; (e) kMER.

The neuron with the highest density estimate of this $\iota + 1$ neurons in the i^{th} hypersphere is called the top-weight, say weight $\mathbf{w}_j$. It is said that weight $\mathbf{w}_i$ points to $\mathbf{w}_j$. Interesting weights are those top-weights that point towards themselves, namely, the ultimate-top-weights. This weights belong to separate clusters, since their density estimates represent local maxima in the input density. Hence, the number of modes in the density estimate is obtained, for a given choice of the free hypersphere parameter ι.

Although this algorithm does not require a spatially continuous function, the ability to build continuous density functions will tend to provide more sensible results. HC-matrix highlights all estimated modes by the previous HC routine. In simulations, this set of local maxima is contrasted with the set of true modes in the data-generating density.

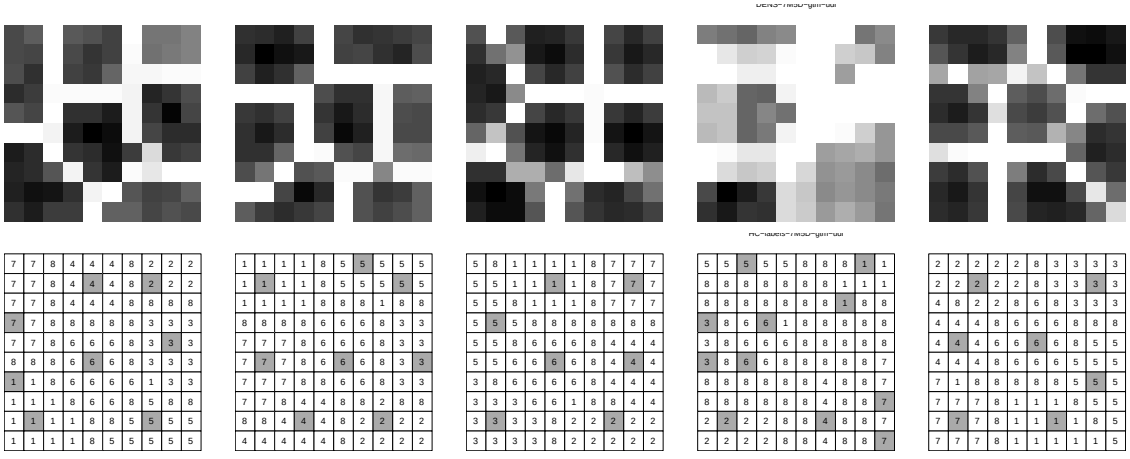
6 Experimental results

First, some artificial data sets of varying nature concerning balanced Gaussian mixtures are investigated. A 2D trimodal and 5D seven-modal data sets, together with a 10D bimodal, a 50D unimodal, a 50D four-modal and a 50D five-modal data sets are explored. As pointed before, in all high-dimensional cases the maps projected via Sammon are shown. By displaying this set of projections together with the inherited connections between immediate neighbours, the degree of self-organization in the underlying SOM structure can be informally assessed. The maps discussed here have all been monitorized following UDL criterion.

6.1 Three modes in 2D

Trimodal 2D 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. Good results are obtained using GTM (Figure 1 top plots). A precise approximation to the true modes' locations is obtained in the sense that each of the highlighted units in Figure 1e is also the closest to one of the three modes. As regards kMER, we see a mild border effect in Figure 1b bottom plot, yet mode estimation is very precise again in Figure 1e. Very different densities arise in Figures 1c, though, kMER appearing better suited to describe the larger gap in these data.

In Figure 2 the estimated density matrices of standard SOM, SOM-bacth, SOM-cx, GTM and kMER algorithms are shown. HC procedure is highlighted together with the label of each neuron. An accurate approximation to the true modes' locations is obtained in all cases. However, GTM shows a false positive in the second component (see Figure 2d).



a b c d e

FIG. 3 – 7M-5D data set, DENS-matrix (top) and HC-Labels-matrix (bottom). (a) SOM; (b) SOM-batch; (c) SOM-cx; (d) GTM; (e) kMER.

6.2 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 3 shows the estimated density matrices of the five algorithms. The seven components of the mixture are well identify in all DENS-matrices. Moreover, HC procedure highlights each mode of each component. Note that standard SOM and SOM-batch trained maps show one false positive, in the first and fifth components, respectively (see Figures 3a,b). HC-matrix of trained GTM algorithm shows four false positives, in the first, third, sixth and seventh component. On the contrary, the UDL-monitored maps of SOM-cx and kMER algorithms obtain a precise estimation to the seven modes' locations.

6.3 Two modes in 2D space

Next, a data set generated from a mixture of two 10D Gaussians with modes relatively apart from each other, called 2M-10D, is studied.

Figures 4 and 5 show the analysis based on early-stopped and fully-trained maps in this simple high-dimensional problem. As it concerns GTM, good results are obtained at UDL-stage (top plots of Figure 4). A well-organized map is reflected, over which the density-based approach provides an excellent estimation of location : the two HC winners correspond exactly to the two neurons closest to the real modes. As GTM continues all the way to the end (Figure 4 bottom plots), the organization seems weaker, leading to a diffuse DENS-matrix— the density-based approach is spoiled by too many local optima (Figure 4d). Indeed, as highlighted before, the higher the level of organization appreciated in its Sammon's map, the more we tend to trust the conclusions derived from this topographic map.

Figure 5 illustrates the analysis of the two maps obtained using kMER. Accurate mode estimation is obtained again at UDL-stage (Figure 5 top plots), where a well-organized map and a rather uniform DL-matrix are achieved. The density-based approach pinpoints the modes' locations exactly (Figure 5d). By the end of training (Figure 5 bottom plots), however, the density estimate is somewhat diluted— being difficult to make a distinction between correct estimations and noise (Figure 5d).

6.4 Single mode in 50D

The single 50D spherical Gaussian centered at the origin is considered in Figure 6. Whenever the training data exhibits a single mode at the origin, a gray-scaled matrix based on the $\| \mathbf{w}_i \|$ norms is used to assist the assessment of the strategy ; this is termed the Distance from each Pointer to the Origin or DPO-matrix. Note that, even when faced in isolation, Gaussian modes are increasingly harder to notice in higher dimensions.

In the case of GTM (Figure 6 top plots), the map provides very precise information : the density estimate is unimodal, and the neuron with the highest density is also the closest to the origin— indeed, note the similarity between Figures 6c and 6d. Sammon's projection suggests that a good deal of organization is obtained in this case. Turning now to kMER (Figure 6 bottom plots), we see that a well-organized map is obtained in this case. Indeed, the quality of density estimation is accurate again.

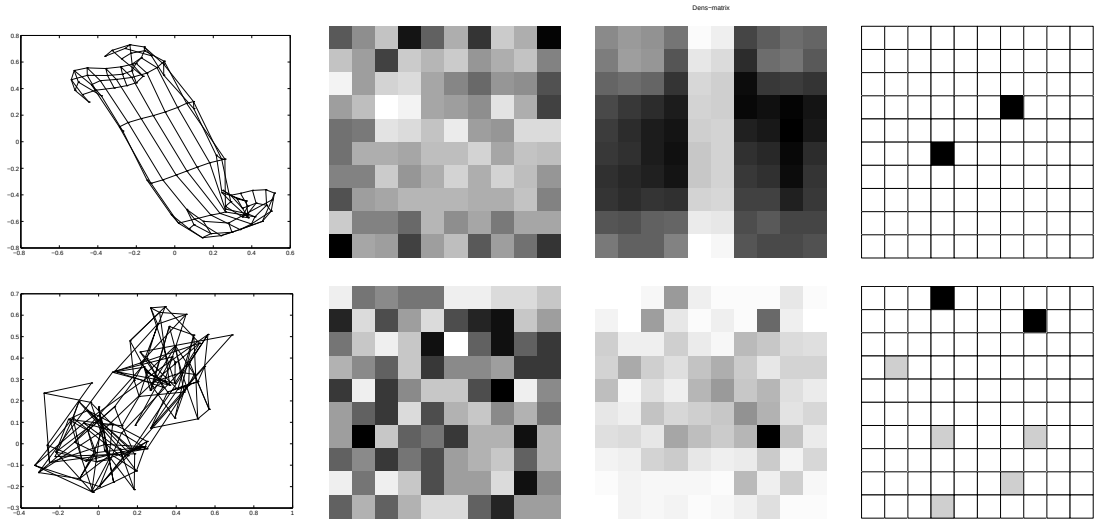


FIG. 4 – GTM on 2M-10D data set. A partially trained map (top) and the result of full training (bottom). (a) Sammon's projected map ; (b) DL-matrix ; (c) DENS-matrix ; (d) HC-matrix showing true modes (solid squares) and false positives (gray squares).

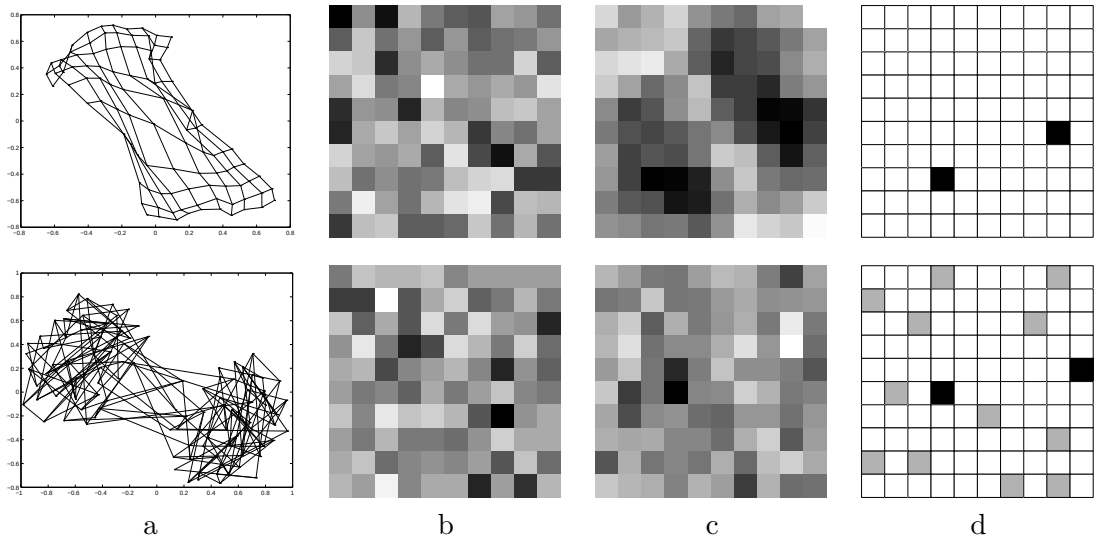


FIG. 5 – kMER on 2M-10D data set. Interpretation is as in Figure 4.

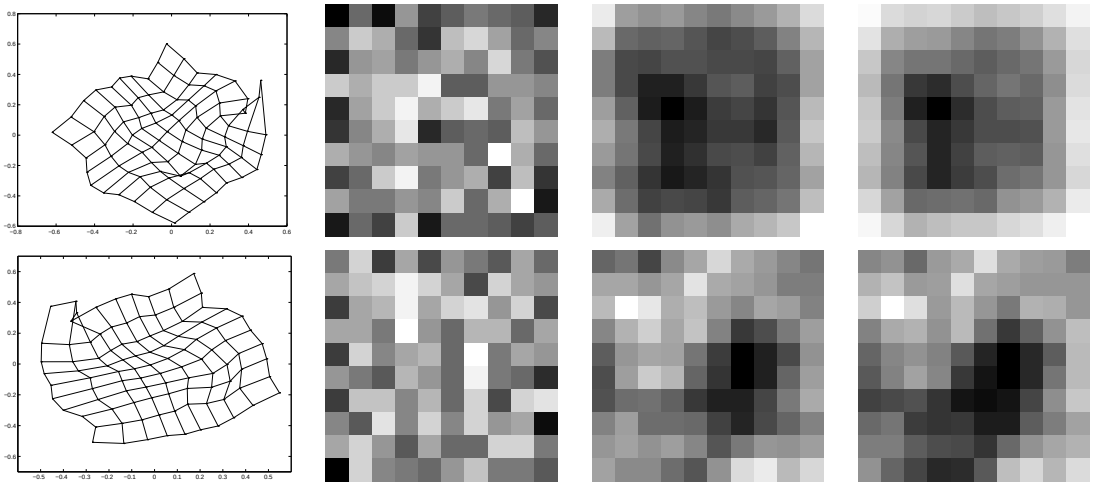


FIG. 6 – Early stopped maps on the single-mode problem 1M-50D data set (top : GTM, bottom : kMER). (a) Sammon's projected map ; (b) DL-matrix ; (c) DPO-matrix ; (d) DENS-matrix.

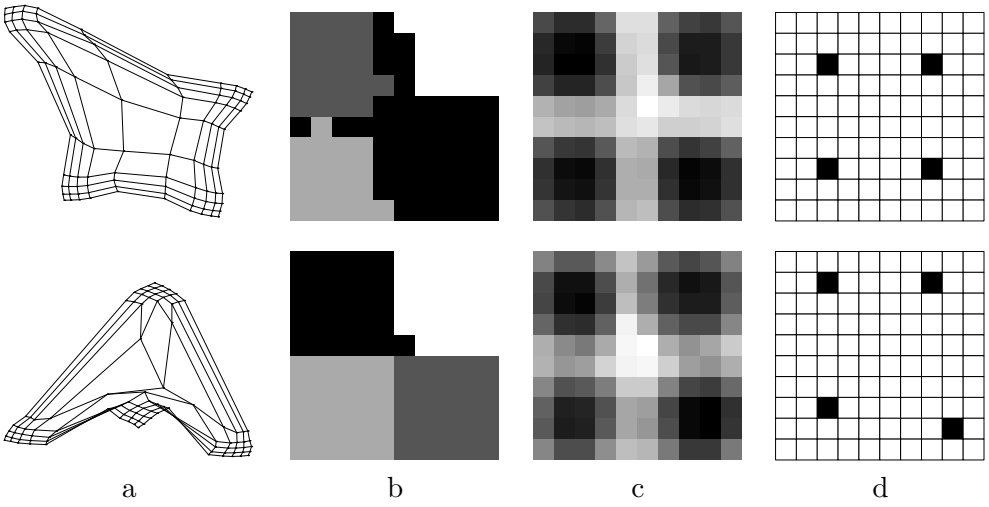


FIG. 7 – Performance on 4M-50D data set by sequential SOM (top) and kMER (bottom). (a) Sammon's projected map ; (b) LAB matrix ; (c) DENS matrix with Variable Kernel approach ; (d) HC matrix with Variable Kernel approach.

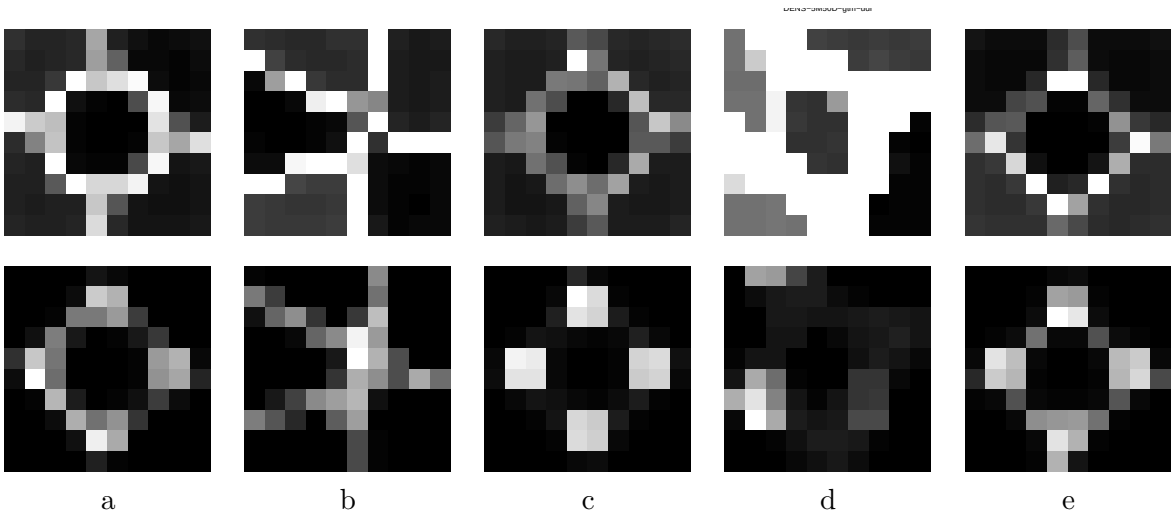


FIG. 8 – 5M-50D data set. Density matrices (top), MID matrices (bottom) performed by (a) sequential SOM ; (b) SOM-batch ; (c) SOM-cx ; (d) GTM ; (e) kMER.

6.5 Four modes in 50D space

The data set consists of four 50D spherical Gaussians, namely 4M-50D, with all the mixture components well-separated, although one of them is more separated from the others. Figure 7 presents the density-based mode detection approach developed over sequential SOM and kMER UDL monitored maps.

Top plots of Figure 7 show SOM UDL-monitored map. Despite of the high dimensionality, the underlying SOM structure presents a high degree of self-organization, since the amount of overcrossing connections in the respective Sammon's map is closely null. Label and DENS matrices reflects the existence of four separated clusters. Further, HC-matrix point out the existence of the four modes.

Bottom plots of Figure 7 illustrate kMER optimal map, showing a completely well-organized Sammon's projection. A precise Label-matrix is observed, with few nonactive neurons in the low density area. When looking to the Sammon's map and DENS-matrix, one can conclude the existence of four components in the mixture. Moreover, HC-matrix highlights the four modes.

6.6 Five modes in 50D space

Before jump to a real-world example, we consider a mixture of five Gaussians in 50D. Figure 8 shows how, when dealing with highly dimensional data sets, the multimodality is better analyzed with DENS-matrix than with MID-matrix. Note that top plots of Figure 8 have more intermediate gray values than the bottom plots (where most of the cells are just black or white).

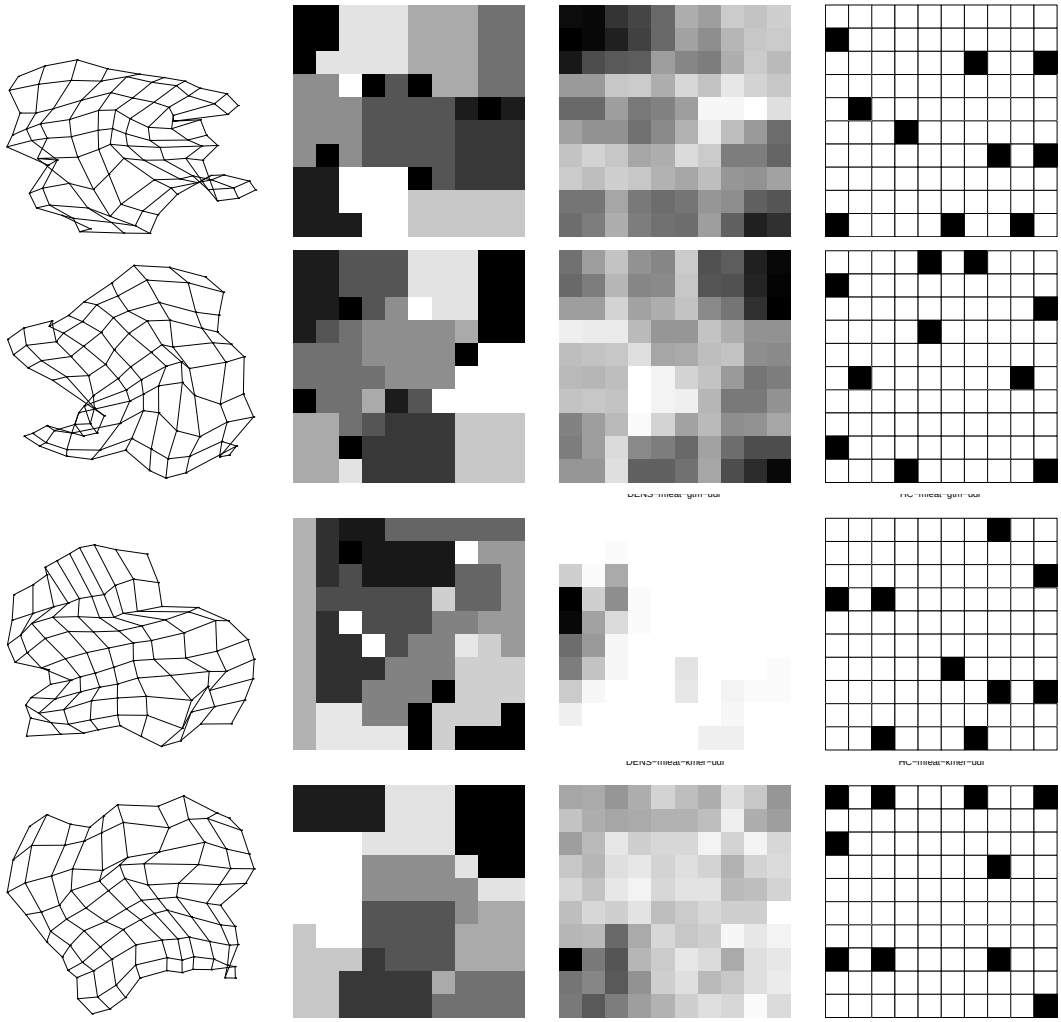


FIG. 9 – Performance on Mfeat data set by sequential SOM, SOM-batch, GTM and kMER, from top to bottom. (a) contour plot ; (b) label matrix ; (c) DENS matrix with Variable Kernel approach ; (d) HC matrix with Variable Kernel approach.

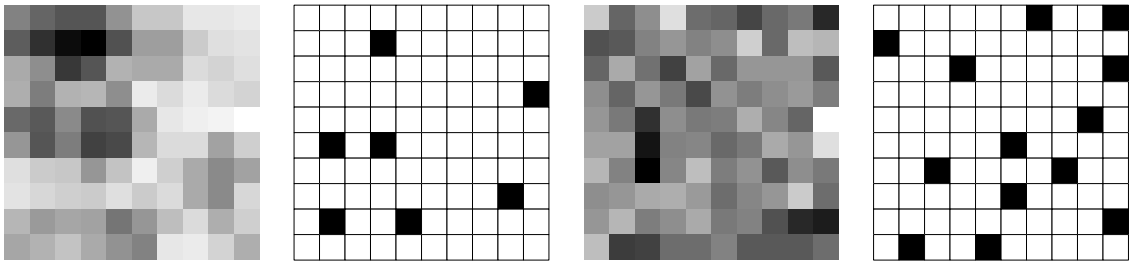


FIG. 10 – Performance on Mfeat data set by sequential SOM and SOM-batch. (a,c) DENS matrix with Fixed Kernel approach ; (b,d) HC matrix with Fixed Kernel approach.

6.7 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. Figure 9 shows UDL monitored maps of sequential SOM, SOM-batch, GTM and kMER learning rules. Sammon maps are nearly-perfect organized— some corners are a bit twisted but note that we are in a 649th dimensional space. The labelled matrices split the region in 10 accurate and precise areas. Although the four algorithms present well-defined labelled matrices (only the edges of the 10 areas could lead to misclassification problems), kMER optimal map seems the best, with the best organized Sammon's projection and the clearest matrix of labels. In SOMs some nonactive neurons have appeared. kMER DENS-matrix, as well as the Variable Kernel approach to SOM Batch and sequential SOM provide an accurate Hill-Climbing procedure, which places a mode in each of the clusters marked by the corresponding labelling matrices.

It can be observed that, as in the low dimensional case, the Fixed Kernel approach reflects well the nature of the data set, but in a stiffer way than the Variable approximation. Indeed, the drawbacks of Fixed Kernel approach are patent in Figure 10, where the addition of false positives or the remove of true ones from HC-matrices is appreciated.

7 Summary and conclusions

We have introduced a new monitorization idea for equiprobabilistic maps and we have tested it using two particular learning rules, GTM and kMER. Early stopping seems indeed very appropriate (almost a requirement) if the ultimate goal of the analysis depends on having a faithful approximation to the data-generating distribution. The proposed UDL criterion is easy to apply and has provided sensible answers in our test problems. Hence, the benefits of the new criterion should be more carefully evaluated against other methods. We have briefly discussed here other such stopping criteria based on the resulting density estimate. Further research is needed on this point. Specifically, various ideas related to other uses of data combined with the predictive densities deserve further study.

Références

Bauer, H. U., Der, R. and Hermann, M. (1996). Controlling the magnification factor of self-organizing feature maps, in *Neural Computation*, vol. 8, pp. 757–771.

Bishop, C. M., Svensén, M. and Williams, C. K. I. (1997). 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`.

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

Heskes, T. (2001). Self-organizing maps, vector quantization, and mixture modeling, *IEEE Trans. Neural Networks*, **12(6)** : 1299–1305.

Howe, D. (2005). FOLDOC : Free on-line dictionary of computing., `http ://www.nightflight.com/foldoc-bin`.

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.

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

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

Lin, J. K., Grier, D. G. and Cowan, J. D. (1997). Faithful representation of separable distributions, *Neural Computation*, **9** : 1305–1320.

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

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

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.

Van Hulle, M. M. and Gautama, T. (2002). Monitoring the formation of kernel-based topographic maps with application to hierarchical clustering of music signals, *J. VLSI Signal Processing Systems for Signal, Image, and Video Technology*, **32** : 119–134.

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.

Yin, H. and Allinson, N. (2001). Self-organizing mixture networks for probability density estimation, *IEEE Transactions on Neural Networks*, **12(2)** : 405–411.