Journal of Classification 13:299-313 (1996)

Constrained Clustering and Kohonen Self-Organizing Maps

Christophe Ambroise
URA CNRS 817

Gérard Govaert
URA CNRS 817

Abstract: The Self-Organizing Feature Maps (SOFM; Kohonen 1984) algorithm is a well-known example of unsupervised learning in connectionism and is a clustering method closely related to the k-means. Generally the data set is available before running the algorithm and the clustering problem can be approached by an inertia criterion optimization. In this paper we consider the probabilistic approach to this problem. We propose a new algorithm based on the Expectation Maximization principle (EM; Dempster, Laird, and Rubin 1977). The new method can be viewed as a Kohonen type of EM and gives a better insight into the SOFM according to constrained clustering. We perform numerical experiments and compare our results with the standard Kohonen approach.

Keywords: EM algorithm; Gaussian mixture; Kohonen maps; Constrained clustering.

1. Introduction

In the field of neural networks there are two main directions of research. The biologist tries to understand real biological computations, and the engineer borrows some biological concepts to design new pattern recognition algorithms. Kohonen Self-Organizing Feature Maps (Kohonen 1982) are

---

The authors thank the referees and the editor for several comments which helped to improve the presentation.

Authors' Addresses: Christophe Ambroise and Gérard Govaert, URA CNRS 817, Université de technologie de Compiègne, BP 649, 60206 Compiègne Cedex, France. E-mail: gerard.govaert@utc.fr

closely modeled after neurobiological structures. This robust method can be used either for cluster analysis or for dimensionality reduction.

This paper extends some ideas of Kohonen to the EM algorithm family (Dempster, Laird, and Rubin 1977; Celeux and Diebolt 1985; Celeux and Govaert 1992). We first introduce the SOFM algorithm in a clustering context and then show how it is related to the classical k-means (MacQueen 1967) and the probabilistic approach of clustering (Wolfe 1970). In Section 3 we present a new algorithm, which combines the Kohonen idea of neighborhood interaction function and the structure of the Classification EM algorithm (Celeux and Govaert 1992). The algorithm integrates the topology preservation property of the SOFM and can be viewed as a Topology Preserving EM algorithm (TPEM). Its convergence is demonstrated and a stochastic version of TPEM (STPEM) is proposed. Section 4 is devoted to numerical experiments to compare the practical behavior of SOFM, CEM, and TPEM.

2. Kohonen Self-Organizing Maps and CEM Algorithms

2.1 Self-Organizing Feature Maps Algorithm

The Kohonen model takes its inspiration from the adaptative formation of topology-conserving neural projection in the brain. Its aim is to generate a mapping of a set of high-dimensional input signals onto a one- or two-dimensional array of formal neurons. Each neuron becomes representative of some input signals such that the topological relationship between input signals in the input space is reflected as faithfully as possible in the arrangement of the corresponding neurons in the array (also called output space). When using this method for clustering, it is possible either to match each neuron with a unique cluster or to match many neurons to one cluster. In the latter case the Kohonen algorithm produces a reduced representation of the original data set, and clustering algorithms may operate on this new representation (Murtagh 1995). When using the term "clustering" in this paper, we refer only to the case where a unique neuron corresponds to one cluster.

In a clustering framework, constraining the topology may have some advantages: In image segmentation for example, it is possible to use the topological relationship to represent two close classes by two close colors, and this kind of segmentation may facilitate the interpretation of the segmented image (Tomasini 1993). Another field where topology preservation has proved to be of practical interest is vector quantization: Luttrel (1990) showed that topographic mappings arise naturally in a 2-stage vector quantizer whose output is made robust with respect to various types of distortion.

The Kohonen Self-Organizing Feature Maps method is often implemented as an adaptive algorithm. This means that at each iteration a unique

input sample which is a d-dimensional vector, $\mathbf{x}_i$, from the data set, is taken into account to update the weight vector of some neurons. Let us describe this algorithm below:

1. Some parameters have to be initialized: An array of neurons is defined. Each neuron k has a location $\mathbf{r}_k$ in the output space and is represented by its weight vector μ_k in the input space. The weight vectors are randomly initialized. An initial learning step $\alpha(0)$ is chosen. A neighborhood interaction function $h(k,l)$ is chosen. It is a function of the array distance $d_{kl} = \|\mathbf{r}_k - \mathbf{r}_l\|$. $h(k,l) = 1$ when $k = l$ and decreases to zero as $\|\mathbf{r}_k - \mathbf{r}_l\|$ increases.

   A box function may be used as neighborhood interaction function:

$$h(k,l) = \begin{cases} 1 & \text{if } d_{kl} < \sigma_m; \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

   where σ_m is the width of the neighborhood taken into account at iteration step m.

2. For each iteration, the following steps are executed unless a stopping condition is reached:
   (a) A d-dimensional input pattern, $\mathbf{x}_i$, is randomly chosen and presented to the interconnected network of neurons.
   (b) The best matching unit (BMU) is located:[1]

$$k^* = \text{argmin}_k \|\mathbf{x}_i - \mu_k\| . \tag{2}$$

   (c) The matching of the unit and its neighbors is increased:

$$\mu_k(m+1) = \mu_k(m) + \alpha(m) \cdot h(k,k^*) \cdot [\mathbf{x}_i - \mu_k(m)] \; \forall \; k ,$$

   where k^* is the index of the best matching unit, $h(k,k^*)$, the neighborhood interaction function and $\alpha(m)$, a decreasing learning step, satisfying the stochastic approximation conditions (see Benveniste, Metivier, and Priouret 1987, ch. 1, p. 32):

$$\sum_{m=1}^{\infty} \alpha(m) = \infty \text{ and } \sum_{m=1}^{\infty} \alpha(m)^2 \leq \infty .$$

1. The argmin function returns the value of the index vector which minimizes the argument.

Kohonen (1991) relates his algorithm to the theory of stochastic approximation (Robbins and Monroe 1951). Note that $h(k,k^*)$ is defined and unique in the input space except when an input pattern $\mathbf{x}_i$ has exactly the same distance from two or more μ_k. But considering the subset of the input space where the neighborhood interaction function is uniquely defined, the update rule (3) may be derived from the following potential:

$$E = \sum_{\mathbf{x}_i \in S} \sum_{\mu_k \in A} h(k,k^*) \cdot \|\mathbf{x}_i - \mu_k\|^2 , \tag{4}$$

where S is a finite samples set, A the array of formal neurons, and $k^* = \text{argmin}_k \|\mathbf{x}_i - \mu_k\|$.

From a standard optimization point of view, the algorithm minimizes an inertia criterion. It is possible to formulate this optimization problem with different clustering approaches: In traditional partitioning, a finite set of samples $S = \{\mathbf{x}_1,..,\mathbf{x}_n\}$ is partitioned into K classes. The cluster centers (prototypes, weights) are denoted $\mu_1, \ldots, \mu_k$. The quality of the partition is measured by a criterion. A well-known example of partitioning method is the k-means algorithm (MacQueen 1967), which minimizes the within-class sum of squared errors criterion:

$$W = \sum_{\mathbf{x}_i \in S} \sum_{\mu_k \in A} u_{ik} \cdot \|\mathbf{x}_i - \mu_k\|^2 , \tag{5}$$

where u_{ik} are the membership coefficients ($u_{ik} = 1$ when $\mathbf{x}_i$ is member of the k-th class; $u_{ik} = 0$ otherwise).

If we consider a zero-neighbor interaction function, $h(k,l) = \delta(k,l)$ (where $\delta(k,l)$ is the Kronecker delta function), the potential E of Equation 4 is reduced to within-class sum of squared errors criterion (Equation 5), and we can write:

$$h(k,k^*) = u_{ik^*} . \tag{6}$$

In this context, $h(k,\text{argmin}_l \|\mathbf{x}_i - \mu_l\|)$ is interpreted as the membership of $\mathbf{x}_i$ in the k-th cluster.

2.2 The Classification EM Algorithm

In cluster analysis based on Gaussian mixture models, data are $\boldsymbol{R}^d$-valued vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ assumed to be a sample from a mixture of densities:

$$f(\mathbf{x}) = \sum_{k=1}^{K} p_k \, \Phi(\mathbf{x} \mid \mu_k, \Sigma_k) , \tag{7}$$

where the p_k are the mixing proportions ($0 < p_k < 1$, for all $k = 1,..,K$ and $\sum_k p_k = 1$), and $\phi(\mathbf{x} \mid \mu_k, \Sigma_k)$ denotes the density of a Gaussian distribution

with mean vector μ_k and variance matrix Σ_k. A possible classification criterion may be:

$$\mathrm{CML}(\mathbf{U},\theta) = \sum_{k=1}^{K}\sum_{i=1}^{n} u_{ik} \log\left(p_k \Phi(\mathbf{x}_i \mid \mu_k, \Sigma_k)\right). \tag{8}$$

Celeux and Govaert (1992) show that maximizing the CML criterion for a Gaussian mixture with equal mixing weights and a common covariance matrix of the form $\sigma^2 I$ (σ^2 unknown) is equivalent to minimizing the within-class sum of squared errors criterion.

The CML criterion can be optimized by making use of an EM algorithm classification version, the so-called CEM algorithm (Celeux and Govaert 1992), which is briefly described below:

1. Data are randomly partitioned into K classes. An initial classification matrix $\mathbf{U}^0 = [u_{ik}^0]$ is computed.
2. For each iteration the following steps are executed unless a stopping condition is reached:
   (a) *E-step:* The current conditional probabilities $t_k^m(\mathbf{x}_i)$ that each $\mathbf{x}_i$ belongs to the k-th cluster are computed:

$$t_k^m(\mathbf{x}_i) = \frac{p_k^m \Phi(\mathbf{x}_i \mid \mu_k^m, \Sigma_k^m)}{\sum_{l=1}^{K} p_l^m \Phi(\mathbf{x}_i \mid \mu_l^m, \Sigma_l^m)}. \tag{9}$$

   (b) *C-step:* The updated partition is calculated by assigning each $\mathbf{x}_i$ to the cluster which provides the current maximum conditional probability $t_k^m(\mathbf{x}_i)$. The classification matrix $\mathbf{U}^m = [u_{ik}^m]$ corresponding to this new partition is computed.
   (c) *M-step:* Find the mixture parameters which maximize CML:

$$\mu_k^{m+1} = \frac{\sum_{i=1}^{n} u_{ik}^m \mathbf{x}_i}{\sum_{i=1}^{n} u_{ik}^m}; \tag{10}$$

$$\Sigma_k^{m+1} = \sum_{k=1}^{K}\sum_{i=1}^{n} u_{ik}^m (\mathbf{x}_i - \mu_k^{m+1})(\mathbf{x}_i - \mu_k^{m+1})'; \tag{11}$$

$$p_k^{m+1} = \frac{\sum_{i=1}^{n} u_{ik}^m}{n}. \tag{12}$$

We have shown that E and CML (Equations 4 and 8) criteria are both related to the within-class sum of squared errors criterion. The topological constraints in the E criterion are expressed using the coefficients $h(k,l)$ (Equation 1), which may be interpreted as membership coefficients, as is the case for the u_{ik} of the CML criterion. In the next section, we show how to introduce topological constraints in CEM using these membership coefficients.

3. Topological Constraints in CEM

3.1 The Topology Preserving EM Algorithm

The Topology Preserving EM (TPEM) algorithm proposed here is a generalization of Kohonen's "Batch Map" algorithm (Kohonen 1993). This new method is based on the CEM algorithm to optimize the CML criterion and finds a partition of the data into K classes and a relationship among these clusters. Like the SOFM algorithm, it can be used for clustering and dimensionality reduction. All the training samples are taken into account at each iteration, so this algorithm can be regarded as a batch version of the Kohonen algorithm. It integrates the idea of the neighborhood interaction function of the SOFM with the CEM algorithm. This function is used to compute the classification matrix $\mathbf{U}$ during the C-step of CEM:

1. Data are randomly partitioned into K classes. An initial classification matrix $\mathbf{U}^0 = [u_{ik}^0]$ is computed. An initial neighborhood width σ_0 and the location of the classes in the output space are defined.
2. For each iteration the following steps are executed unless a stopping condition is reached:
   (a) *E-step:* The current conditional probabilities $t_k^m(\mathbf{x}_i)$ that each $\mathbf{x}_i$ belongs to the k-th cluster are computed according to Equation 9.
   (b) *C-step:* Compute the classification matrix elements:

$$u_{ik}^m = \frac{h^m(k,kmax)}{\sum_{l=1}^{K} h^m(l,kmax)}, \tag{13}$$

   where $kmax = \text{argmax}_k(t_k(\mathbf{x}_i))$ and $h^m(k,l)$ is the neighboring function defined in Equation 1.

   This definition means that the matrix $\mathbf{U}^m$ defines a fuzzy partition. Each $\mathbf{x}_i$ belongs to many clusters: the cluster which provides the maximum posterior probability and its neighboring clusters (in the sense of the neighboring function h).
   (c) *M-step:* The maximum likelihood estimates $(\hat{p}_k, \hat{\mu}_k, \hat{\Sigma}_k)$ are computed using the k-th cluster as a subsample $(1 \leq k \leq K)$.

The preservation of the topology is induced by the neighborhood function h, which takes into account proximity relationship in the output space. Kohonen (1984) suggests starting with a wide neighborhood and then letting this neighborhood shrink with the number of iterations. We have chosen a linear decay rule defined by $\sigma_{m+1} = a\,\sigma_m$ (typically: $0.9 \leq a < 1$).

It is possible to distinguish two phases in the evolution of the process:

1. Qualitatively speaking, a self-organization phase first takes place, during which, the neighborhood function has an influence on the process. Quantitatively speaking, the width of the neighborhood is greater than one ($\sigma_m > 1$).
2. There is a clustering phase. Quantitatively speaking, when the neighborhood width does not exceed one ($\sigma_m \leq 1$), the algorithm is exactly the same as CEM. If we consider the above defined linear function for shrinking the neighborhood, the topology preserving EM algorithm becomes CEM when the number of iterations is greater than $\left\lfloor \frac{\log \sigma_0}{\log a} + 1 \right\rfloor$.

As TPEM becomes CEM after a finite number of iterations and that CEM generates a converging ($\mathbf{U}_m$, p^m, μ^m, Σ^m) sequence (Celeux and Govaert 1992), the sequence produced by the TPEM algorithm converges to a stationary value.

When the process is in the organizing phase, a fuzzy classification matrix $\mathbf{U}^m$ is produced at each iteration step ($\sum_k u_{ik} = 1$ and $u_{ik} \in [0,1]$). As soon as the CEM phase begins, the classification becomes hard ($\sum_k u_{ik} = 1$ and $u_{ik} \in \{0,1\}$).

From a practical point of view, the solution provided by the TPEM algorithm does depend upon the decay rate of the neighborhood width, the initial width, and the initial partition. It may happen that the algorithm stops in the self-organization phase and provides then a "bad" CML value (when we compare with the results obtained by CEM). To overcome this problem, the TPEM algorithm is rerun several times with different setting of the initial parameters, and the solution which provides the best CML criterion is selected. In the following we propose a stochastic version of TPEM, STPEM, which directly addresses the parameter initial-setting dependence.

3.2 The Stochastic Topology Preserving EM Algorithm

The algorithm uses a Random Imputation Principle (RIP, see Celeux and Diebolt 1985) during the Classification step. It has exactly the structure of TPEM except for the C-step:

1. All the parameters are initialized as for TPEM.
2. For the m-th iteration, the following steps are executed unless a stopping condition is reached:
   (a) *E-step.*
   (b) *C-step:* Compute score matrix elements, s_{ik}^m, according to

Equation 13. Notice that this score matrix, $\mathbf{S}^m = [s_{ik}^m]$ of STPEM, is equivalent to the classification matrix $\mathbf{U}^m = [u_{ik}^m]$ of TPEM. Each $\mathbf{x}_i$ is assigned to a class k according to a multinomial distribution with probabilities $s_{i1}, \ldots, s_{iK}$.

(c) *M-step.*

STPEM shares some similarities with simulated annealing (see Klein and Dubes 1989 for more details about simulated annealing in clustering): It is possible to draw a parallel between the width of the neighborhood σ_m in STPEM and the temperature of the system in simulated annealing. The smaller the temperature is (respectively the neighborhood width), the less random decisions influence the system.

When the width of the neighborhood is smaller than one, STPEM becomes CEM. Notice that STPEM, like TPEM, comprises two phases. The first Self-Organizing phase, where CML does not always increase, and a random factor determined by the neighborhood width is important for the computation of the classification matrix. The second phase is identical to the second phase of TPEM: it is a succession of CEM iterations.

Concerning the convergence of STPEM, the same observations apply as for TPEM. As STPEM becomes CEM after a certain number of iterations, the algorithm converges.

In the next section, we report numerical experiments to assess the ability of TPEM and STPEM to produce local maxima of the CML criterion and good reduced representations.

4. Numerical Experiments

We have performed numerical experiments to compare the SOFM, CEM, TPEM, and STPEM algorithms. These numerical experiments attempt to illustrate two different capacities of TPEM and STPEM. Firstly, we want to assess the ability of TPEM and STPEM to partition multidimensional data. Second, we are interested in the topology preservation property of the algorithms and their ability to produce reduced representation.

To compare the performance numerically, we use two different criteria. For measuring the clustering ability, we use W, the within class sum of squared errors criterion (see Equation 5). As mentioned in Section 2, this measure is equivalent to the Classification Maximum Likelihood when assuming that samples are from spherical Gaussian distributions mixed in equal proportion. For measuring the topological preservation we use a criterion that we call the Sum of the Edge Length criterion. Topological conservation between input space and output space means that neighboring nodes in the array must be neighbors in the input space: thus a mathematical measure

of this property may be the sum of the edges of the array:

$$SEL = \sum_{k=1}^{K} \sum_{l=1}^{K} h(k,l) \cdot \|\mu_k - \mu_l\|, \tag{14}$$

where $h(k,l) = 1$ if classes k and l are neighbors in the output space and $h(k,l) = 0$ otherwise. This criterion was used by Durbin and Mitchinson (1990) and Benaim and Tomasini (1991) to design topology preserving algorithms. We have used both real and simulated data with or without cluster structure. The real data set consists of Anderson's iris data (Anderson 1935). It is composed of three classes of 50 four-dimensional samples each and will be called IRIS. The second data set, GAUSS, is composed of three spherical Gaussian clusters of 50 two-dimensional samples each. The means of the three clusters are respectively (0,0) (3,0) and (2,2) and all have I as covariance matrix. We simulated a two-dimensional uniform distribution of 400 samples in a square and in a triangle. These data sets will be denoted respectively UNI_SQR and UNI_TRI. Notice that UNI_SQR and UNI_TRI have no well-separated clusters, contrary to the IRIS and GAUSS data sets.

From a probabilistic point of view, the k-means algorithm assumes that all sought classes are Gaussian with spherical shape, that they are mixed in the same proportion, and have the same volume. The SOFM algorithm is closely related to the k-means algorithm and has the same assumptions. To compare STPEM and TPEM with SOFM and the CEM algorithms, the implementation of the EM family algorithms makes the same hypothesis about the structure of the clusters. Notice that in this particular case, CEM is equivalent to the k-means and TPEM to the Batch Map of Kohonen. Recall that TPEM and STPEM have two different kind of iterations: the first constitute an initialization phase and are followed by some iterations of the CEM algorithm, which is the k-means algorithm for these numerical experiments.

Concerning the neighborhood shrinking function of the form $\theta_{m+1} = a \cdot \theta_m$ (where θ_m is the width of the neighborhood taken into account at step m), we chose $a = 0.98$ and θ_0 as being half the number of classes. From our experience these values provide good solutions within a reasonable number of iterations.

Topology preserving algorithms may be used for cluster analysis or dimensionality reduction. We tested the clustering ability with the IRIS and GAUSS data sets (see Figure 1), seeking a partition of the data into three classes.

When performing dimensionality reduction, we want the output space representation to represent as well as possible the topology of the input samples. It is possible to preserve the information about the topology by using many neurons (or classes in the cluster analysis terminology). Thus we use a

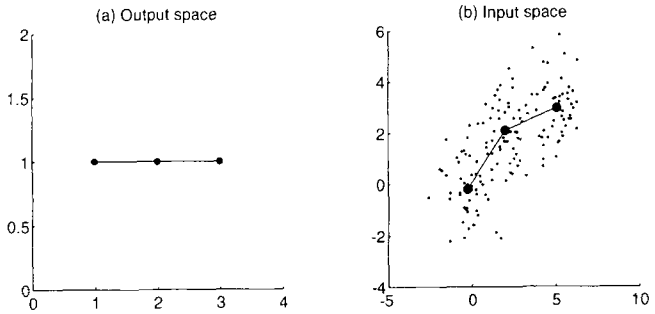

Figure 1. A 3 neuron linear metwork and the GAUSS data set.

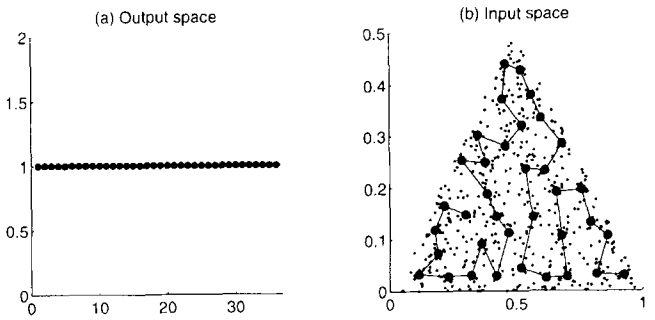

Figure 2. A 36 neuron linear network and the UNI_TRI data set.

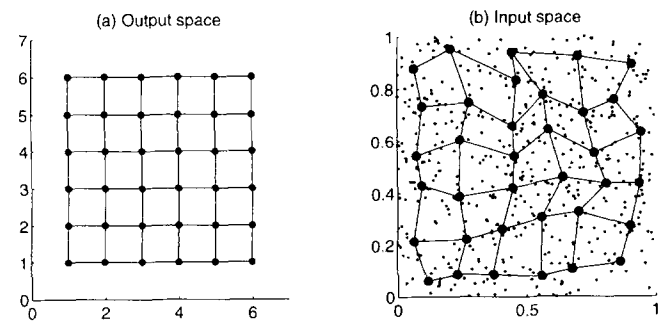

Figure 3. A 36 neuron planar network and the UNI_CAR data set.

Table 1: Summary statistics of the numerical results

		W criterion				SEL criterion			
		best value	worst value	mean	Std	best value	worst value	mean	Std
IRIS	SOFM	79.14	84.54	81.69	1.33	24.88	31.45	29.01	1.53
	k-means	78.85	142.75	87.38	22.09	28.84	72.89	53.57	17.90
	TPEM	78.86	504.45	288.33	213.16	11.03	28.84	19.93	9.05
	STPEM	78.86	78.86	78.86	0	28.84	28.84	28.84	0
GAUSS	SOFM	280.02	300.02	287.84	5.37	34.06	51.67	41.43	4.16
	k-means	274.08	278.73	274.28	0.85	39.41	97.79	81.34	23.46
	TPEM	274.08	926.55	402.11	260.47	16.83	39.87	35.26	9.37
	STPEM	225.98	225.98	225.98	0	41.83	41.83	41.83	0
UNI_TRI	SOFM	0.46	0.49	0.47	0.01	0.51	0.66	0.58	0.04
	k-means	0.63	0.85	0.73	0.05	3.50	6.31	4.85	0.65
	TPEM	0.47	0.56	0.50	0.02	0.49	0.59	0.54	0.03
	STPEM	0.47	0.52	0.50	0.01	0.47	0.63	0.54	0.03
UNI_CAR	SOFM	1.89	1.96	1.92	0.02	3.57	4.23	3.78	0.15
	k-means	2.04	2.74	2.41	0.16	24.32	37.70	29.17	3.05
	TPEM	1.93	2.09	1.99	0.04	3.38	5.48	3.72	0.51
	STPEM	1.94	2.09	2.00	0.03	3.38	4.21	3.71	0.19

linear network of 36 neurons to cope with the UNI_TRI data set and a grid of 6 by 6 neurons with the UNI_CAR data set (see Figures 2 and 3).

Because the solutions obtained with this kind of algorithm depend on the initialization, we ran each algorithm 30 times with different initializations. The CML and SEL criteria obtained by each algorithm with a data set are summarized by a 2-dimensional histogram. Figures 4 through 7 and Table 1 report summary statistics concerning these numerical experiments.

5. Discussion

If the TPEM algorithm stops in the initialization phase, it then gives a small value of the SEL criterion and a big value of the W criterion. An

example of this behavior is visible with the IRIS and GAUSS data sets (Figures 4c and 5c). With these data sets the TPEM algorithm produces two types of results: solutions with very small (good) values of the W criterion and contrasting solutions with big values of the W criterion as the result of aborted runs. Such solutions are uninteresting for clustering and dimensionality reduction because some classes are empty.

When the TPEM algorithm does not stop, the results obtained are statistically very close to STPEM algorithm results. This finding is particularly clear if one looks at Figures 6 and 7 and probably results from the two topology preserving algorithms, which both terminate with k-means iterations.

With all the data sets, the STPEM algorithm always produces better values than the k-means algorithm of the W criterion with a smaller standard deviation. For example, with the IRIS and GAUSS data sets where the cluster structure is strong, STPEM gives the same solution for all 30 runs.

An intuitive explanation is that the initialization phase, which is where the prototypes are organized to preserve the topology, guides the algorithms near some local optima of the W criterion. Experimentally we observe that the topology preservation constraints reduce the number of possible optima.

Kohonen algorithm produces different values of the two criteria for each run, but small standard deviations. The values of the criteria are identical because it is an adaptive algorithm and it converges after an infinite number of iterations (if it converges. That is not proved yet). From a practical point of view this means that the algorithm is very robust and seems to converge toward a small set of optima.

The topology preserving algorithms we have considered are robust algorithms which are less sensitive to the initialization dependence problem than k-means because of the self-organizing phase. The behavior of the Kohonen, TPEM, and STPEM algorithms has been shown to be comparable. TPEM and in particular the STPEM algorithms may be considered as batch versions of the Kohonen algorithm.

6. Conclusion

We have proposed two algorithms which can be considered as batch versions of the SOFM algorithm. They use the ideas of Kohonen about topological relationship between classes to provide some particular initialization to the CEM algorithm and tend to produce fewer local optima than CEM of the CML criterion. Further research should determine in which context each algorithm performs best. It would be especially interesting to explore the possibilities offered by the probabilistic framework (Celeux and Govaert 1995): implementation of TPEM and STPEM for finding classes with different shapes, volumes, and proportions.

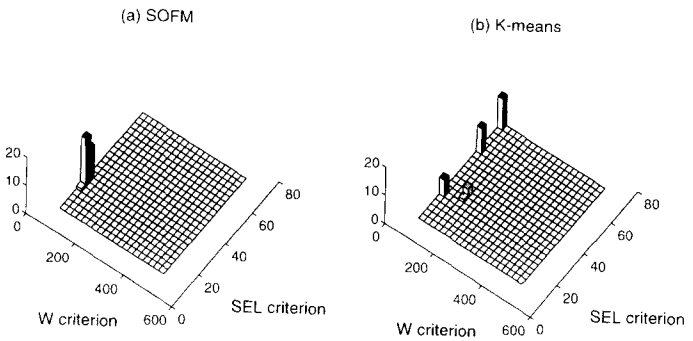

Figure 4. Results obtained with the IRIS data set.

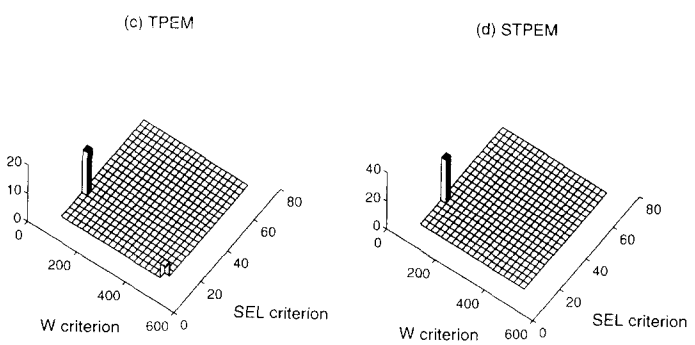

Figure 5. Results obtained with the GAUSS data set.

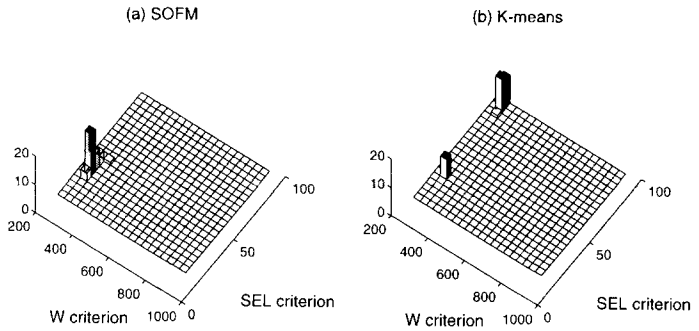

Figure 6. Results obtained with the UNI_TRI data set.

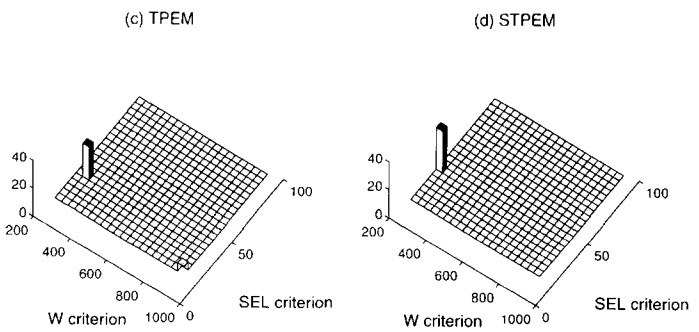

Figure 7. Results obtained with the UNI_CAR data set.

References

ANDERSON, E. (1935), "The irises of the Gaspé Peninsula," *Bulletin of the American Iris Society, 59*, 2-5.

BENAIM, M., and TOMASINI, L. (1991), "Competitive and Self-organizing Algorithms Based on the Minimization of an Information Criterion," in *Artificial Neural Networks*, Eds., T. Kohonen, K. Mäkisara, O. Simula and J. Kangas, Amsterdam: North Holland, 1, 391-396.

BENVENISTE, A., METIVIER, M., and PRIOURET, P. (1987), *Algorithmes adaptatifs et approximations stochastiques*, Paris: Masson.

CELEUX, G., and DIEBOLT, J. (1985), "The SEM Algorithm: A Probabilistic Teacher Algorithm Derived From the EM Algorithm for the Mixture Problem," *Computational Statistics Quaterly, 2*, 73-82.

CELEUX, G., and GOVAERT, G. (1992), "A Classification EM Algorithm for Clustering and Two Stochastic Versions," *Computational Statistics and Data Analysis, 14*, 315-332.

CELEUX, G., and GOVAERT, G. (1995), "Gaussian Parsimonious Clustering Models," *Pattern Recognition, vol. 28, 5*, 781-793.

DEMPSTER, A. P., LAIRD, N. M., and RUBIN, D. B. (1977), "Maximum Likelihood From Incomplete Data Via the EM Algorithm (with discussion)," *Journal of the Royal Statistical Society, Series B, 39*, 1-38.

DURBIN, R., and MITCHINSON, G. (1990), "A Dimension Reduction Framework for Understanding Cortical Maps," *Nature, 343*, 644-647.

KLEIN, R.W., and DUBES, R.C. (1989), "Experiments in Projection and Clustering by Simulated Annealing," *Pattern Recognition, 22*, 213-220.

KOHONEN, T. (1982), "Self Organized Formation of Topological Correct Feature Maps," *Biological Cybernetics, 43*, 59-69.

KOHONEN, T. (1984), *Self Organization and Associative Memory*, (2nd ed), New York: Springer-Verlag.

KOHONEN, T. (1991), "Self-Organizing Maps: An Optimization Approach," in *Artificial Neural Networks*, Eds., T. Kohonen, K. Mäkisara, O. Simula and J. Kangas, Amsterdam: North Holland, 2, 981-990.

KOHONEN, T. (1993), "Things You Haven't Heard About the Self-Organizing Map," *Proceedings of 1993 IEEE International Conference on Neural Networks*, San Francisco, California, 1147-1156.

LUTTREL, S. P. (1990), "Derivation of a Class of Training Algorithms," *IEEE Transactions on Neural Networks, 1, 2*, 229-232.

MACQUEEN, J. (1967), "Some Methods for Classification and Analysis of Multivariate Observations," in *Mathematics, Statistics and Probability: 5th Berkeley Symposium*, Eds., L. M. Le Cam and J. Neyman, Volume 1, Berkeley, California: University of California Press, 281-297.

MURTAGH, F. (1995), "Interpreting the Kohonen Self-Organization Feature Map Using Contiguity-Constrained Clustering," *Pattern Recognition Letters, 16*, 399-408.

ROBBINS, H., and MONRO, S. (1951), "A Stochastic Approximation Method," *Annals of Mathematics and Statistics, 22*, 400-407.

TOMASINI, L. (1993), *Apprentissage d'une représentation statistique et topologique d'un environnement*, Thèse de doctorat, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse.

WOLFE, J. H. (1970), "Pattern Clustering by Multivariate Mixture Analysis," *Multivariate Behavioral Research, 5*, 329-350.