Mode estimation with topographic 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 - *The paper reviews thoroughly a variety of issues related to mode estimation. The potential of self-organizing maps as an approach to mode detection is inquired here. The batch version of the standard SOM and a convex adjustment of it are compared with two kernel-based learning rules, namely, the generative topographic mapping and the kernel-based maximum entropy learning rule. A strategy for mode detection based on the previous topographic maps are tested on a number of synthetic gaussian data, as well as in a highly dimensional real-world example.*

Key words - self-organizing maps, mixture models, mode detection

1 Introduction

The problem of multivariate mode estimation has been treated from many points of view. Popular ideas such as transforming it into a mixture problem Izenman and Sommer (1988); Scott (1992) or developing kernel-based density estimates Good and Gaskins (1980); Terrell and Scott (1992); Gray and Moore (2003); Davies and Kovac (2004) seem to provide generally good results, but drawbacks in higher dimensions should occur Scott and Szewczyk (2000). Here we inquire about the potential of *self-organizing maps* (SOM) Kohonen (2001) as an approach to multivariate mode detection. Inspired by a number of ordered mappings found in certain neural structures, the self-organizing map develops, in an unsupervised way, a mapping from a d-dimensional input space $V \subseteq \Re^d$, into an equal or lower-dimensional discrete lattice with regular, fixed topology Kohonen (2001). In contrast to neurons in the well-known feed-forward multilayer network Cheng and Titterington (1994), all SOM units receive the same input information and produce their output in parallel. Thanks to a simple competitive learning process — whereby only weights connected to the winner (or best matching unit) and its neighbors are updated —, the SOM structure is often topologically ordered. This entails the fundamental property for clustering purposes, namely, that nearby input patterns stimulate nearby output units. The SOM structure derived from the standard fitting algorithm is often found useful for clustering, visualization and other purposes. However, it lacks a statistical model for the data. Recent approaches to SOM training usually incorporate some statistical notions yielding richer models and more principled fitting algorithms.

The organization of the paper is as follows. In section II four topographic map formation algorithms are described. Section III presents a strategy for mode detection as well as the basic tools needed for it. In section IV an exhaustive simulation study based on synthetic and real-world data is developed. Conclusions and further analysis are treated in section V.

2 SOM training algorithms

As we pointed above, the SOM structure lacks a statistical model for the data. The following approaches to SOM training, two variants of the sequential SOM algorithm and two kernel-based learning rules algorithms, incorporate statistical notions yielding richer estimated models.

As often acknowledged Kohonen (2001), the standard SOM training algorithm suffers from some shortcomings : the absence of a cost function, the lack of a theoretical basis to ensure topographic ordering, the absence of any general proofs of convergence, and the fact that the model does not define a probability density. The Generative Topographic Mapping (GTM) proceeds by optimizing an objective function via the EM algorithm Dempster et al. (1977), so the convergence to a local maxima is guaranteed. Since the cost function is of log-likelihood type, a measure is provided on which a GTM model can be compared to other generative models.

While the conditions under which the *self-organization* of the SOM occurs have not been quantified and empirical confirmation is needed in each case, the neighborhood-preserving nature of the GTM mapping *is an automatic consequence of the choice of a continuous function* $\mathbf{y}(\mathbf{x}, \mathbf{W})$ (see below) Bishop et al. (1997). In the same way, the *smoothness* properties of the original SOM are difficult to control since they are determined indirectly by the neighborhood function, while basis functions parameters of the GTM algorithm explicitly govern the

smoothness of the manifold (see below).

Hence, the GTM algorithm seeks to combine the topology preserving properties of the SOM structure with a well defined probabilistic framework. Moreover, since the evaluation of the Euclidean distances from every data point to every Gaussian centre is the dominant computational cost of GTM and the same calculations must be done for Kohonen's SOM, each iteration of either algorithm takes about the same time.

In the case of the standard SOM algorithm, the receptive fields are the standard Voronoi regions defined by the minimum Euclidean distance rule, say,

$$i^* = \arg\min_i \{\|v - w_i\|\}$$

As we shall see in detail below, the Kernel-based Maximum Entropy learning Rule (kMER) differs from the original SOM algorithm in the criterion optimized, i.e., entropy maximization *vs.* distortion minimization, the type of receptive fields used, i.e., overlapping kernel-based receptive fields *vs.* non-overlapping Voronoi receptive fields, and the quality of performance in density estimation, i.e., more equitable weight distribution *vs.* output unit under-utilization problems. It can be seen in kMER algorithm that the receptive field weight centers and its radii 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 represent the component Gaussian densities with equal prior probabilities, providing an heteroscedastic, homogeneous mixture density model Van Hulle (2000), whose log-likelihood function can be computed just like in the GTM case (see Eq. 1 below).

Now the four training algorithms tested in the paper are described in detail, but first some notation is presented.

Basic notation	
w_i	network weight vector
v_m	input data vector
x_i	latent space node
t	training time index
$p(v)$	data-generating density
$p(w_i)$	weight density
d	input space dimensionality
K_i	kernel function centered at w_i
σ_i	mixture component std
i^*_m	best matching unit for v_m
H_{ij}	neighborhood function
$\sigma_H(t)$	neighborhood range
r_i	lattice coordinates of neuron i
η	learning rate
$\ln \ell$	log-likelihood function

2.1 SOM batch

The batch version of Kohonen's SOM training algorithm Kohonen (2001, 1993), say

$$w_i = \frac{\sum_{m=1}^{M} v_m H_{i_m^* i}}{\sum_{m=1}^{M} H_{i_m^* i}},$$

resembles the algorithm in Linde et al. (1980) and thus relates to the standard K-Means algorithm as well as to Heskes approach Heskes (2001). A usual choice is a Gaussian-shaped neighborhood function, with a monotonically decreasing range function. Since SOM_batch contains no learning rate parameter, no convergence problems arise, and more stable asymptotic values for the w_i's are obtained Kohonen (2001). For simplicity, in this paper we consider 2D SOMs only ; besides, we restrict consideration to squared maps equipped with the standard topology. Thus, we leave for future work any refinement that might follow from modifying basic aspects of the SOM structure (dimension, topology or shape).

2.2 Convex adjustment (SOM_Cx)

A convex adjustment for the original SOM algorithm has been studied by Zheng and Greenleaf Zheng and Greenleaf (1996). They actually present two nonlinear models of weight adjustments to obtain desirable densities of output units. One of the models approaches the probability distribution $p(v)$ of the inputs using a *convex* model to adjust weights. This is seen to provide more efficient data representation for vector quantization, whereas the convergence rate is comparable to that of the linear model. In this case, the standard competitive learning rule

$$\Delta w_i = \eta H_{i_m^* i}(v_m - w_i),$$

becomes

$$\Delta w_i = \eta H_{i_m^* i}(v_m - w_i)^{\frac{1}{\kappa}},$$

where κ is a positive odd integer Zheng and Greenleaf (1996) and η is a learning rate (which can also be a monotonically decreasing function of time Kohonen (2001)).

2.3 Generative Topographic Mapping (GTM)

GTM Bishop et al. (1997) defines a non-linear, parametric mapping $\mathbf{y}(x, \mathbf{W})$ from an L-dimensional latent space to a d-dimensional data space, where $L < d$. By suitably constraining the model to a lattice in latent space, a posterior distribution over the latent grid is readily obtained using Bayes' theorem for each data point. More specifically, GTM training is based on a standard EM procedure aimed at the standard Gaussian-mixture log-likelihood Bishop et al. (1997, 1996)

$$\ln \ell = \sum_{m=1}^{M} \ln \left\{ \sum_{i=1}^{N} p(v_m|i) P(x_i) \right\}, \tag{1}$$

where $P(x_i)$ is the prior mass at each point in the latent grid and $p(\cdot|i)$ is the Gaussian density centered at $\mathbf{y}_i = \mathbf{y}(x_i, \mathbf{W})$ (equal, of course, to our weight w_i) and spherical covariance with common variance β^{-1}. A generalized linear regression model is typically chosen for the embedding map, namely $\mathbf{y}(x, \mathbf{W}) = \mathbf{W}\Phi(x)$, where $\Phi = \Phi(x)$ is a $N \times B$ matrix containing the scores by B fixed basis functions and $\mathbf{W}$ is a free matrix to be optimized together with β.

2.4 Kernel-based Maximum Entropy learning Rule (kMER)

kMER Van Hulle (1998) was introduced as an unsupervised competitive learning rule for non-parametric density estimation, whose main purpose is to obtain equiprobabilistic topographic maps on regular, fixed-topology lattices. Here, the receptive fields of neurons are (overlapping) radially symmetric kernels, whose radii are adapted to the local input density together with the weight vectors that define the kernel centroids. A neuron w_i is 'activated' by an input data v_m if it is contained within the hypersphere S_i centered at w_i and with radius σ_i. Since the hyperspheres are allowed to overlap, several neurons can be active for a given input data. An online together with a batch version of kMER are developed in Van Hulle (1998). We focus on the online version of kMER, that is,

$$\Delta w_i = \eta \sum_{j=1}^{N} H_{ji} \Xi_j(v) Sgn(v_m - w_i),$$

where $Sgn(\cdot)$ is the sign function taken componentwise, Ξ is a fuzzy code membership function and $H(\cdot)$ the neighborhood function, which decreases over time to achieve a topographically organized lattice. 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 a fixed scale factor ρ, that controls the degree of overlap among receptive fields. Note that, unlike GTM, kMER derives a different standard deviation σ_i for each mixture Gaussian component.

3 Basic tools

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 consider *median interneuron distances*, *dataloads* and *Sammon's projections*. The first two quantities are introduced in detail below, as well as the strategy for mode detection.

To visualize high-dimensional SOM structures, use of Sammon's projection Sammon (1969) is customary. Sammon's map provides a useful global image while estimating all pairwise Euclidean distances among SOM pointers and projecting them directly onto 2D space. Thus, since pointer concentrations in data space will tend to be maintained in the projected image, we can proceed to identify high-density regions directly on the projected SOM. Furthermore, by displaying the set of projections together with the connections between immediate neighbours, the degree of self-organization in the underlying SOM structure can be expressed intuitively in terms of the amount of *overcrossing* connections. This aspect of the analysis is rather important as 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 when working with high-dimensional data Kiviluoto and Oja (1998); Kohonen (2001). On this matter, Kiviluoto and Oja Kiviluoto and Oja (1998) suggested the use of PCA-initialization instead of random-initialization to obtain suitably organized GTM structures ; they also proposed the *S-MAP* method combining GTM and SOM. In addition, 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. On the other hand, if overcrossing in Sammon's projection plot is (closely) null, we can proceed with some confidence.

3.1 Median Interneuron Distances

Since we are interested in regions with higher pointer (or gaussian centre) density, the relative distance from each pointer to its immediate neighbors on the network will provide a useful bit of information. The inspection of pointer interdistances was pioneered by Ultsch Ultsch (1993), who defined the *unified-matrix* (U-matrix) to visualize Euclidean distances between reference vectors in Kohonen's SOM. Emphasis in the U-matrix is on cluster analysis.

Although modes may be associated with clusters Good and Gaskins (1980), a problem exists with high-dimensional data : "while one cluster might be well-represented by a single multivariate Gaussian, another cluster may required dozens of Gaussian components to capture skewness and yet still be unimodal" Scott and Szewczyk (2000). So the relationship between the number of modes and the number of mixture components is not straightforward. In particular, when dealing with multivariate data the mixture may have more modes than mixture components Scott and Szewczyk (2000).

In this paper, we will work with the alternative *median interneuron matrix* (MID-matrix) proposed in Muruzábal and Muñoz (1997). This is a $\sqrt{M}\mathrm{x}\sqrt{M}$ matrix whose (i, j) entry, is the *median* of the Euclidean distances between the gaussian centre and all pointers belonging to a star-shaped, fixed-radius neighborhood containing typically eight units. To facilitate the visualization of higher pointer concentrations, a linear transformation onto a 256-tone gray scale is standard (here the lower the value, the darker the cell).

3.2 Dataloads

The number of data vectors projecting onto each unit, namely the pointer *dataload* $\widehat{\pi}(i, j)$, is the natural estimate of the weight $\pi(i, j)$ obtained from the true underlying distribution f Bodt et al. (1997),

$$\pi(i, j) = \int_{V(i,j)} f(\mathbf{x}) d(\mathbf{x}), \tag{2}$$

where $V(i, j)$ collects all input vectors which are closest to unit (i, j). Again, to easily visualize the dataload distribution over the map, a similar gray image is computed, namely, the DL-matrix. Note that, in this case, darker means higher.

It is important to realize that the "density" of pointers in the trained GTM should serve as an estimate of the density underlying the data. In this ideal case, each neuron would cover about the same proportion of data, that is, a uniform DL-matrix should be obtained. Another interesting way to deal with mode detection and density estimation is to obtain uniformly distributed pointers (over the observed range). Now neurons will present markedly different dataloads, higher densities relating intuitively to larger dataloads. Throughout this paper we focus on the first approach, in which mode detection is based on different pointer concentrations. However, the second idea also seems feasible and is under study.

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.

1. Train the model until the Sammom's projected pointers show a good level of organization and a (nearly) uniform DL-matrix is obtained. Check also the stability of log-likelihood values.

2. Compute MID matrix. If the existence of more than one mode is suggested, build subsets of data and return to STEP1 to fit individual topographic maps at each unimodal subset.
3. Combining MID-matrix's darkest region and pointer concentration on Sammon's projection an approximation to the single mode's location can be performed.

If a point estimate is required, one can simply hill-climb the MID surface. For example, Figure 1 reflects the success of this hill-climbing strategy (used below predominantly on the density estimate) : all three modes are pinpointed without even splitting the data set. This completes our theoretical presentation. We are now ready to examine some empirical evidence.

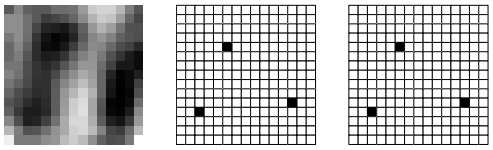

FIG. 1 – GTM on 3M-2D data set, see Figure 2. (a) MID-matrix ; (b) modes detected by hill-climbing the MID surface ; (c) neurons actually lying closest to the real modes.

4 Simulation study

Here we summarize our main experimental results. We first analyze a trimodal 2D data set which we call 3M-2D. This data set is simulated from a mixture of three Gaussians, with two of the three modes close enough to illustrate the accuracy of each algorithm. We then continue with a data set generated from two well-separated 10D Gaussians (2M-10D). Finally, we examine a more difficult highly multimodal real-world problem, the Multiple Features Database developed by Jain et al. (2000) and extracted from Blake and Merz (1998). This data set, refer to as Mfeat, consists of features of handwritten numerals $(0, \ldots, 9)$, with 200 patterns per class for a total of 2,000 patterns, 649 attributes and 10 classes. In this case, $d = 649$, $N = 2000$ and we look for 10 higher density regions. Our real data set is used to verify that the basic procedure still works fine even when the mixture is not balanced and its components are far from Gaussian. Here we focus on GTM and kMER only. We analyze performance by all training methods and test all detection ideas presented in the paper. The maps discussed here have been all stopped early and their DL-matrices exhibit a reasonable amount of uniformity.

All algorithms are run in an Intel Pentium IV machine. `MATLAB` code for both SOM_batch and GTM is available at `http ://www.ncrg.aston.ac.uk/GTM`. SOM_Cx and kMER algorithms have been developed following the descriptions and guidelines in Zheng and Greenleaf (1996) and Van Hulle (2000) respectively. Training parameters for each algorithm and each data set are the following. For the four algorithms we train a 15×15 sized map for the 3M-2D data set and a 10×10 sized map for 2M-10D and Mfeat data sets.

For SOM_batch algorithm 1,000 training cycles are considered in both sets, as well as a Gaussian neighborhood and a sheet shaped rectangular lattice. For 3M-2D data set the

started neighborhood radius is set to 6 and to 0.5 the final one. For 2M-10D data set the neighborhood radii move from 1.5 to 0.001.

SOM_Cx algorithm trains both data sets during 6,000 cycles of training, and considers also a Gaussian neighborhood and a sheet shaped rectangular lattice. The neighborhood radii move from 4 to 0.2 in 3M-2D data set and from 5 to 0.01 in 2M-10D data set. The initial learning parameter are set to 0.6 and 0.5 in 3M-2D and 2M-10D data sets, respectively. The convex factor is 9 for 3M-2D and 31 for 2M-10D data sets.

GTM learning rule trains the three data sets over 100 training cycles for 3M-2D and 2M-10D, and 2,000 for Mfeat. The initial pointers configuration follows a PCA study. The size of the basis function grid is set to 6×6 for 3M-2D data set, 8×8 for 2M-10D and 7×7 for Mfeat data set. The common width of the basis functions is 1.2, 0.9 and 1 for 3M-2D, 2M-10D and Mfeat data sets, respectively, and the regularization term value is 2, 10 and 0 for 3M-2D, 2M-10D and Mfeat.

kMER algorithm employs 100,000 cycles of training on 3M-2D and 2M-10D data sets, and 2,000 for Mfeat. Both pointers and mixture component standard deviation are randomly initialized. The initial neighborhood range is set to 10, 4 and 5 for 3M-2D, 2M-10D and Mfeat. The scale factor is fixed to 1.9, 2.1 and 2, and the learning rate values are 0.01, 0.01 and 0.00015, respectively.

4.1 Three modes in 2D space

Figure 2 shows how well the algorithms fit this 2D trimodal data set. In the case of SOM_batch, note that dead neurons appear (as it is typically the case with this algorithm), and a pronounced overload can be seen at two corners of DL-matrix. Still, the basic approach to mode detection can be sensibly developed to some extent. We appreciate how the MID-matrix is able to identify two of the three modes, missing the distinction between the two closer modes.

SOM_Cx clearly yields a better usage of the map resources (the number of dead neurons has decreased substantially). Indeed, this DL-matrix shows more uniformity than before, leading in particular to a MID-matrix capable of distinguishing the three existing modes (compare Figures 2c and 2f).

Nice results are obtained using GTM on this data set. The trained map shows a good level of organization and pointer concentration, and it also seems to span the support of the sampling distribution better than the previous algorithms. As DL-matrix reflects an appropriate level of uniformity, the basic approach provides excellent results identifying the three modes.

Turning finally to the kMER algorithm, we see that a nearly uniform DL-matrix is obtained. A useful MID-matrix is observed as well. In particular, mode estimation is very precise again.

4.2 Two modes in 10D space

We inquire now about mode detection in a data set generated from the mixture of two 10D Gaussians (2M-10D). These Gaussians are centered at the points $(0, ..., 0)$ and $(0.35, ..., 0.35)$, with a common spherical covariance matrix $0.2 \cdot \mathbf{I}$. We use this data set to test the four algorithms in a simple high-dimensional problem.

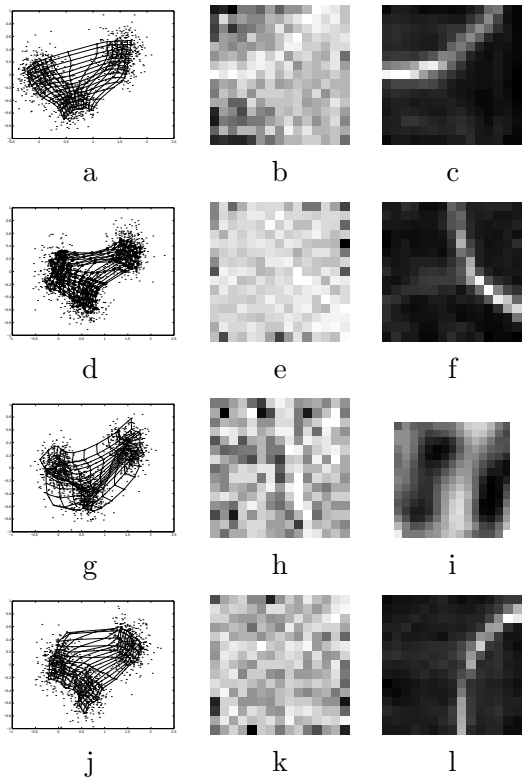

FIG. 2 – Performance by SOM_batch (top), SOM_Cx (middle-up), GTM (middle-down) and kMER (bottom) on 3M-2D data set : (a,d,g,k) trained map with data set highlighted ; (b,e,h,k) DL-matrix ; (c,f,i,l) MID-matrix.

Figure 3 shows the maps obtained by the algorithms in this case. In the case of SOM_batch we appreciate how the MID-matrix is able to distinguish the two clusters neatly.

The results by the convex adjustment algorithm are presented now. Likewise, both MID-matrix and Sammon's projected map make a distinction between the two modes.

As it concerns GTM good results are obtained. The basic approach reflects a well-organized map and a MID-matrix that categorically splits the grid of pointers.

The analysis of the map obtained using kMER reflects that an accurate mode estimation is obtained again, a well-organized map and a rather uniform DL-matrix are achieved. The MID-matrix is able to identify two higher density areas.

4.3 Real-world example

We finally examine how our tentative criterion does with a highly dimensional real-world example, the Mfeat data set. In this case, $d = 649$, $N = 2000$ and we look for 10 higher density regions. A labelled matrix, say LAB_matrix, where each neuron is marked with the label that occurs most within its activation region, is exhibit in this case. To verify the precision of LAB_matrix, another matrix revealing the ratio of times that each label appears in its activation region is introduced. In this confidence matrix, called CONF-matrix, the higher the values, the higher the confidence.

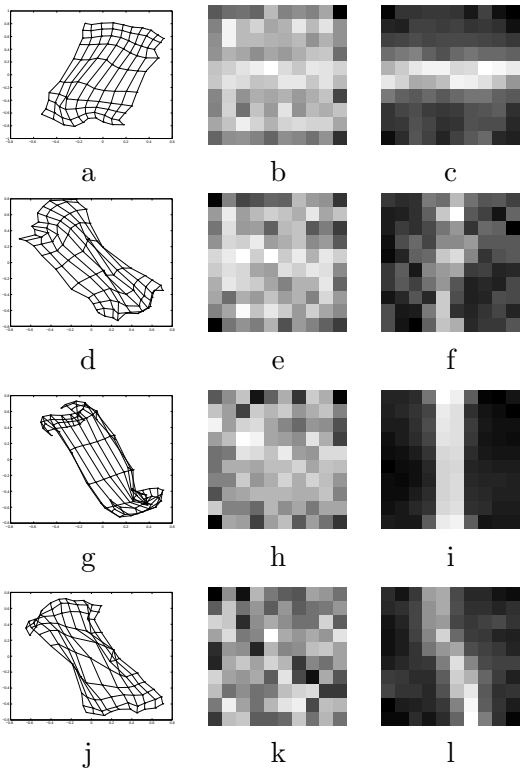

FIG. 3 – Performance by SOM_batch (top), SOM_Cx (middle-up), GTM (middle-down) and kMER (bottom) on 2M-10D data set : (a,d,g,k) Sammon's projected map ; (b,e,h,k) DL-matrix ; (c,f,i,l) MID-matrix.

In Figure 4 the optimal maps obtained after applying the monitoring policy to GTM and kMER are showed. In the Sammon's projections we can appreciate the nearly-perfect organized maps— note that we are in a 649th dimensional space !. LAB_matrices split the region in 10 accurate and precise areas. Although both algorithms present well-defined labelled matrices, where only the edges of the 10 areas could lead to misclassification problems, kMER optimal map seems to be the best, leading to the clearest matrix of labels. Moreover, CONF_matrix of kMER present the highest confidence values, while in the one of GTM some neurons with zero value appeared. kMER MID_matrix provide an accurate Hill-Climbing procedure, that "places a mode" in each of the clusters marked by the corresponding LAB_matrices. On the other hand, the drawbacks of GTM algorithm due to it stiffness are patent in this example— HC-matrix is able to find 6 out of 10 modes.

Summing it up, even in a highly dimensional real-world example an accurate estimation of the true underlying modes can be achieved following the strategy for mode detection developed here.

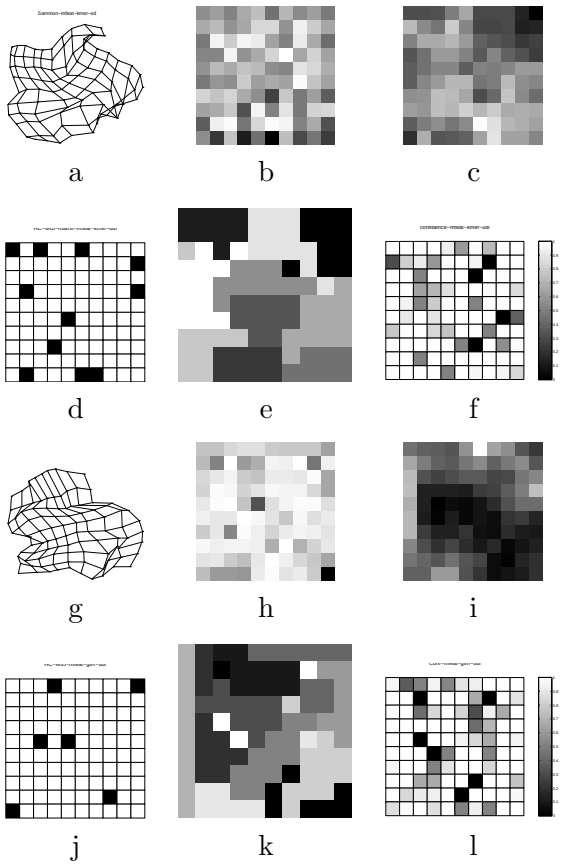

FIG. 4 – Performance by kMER (top) and GTM (bottom) on Mfeat data set : (a,g) Sammon's projected map ; (b,h) DL-matrix ; (c,i) MID-matrix ; (d,j) HC-matrix on MID-matrix ; (e,k) LAB-matrix ; (e,j) CONF-matrix ;

5 Summary and conclusions

We have described a number of approaches to mode detection, and tested mode estimation based on topographic map structure. Four training algorithms have been examined over two synthetic data sets and a highly dimensional real world example.

The convex adjustment of the standard Kohonen rule shows that we can expect a good deal from the basic diagnostics even in the absence of any statistical modelling. Gaussian kernels in GTM are constrained by the transformation from lattice to input space coordinates, so these kernels cannot move freely when needed at some point along the training process. This appears to be the main disadvantage of GTM in its present form, its relative stiffness (see also Vegas-Azcárate and Muruzábal (2003)). We have seen that many of the previous drawbacks are avoided by kMER, which produces more flexible maps. On the other hand, a number of ideas can be tried in GTM in an effort to alleviate the previous problem.

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 strategy for mode detection is easy to apply and provides sensible answers in our test problems. We have also seen that maps obtained with this strategy, consti-

tute examples of well-organized structures which deserve further attention on their own. We have prove here that, despite of the different shades derived from the diverse nature of each algorithm, the fourth of them are able to identify modes' location if the presented strategy is followed.

Références

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. (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`.

Bodt, E., Verleysen, M. and Cottrell, M. (1997). Kohonen maps versus vector quantization for data analysis, in *European Symposium on Artificial Neural Networks, ESANN'97*, pp. 211–220, Brussels : D-Facto publications.

Cheng, B. and Titterington, D. (1994). Neural networks : A review from a statistical perspective, *Statistical Science*, **9(1)** : 2–54.

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.

Good, I. J. and Gaskins, R. A. (1980). Density estimation and bump-hunting by penalized likelihood method exemplified by scattering and meteorite data, *Journal of the American Statistical Association*, **75** : 42–73.

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.

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

Izenman, A. J. and Sommer, C. J. (1988). Philatelic mixtures and multimodal densities, *Journal of the American Statistical Association*, **83** : 941–953.

Jain, A., Duin, R. and Mao, J. (2000). Statisitcal pattern recognition : A review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22(1)** : 4–37.

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. (1993). Physiological interpretation of the self-organizing map algorithm, *Neural Networks*, **6** : 895–905.

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

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.

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

Scott, D. W. (1992). *Multivariate Density Estimation*, New York : John Wiley and Sons, Inc.

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

Terrell, G. R. and Scott, D. W. (1992). Variable kernel density estimation, *Ann. Statist.*, **20** : 1236–1265.

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.

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. (2003). On the use of the GTM algorithm for mode detection, in *LNCS*.

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.