

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Learning the number of clusters in Self Organizing Map

Guénaél Cabanes and Younès Bennani  
*University of Paris 13, LIPN-CNRS  
France*

## 1. Introduction

The Self-Organizing Map (SOM: Kohonen (1984, 2001)) is a neuro-computational algorithm to map high-dimensional data to a two-dimensional space through a competitive and unsupervised learning process. Self-Organizing Maps differ from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space. This unsupervised learning algorithm is a popular nonlinear technique for dimensionality reduction and data visualization.

The SOM is often used as a first phase for unsupervised classification (i.e. clustering). Clustering methods are able to perform an automatic detection of relevant sub-groups or clusters in unlabeled data sets, when one does not have prior knowledge about the hidden structure of these data. Patterns in the same cluster should be similar to each other, while patterns in different clusters should not (internal homogeneity and the external separation). Clustering plays an indispensable role for understanding various phenomena described by data sets. A clustering problem can be defined as the task of partitioning a set of objects into a collection of mutually disjoint subsets. Clustering is a segmentation problem which is considered as one of the most challenging problems in unsupervised learning. Various approaches have been proposed to solve the problem (Jain & Dubes, 1988).

An efficient method to grouping problems is based on the learning of a Self-Organizing Map. In the first phase of the process, the standard SOM approach is used to compute a set of reference vectors (prototypes) representing local means of the data. In the second phase, the obtained prototypes are grouped to form the final partitioning using a traditional clustering method (e.g. K-means or hierarchical methods). Such an approach is called a two-level clustering method. In this work, we focus particular attention on two-level clustering algorithms. One of the most crucial questions in many real-world cluster applications is how to determine a suitable number of clusters  $K$ , also known as the model selection problem. Without a priori knowledge there is no simple way of knowing that number. The purpose of our work is to provide a simultaneous two-level clustering approach using SOM, by learning at the same time the structure of the data and its segmentation, using both distance and density information. This new clustering algorithm assumes that a cluster is a dense region of objects surrounded by a region of low density (Yue et al., 2004; Ultsch, 2005; Ocsa et al., 2007; Pamudurthy et al., 2007). This approach is very effective when the clusters are

irregular or intertwined, and when noise and outliers are present. The proposed clustering algorithm divides automatically a given dataset into a collection of subsets (clusters), i.e., the number of clusters is determined automatically during the learning process, i.e., no a priori hypothesis for the number of clusters is required. This approach has been tested on a set of critical clustering problems and shows excellent results compared to usual approaches.

The remainder of this chapter is organized as follows. Section 2 presents the DS2L-SOM algorithm (Density-based Simultaneous Two-Level - SOM). Section 3 describes the validation databases and experimental protocol. In section 4 we show validation results and their evaluation. Conclusion and future work perspectives are given in Section 5.

## 2. Local Density-based Simultaneous Two-Level Clustering

High dimension data may be sparse (the curse of dimensionality), making it difficult for a clustering algorithm to find any structure in the data. Indeed, when dimensionality increases, data become increasingly sparse. Definitions of density and distance between objects, which is critical for clustering and outliers detection, become less meaningful. To improve the solution for this problem, a large number of dimension reduction approaches have been developed and tested in different application domains and research communities. The main idea behind these techniques is to map each pattern into a lower dimensional space that preserves the topology of data. The reduced data present at the lower dimensional representation can be used to perform clustering more efficiently. Various approaches have been proposed for the two-level clustering problem (Aupetit, 2005; Bohez, 1998; Hussin et al., 2004; Ultsch, 2005; Korkmaz, 2006). The key idea of the two-level clustering approach based on SOM is to combine the dimensionality reduction and the fast learning capabilities of SOM in the first level to construct a new reduced vector space. Then another clustering method is applied in this new space to produce a final set of clusters at the second level (Hussin et al., 2004; Ultsch, 2005). Although the two-level methods are more interesting than the traditional approaches, the data segmentation obtained from the SOM is not optimal, since a part of information is lost during the first stage (dimensionality reduction).

We propose here a new unsupervised learning algorithm (DS2L-SOM) which learns simultaneously the structure of the data and its segmentation using both distance and density information.

### 2.1 Principle

Kohonen SOM (Kohonen, 1984, 2001) can be defined as a competitive unsupervised learning neural network. When an observation is recognized, the activation of an output cell - competition layer - inhibits the activation of other neurons and reinforces itself. It is said that it follows the so called "Winner Takes All" rule. Actually, neurons are specialized in the recognition of one kind of observation. A SOM consists in a two dimensional map of neurons which are connected to  $n$  inputs according to  $n$  weights connections  $w^{(i)}=(w_0^{(i)}, \dots, w_n^{(i)})$  and to their neighbors with topological links. The training set is used to organize these maps under topological constraints of the input space. Thus, a mapping between the input space and the network space is constructed; two close observations in the input space would activate two close units of the SOM. An optimal spatial organization is determined by the SOM from the input data, and when the dimension of the input space is

lower than three, both position of weights vectors and direct neighborhood relations between cells can be represented visually. Thus, a visual inspection of the map provides qualitative information about the map and the choice of its architecture. The winner neuron updates its prototype vector, making it more sensitive for later presentation of that type of input. This allows different cells to be trained for different types of data. To achieve a topological mapping, the neighbors of the winner neuron can adjust their prototype vector towards the input vector as well, but at a lesser degree, depending on how far away they are from the winner. Usually a radial symmetric Gaussian neighborhood function  $K_{ij}$  is used for this purpose.

## 2.2 DS2L-SOM algorithm

Connectionist learning algorithms are often presented as a minimization of a cost function. In our case, it will correspond to the minimization of the distance between the input samples and the map prototypes, weighted by a neighborhood function  $K_{ij}$ . To do that, we use a gradient algorithm. The cost function to be minimized is defined by:

$$R(w) = \frac{1}{N} \sum_{k=1}^N \sum_{j=1}^M K_{j,u^*(x^{(k)})} \|w^{(j)} - x^{(k)}\|^2$$

$N$  represents the number of learning samples,  $M$  the number of neurons in the map,  $u^*(x^{(k)})$  is the index of the neuron whose weight vector is the closest to the input pattern  $x^{(k)}$  (the best match unit: BMU), and  $K_{ij}$  is a positive symmetric kernel function: the neighborhood function. The relative importance of a neuron  $i$  compared to a neuron  $j$  is weighted by the value of the kernel function  $K_{ij}$  which can be defined as:

$$K_{i,j} = \frac{1}{\lambda(t)} \times e^{-\frac{d_1^2(i,j)}{\lambda^2(t)}}$$

$\lambda(t)$  is the temperature function modeling the topological neighborhood extent, defined as:

$$\lambda(t) = \lambda_i \left( \frac{\lambda_f}{\lambda_i} \right)^{\frac{t}{t_{\max}}}$$

$\lambda_i$  and  $\lambda_f$  are respectively the initial and the final temperature (for example  $\lambda_i = 2$ ,  $\lambda_f = 0.5$ ).  $t_{\max}$  is the maximum number allotted to the time (number of iterations for the  $x$  learning sample).  $d_1(i,j)$  is the Manhattan distance defined between two neurons  $i$  and  $j$  on the map grid, with the coordinates  $(k,m)$  and  $(r,s)$  respectively:

$$d_1(i,j) = |r - k| + |s - m|$$

The DS2L-SOM algorithm is an adaptation of the S2L-SOM algorithm (Cabanes & Bennani, 2007). In S2L-SOM, each neighborhood connection is associated with a real value  $v$  which indicates the relevance of the connected neurons. The value of this connection is adapted

during learning process. It was proved by Martinetz (Martinetz, 1993) that the so generated graph is optimally topology-preserving in a very general sense. In particular each edge of this graph belongs to the Delaunay triangulation corresponding to the given set of reference vectors. For each data, both best close prototypes are linked by a topological connection. The value of this connection will be increased, whereas the value of all other connections from the best match unit will be reduced. Thus, at the end of the training, the set of inter-connected prototypes will be an artificial image of well separated sub-groups of the whole data set. Indeed, S2L-SOM can only detect borders defined by large inter cluster distances. However, the core part of a cluster can be defined as a region with high density and, in most cases, the cluster borders are defined either by an "empty" region between clusters (i.e. large inter-cluster distances) or by a low density region (Ultsch, 2005). In the DS2L-SOM algorithm, we propose also to associate each unit  $i$  to an estimate of the local data density  $D^{(i)}$ , so as to detect local fluctuations of density, which define the borders of touching clusters (low density regions). For each data, this density value will be increased for all units, as a function of the Euclidean distance between the related prototype  $w^{(i)}$  and the data. This method of evaluation is similar to the one proposed by Pamudurthy et al. (2007). Silverman (1986) has shown that when the number of data points tends to infinity, the estimator  $D$  converges asymptotically to the true density function. One can notice that, in the DS2L-SOM algorithm, the estimation of the local density data is made during the training of the map, i.e. it is not necessary to keep the data in memory.

The DS2L-SOM learning algorithm proceeds essentially in three phases:

Input:

- Data  $X = \{x^{(i)}\}_{i=1..N}$ .
- SOM with  $M$  prototypes  $\{w^{(i)}\}_{i=1..M}$ .
- $t_{\max}$  : maximum number of iterations.

Output:

- A partition  $P = \{C_i\}_{i=1..L}$  sets of inter-connected units.
- Density values  $\{D^{(i)}\}_{i=1..M}$  associated to each unit.

1. Initialization phase:

- Initialize all neighborhood connections values  $v$  to zero.
- Initialize all unit density values  $D^{(i)}$  to zero

2. Competition phase :

- Present an input pattern  $x^{(k)}$  to the SOM.
- Choose the first BMU  $u^*$  and the second BMU  $u^{**}$ :

$$u^*(x) = \operatorname{argmin}_{1 \leq i \leq M} \|x^{(k)} - w^{(i)}\|^2$$

$$u^{**}(x) = \operatorname{argmin}_{i \neq u^*(x)} \|x^{(k)} - w^{(i)}\|^2$$

3. Adaptation phase:

- Update prototype vectors  $w^{(i)}$  of each unit  $i$  according to the learning rate  $\varepsilon(t)$ :

$$w^{(i)}(t) = w^{(i)}(t-1) - \varepsilon(t) K_{i, u^*(x^{(k)})} (w^{(i)}(t-1) - x^{(k)})$$

- Increase local density value  $D^{(i)}$  for each unit  $i$  :

$$D^{(i)}(t) = D^{(i)}(t-1) + r(t)e^{-\frac{\|x^{(k)} - w^{(i)}(t)\|^2}{2\lambda^2(t)}}$$

with

$$r(t) = \frac{1}{1 + e^{\left(-\frac{t}{t_{max}}\right)}}$$

- Let  $\mathfrak{N}(u^*)$  be the topological neighborhood of the first BMU; update the neighborhood connections values  $v$  according to the following rules:

$$v_{u^*, u^{**}}(t) = v_{u^*, u^{**}}(t-1) + r(t)|\mathfrak{N}(u^*)|$$

and

$$v_{u^*, i}(t) = v_{u^*, i}(t-1) - r(t)|\mathfrak{N}(u^*)| \\ \forall i \in \mathfrak{N}(u^*), i \neq u^{**}$$

where  $|\mathfrak{N}(u^*)|$  is the number of unit in  $|\mathfrak{N}(u^*)|$ .

4. Repeat steps 2 and 3 until  $t=t_{max}$ .
5. Extract all clusters : Let  $P = \{C_i\}_{i=1..L}$  the set of the  $L$  groups of linked units such as  $v>0$  (see Fig.1(b)).
6. Return  $P = \{C_i\}_{i=1..L}$ .

At the end of the learning process, we use a refinement algorithm, which exploits connection and density information to detect clusters:

Input:  $P$  and  $\{D^{(i)}\}_{i=1..M}$ .

Output: The refined clusters.

1. For each  $C_k \in P$  do:
  - Find the set  $M(C_k)$  of density maxima (i.e. density mode, see Fig. 1(c)).

$$M(C_k) = \{\text{unit } i \in C_k \mid D^{(i)} \geq D^{(j)}, \forall j \in \mathfrak{N}(i)\}$$

- Determine the threshold matrix:

$$S = [S(i, j)]_{i, j: 1 \dots |M(C_k)|}$$

with

$$S(i, j) = \left( \frac{1}{D^{(i)}} + \frac{1}{D^{(j)}} \right)^{-1}$$

- For all unit  $i \in C_k$ , label the unit  $i$  with one element  $label(i)$  of  $M(C_k)$ , according to an ascending density gradient along the topological connections. Each label represents a sub-cluster (Fig. 1(d)).

- For each pair of neighbors unit  $(i,j)$  in  $C_k$ , if  $label(i) \neq label(j)$  and if both  $D^{(i)} > S(label(i), label(j))$  and  $D^{(j)} > S(label(i), label(j))$  then merge the two sub-clusters (Fig. 1(e)).

2. Return refined clusters.

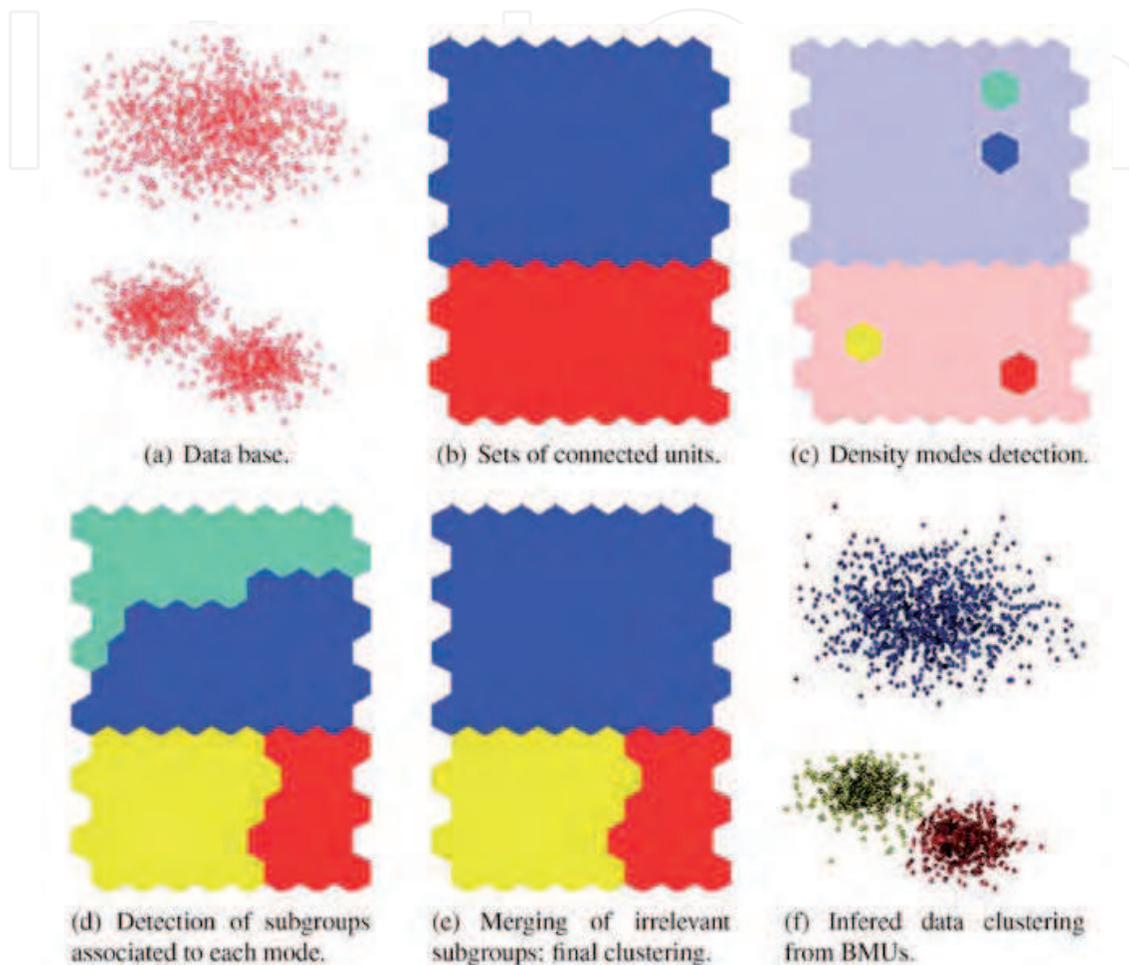


Fig. 1. Example of a sequence of the different stages of the refinement algorithm.

At the end of the learning process, the prototypes linked together by neighborhood connections having value  $v > 0$  define well separate clusters. Thus, we use a “Watersheds” method (see Vincent & Soille (1991)) on the density map of each of these clusters, in order to characterize density defined sub-clusters. For each pair of adjacent subgroups we use a density-dependent index (Yue et al., 2004) to check if a low density area is a reliable indicator of the data structure, or whether it should be regarded as a random fluctuation in the density. This process is very fast because of the small number of prototypes. The combined use of these two types of group definition can achieve good results despite the low number of prototypes in the map. This allows different cells to be trained for different types of data. To achieve a topological mapping, the neighbors of the winner neuron can adjust their prototype vector towards the input vector as well, but at a lesser degree, depending on how far away they are from the winner. Usually a radial symmetric Gaussian neighborhood function  $K_{ij}$  is used for this purpose.

### 3. Experiments

#### 3.1 Databases description

To demonstrate the effectiveness of the proposed two-level clustering method, the performances of the DS2L-SOM algorithm have been tested on 10 databases presenting various clustering difficulties (see Fig.2).

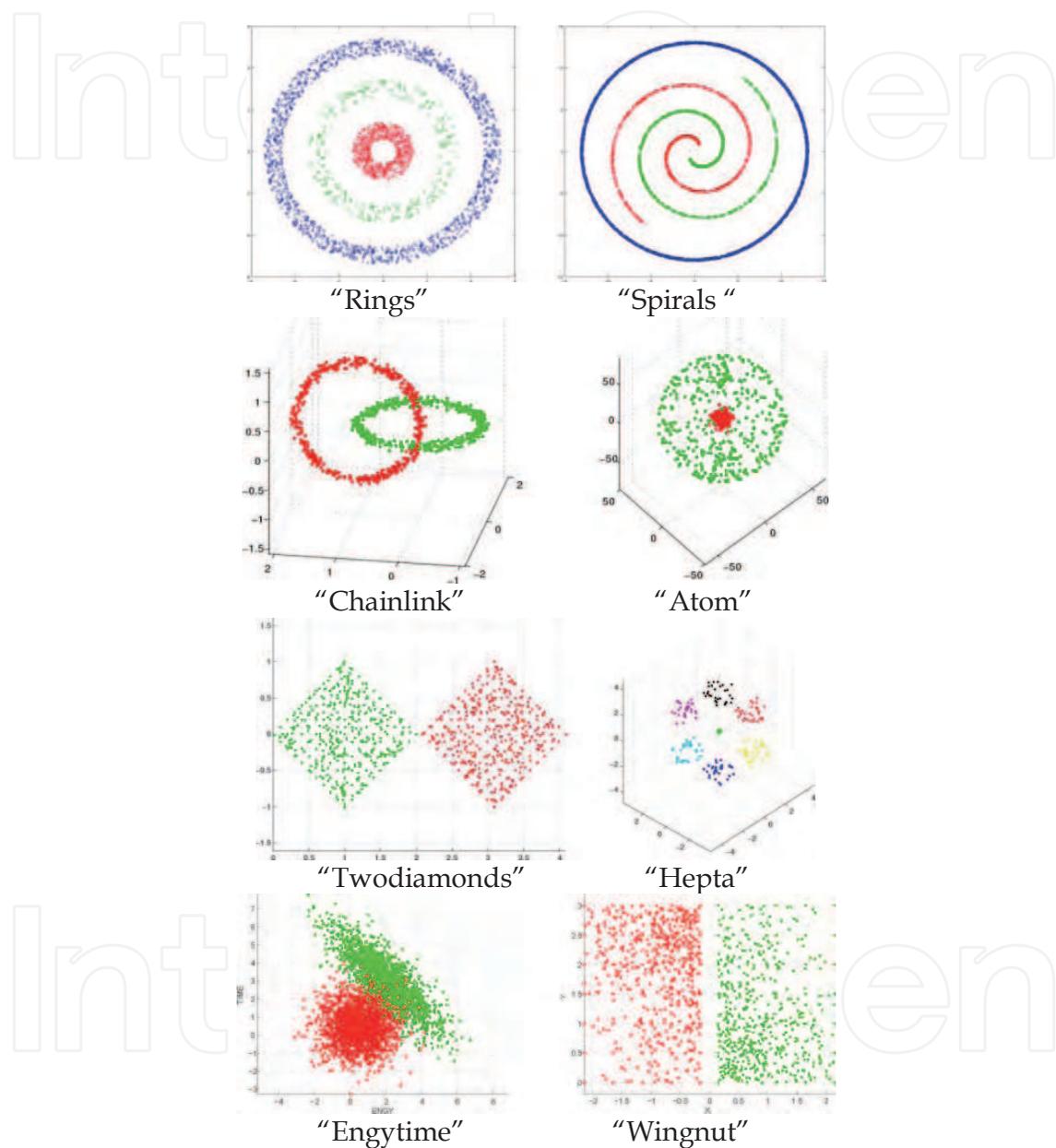


Fig. 2. Data visualizations.

The databases "Hepta", "Chainlink", "Atom", "Twodiamonds", "Engytime" and "Wingnut" come from the Fundamental Clustering Problem Suite (FCPS: Ultsch (2005)). We also generated four other interesting data bases ("Rings", "Spirals", "HighDim" and "Random"). "Rings" is made up of 3 groups in 2 dimensions not linearly separable with different densities and variances: a ring of radius 1 with 700 points (strong density), a ring of

radius 3 with 300 points (low density) and a ring of radius 5 with 1500 points (average density). "HighDim" consists of 9 quite separate groups of 100 points each one in a 15 dimensions space. "Random" is a random generation of 1000 points in 8 dimensions space. Finally "Spirals" consists of two parallel spirals of 1000 points each one in rings of 3000 points. The density of points in the spirals decreases with the radius.

### 3.2 Experimental protocol

We compared the performances of the DS2L-SOM algorithm, in term of segmentation quality and stability, to S2L-SOM (Cabanes & Bennani, 2007), and to the traditional two levels methods. The selected algorithms for comparison are K-means and SingleLinkage applied to the prototypes of the trained SOM. The Davies-Bouldin index (Davies & Bouldin, 1979) is used to determine the best cutting of the dendrogram (SingleLinkage) or the optimal number K of centroids for K-means. This index, suggested by Davies and Bouldin (Davies & Bouldin, 1979) for different values k (cluster number), is defined as in the following, in order to combine the concepts of cluster separation (denominator) and cluster compactness (numerator):

$$DB(k) = \frac{1}{K} \sum_{i=1}^K \frac{s_i + s_j}{\|c_i - c_j\|^2}$$

being  $s_i$  the square root of the average error (within-cluster variance) of cluster  $i$  with the centroid  $c_i$ . S2L-SOM and DS2L-SOM determine the number of clusters automatically and do not need to use this index.

For the single link hierarchical clustering, the proximity of two clusters is defined as the minimum of the distance between any two objects in the two different clusters. The single link technique is good at handling non-elliptical shapes, but is sensitive to noise and outliers.

In this paper the quality of the clustering has been evaluated using external criteria (Overlap Indices Rand and Jaccard) frequently used (Halkidi et al., 2001, 2002) :

$$\begin{aligned} \text{Rand} &= \frac{a_{00} + a_{11}}{a_{00} + a_{01} + a_{10} + a_{11}} \\ \text{Jaccard} &= \frac{a_{11}}{a_{01} + a_{10} + a_{11}} \end{aligned}$$

Where  $a_{11}$  denotes the number of object pairs belonging to the same label and to the same cluster,  $a_{10}$  denotes the number of pairs that belong to the same label but different clusters, and  $a_{01}$  denotes the pairs in the same cluster but with different labels. Finally,  $a_{00}$  denotes the number of object pairs sharing neither label nor cluster.

Indeed, if data-independent labels (categories) are available, the question may be asked of how well a given cluster solution corresponds to these external labels.

The concept of cluster stability is also used as an indicator for assessing the validity of data partitioning found by different algorithms. In order to evaluate the stability of the various algorithms, we use sub-sampling based method for each data bases (Ben-Hur et al., 2002). We then compute the difference between two different segmentations using the Jaccard index. We repeat this procedure several times to averaging the repeated index values. This average is regarded as a reliable estimate of the clustering stability.

#### 4. Results

The results for the external indices show that for all the databases DS2L-SOM is able to find without any error the expected data segmentation and the right number of clusters. This is not the case of the other algorithms, when the groups have an arbitrary form, or when there is no structure in the data or the groups are in contact (see Table 1 and Fig. 3).

Data	SSL	SKM	S2L-SOM	DS2L-SOM	True
HighDim	4	9	9	9	9
Chainlink	2	11	2	2	2
Atom	6	6	2	2	2
Twodiamonds	2	2	2	2	2
Rings	7	13	3	3	3
Spirals	12	10	3	3	3
Hepta	5	7	7	7	7
Random	10	15	1	1	1
Wingnut	6	11	1	2	2
Engytime	10	10	1	2	2

Table 1. Number of clusters obtained by various clustering methods (SSL = SOM+SingleLinkage, SKM = SOM+K-means).

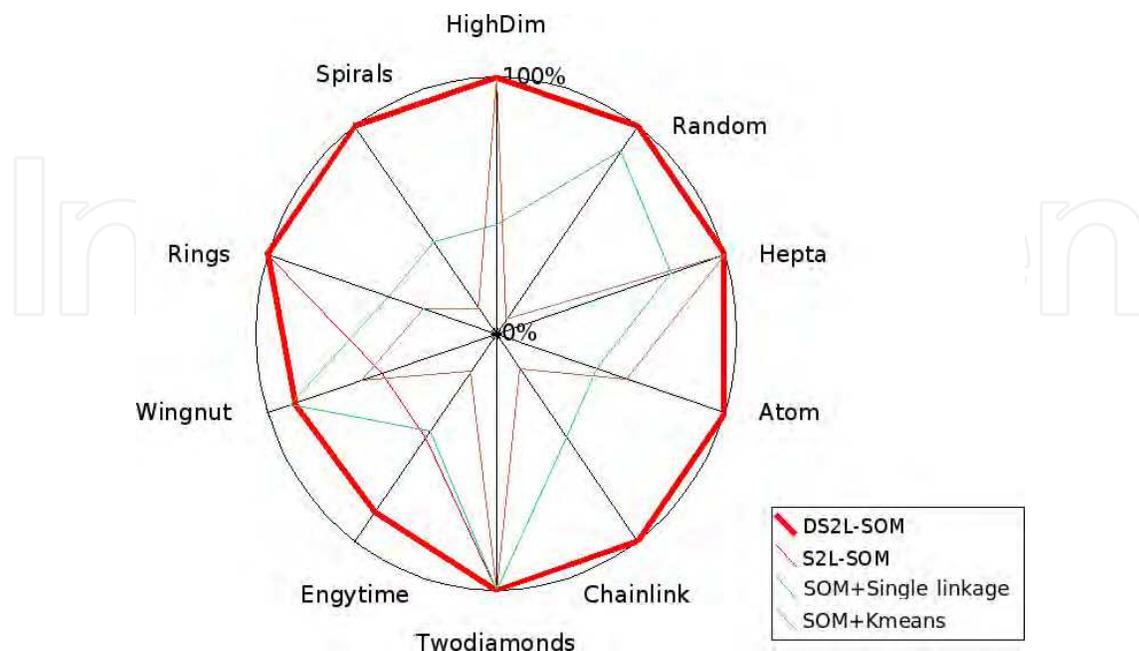


Fig. 3. Clustering quality using the Jaccard index for each algorithm on each database.

Considering the stability, the DS2L-SOM algorithm, like the S2L-SOM algorithm, shows excellent results for the data grouped in hyperspheric clusters, whatever the dimension ("Hepta" and "HighDim"), and also in the cases where the groups have arbitrary forms in two dimensions ("Rings" and "Spirals") and when the data are not structured ("Random"). It is worth noticing that in this last case the segmentation obtained by the most traditional methods is extremely unstable (Fig. 4).

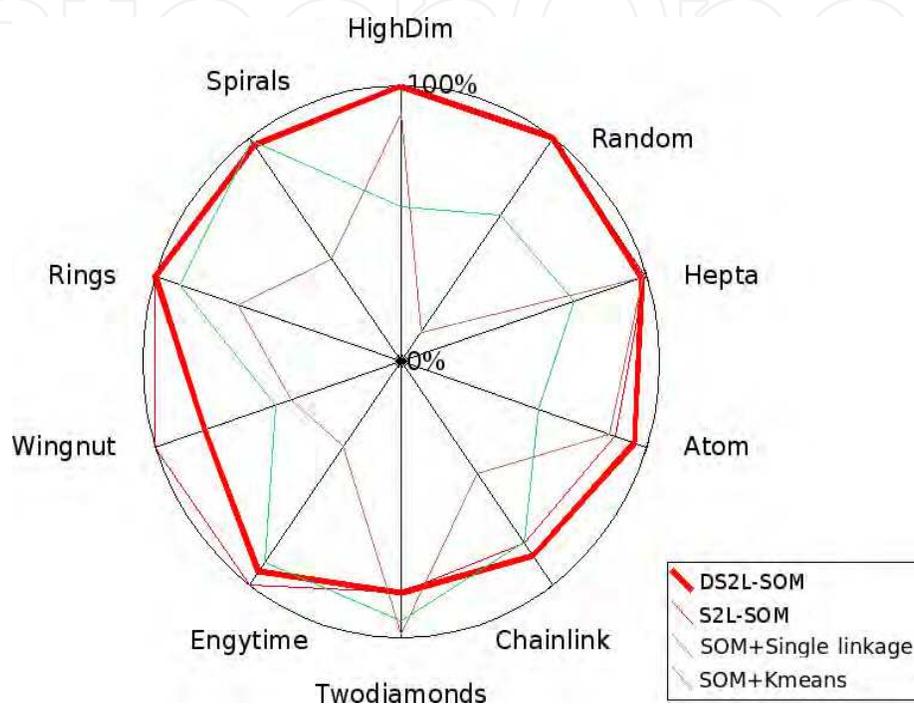


Fig. 4. Stability index for each algorithm on each database.

When the data are not linearly separable in dimensions higher than two ("Atom" and "Chainlink"), S2L-SOM and DS2L-SOM are limited by the topological constraint in two dimensions of the SOM network. Consequently, the stability of the segmentation is not maximum. However one can note that even in this case the DS2L-SOM algorithm remains more stable than the other methods. Moreover, when the clusters are defined only by the density ("Twodiamonds", "Engytime", "Wingnut"), sub-sampling may smooth the fluctuations of data density. This reduces the stability of the segmentation. In this case S2L-SOM is more stable than DS2L-SOM, because it can't separate this kind of groups and always finds one group in each sub-sample.

The results are also confirmed by visual inspection. Indeed, the DS2L-SOM clustering algorithm is a powerful tool for visualization of the obtained segmentation in two dimensions. Clusters are easily and clearly identifiable, as well as regions without data (unconnected neurons). As one can notice it from figures 5 to 9, the results obtained by the DS2L-SOM algorithm are closer to reality than those found by the other algorithms. Figures 7 and 8 show that DS2L-SOM is able to detect density-defined clusters.

In these figures, each hexagon represents a prototype of the SOM together with its associated data. Hexagons showing the same color are in the same cluster. White hexagons are not part of any cluster.

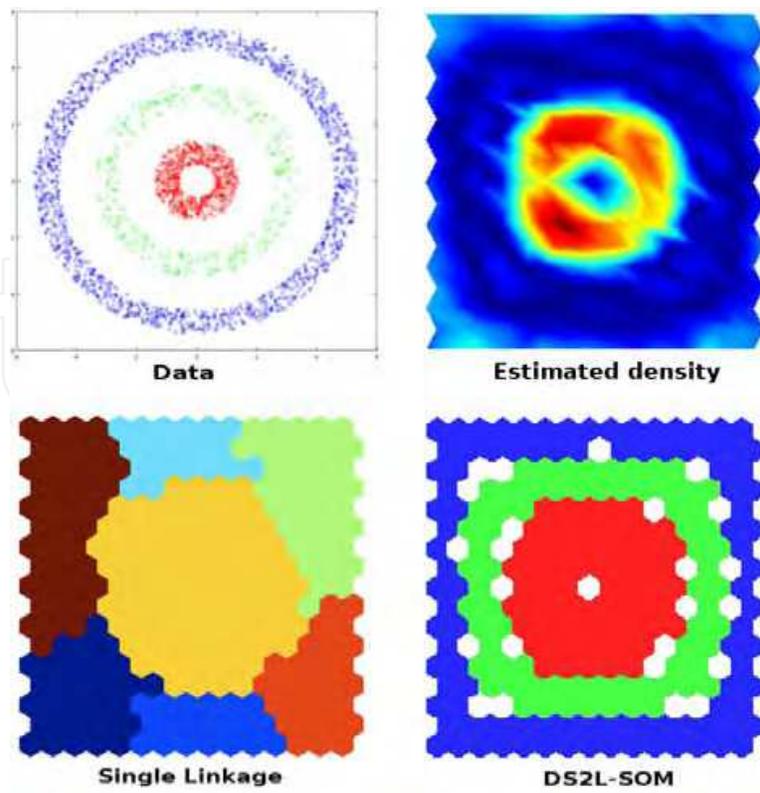


Fig. 5. Clustering of "Rings" data.

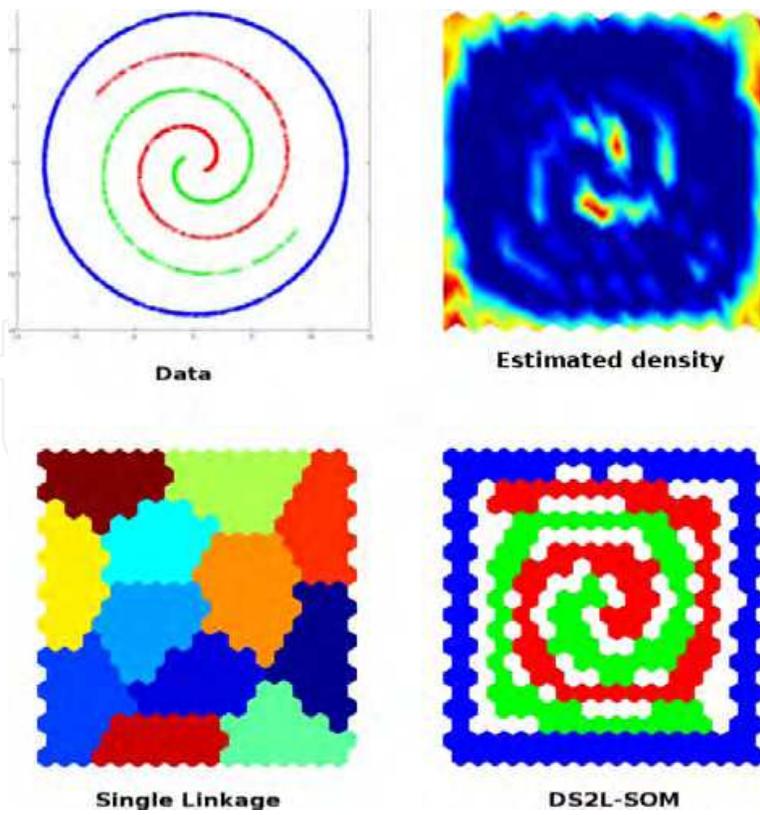


Fig. 6. Clustering of "Spirals" data.

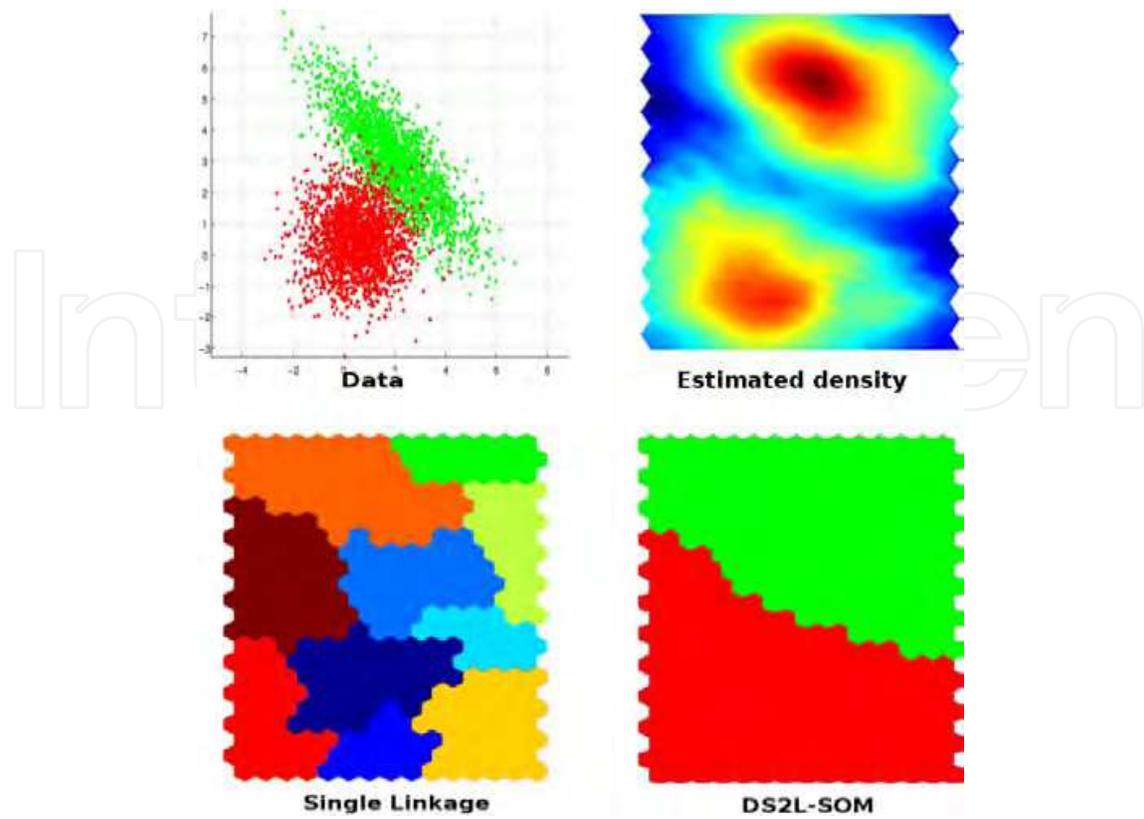


Fig. 7. Clustering of "Engytime" data.

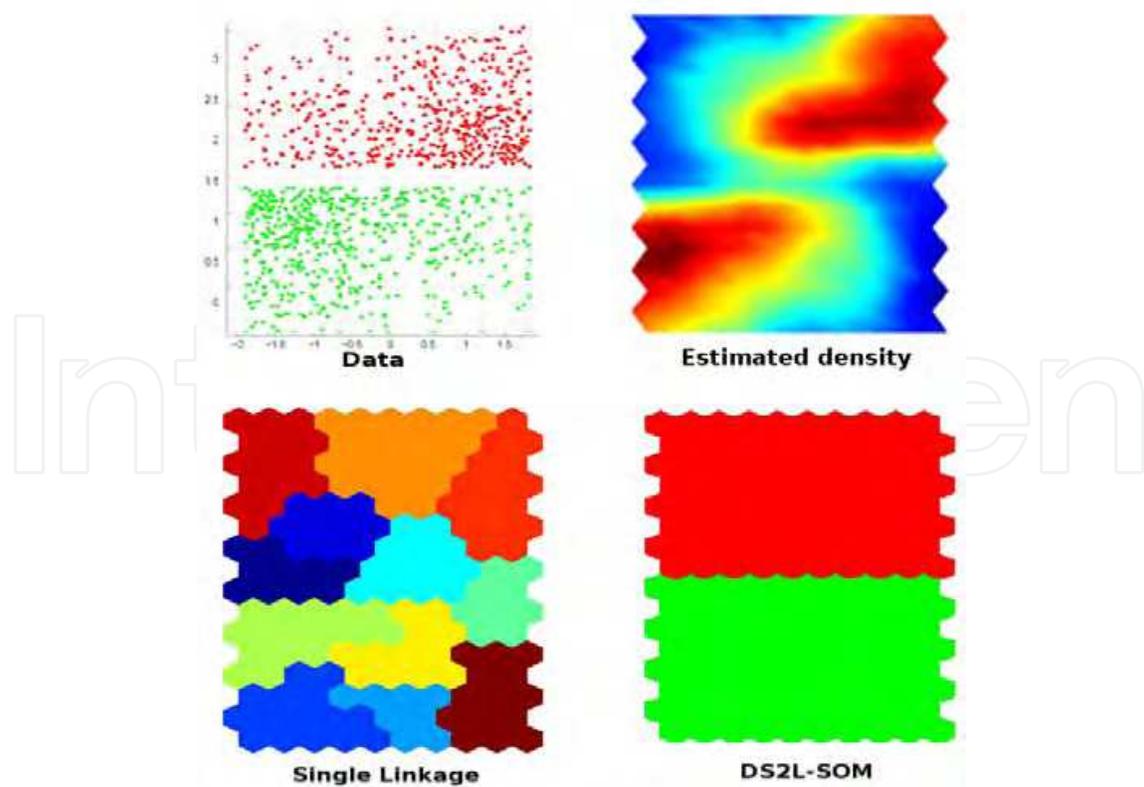


Fig. 8. Clustering of "Wingnut" data.

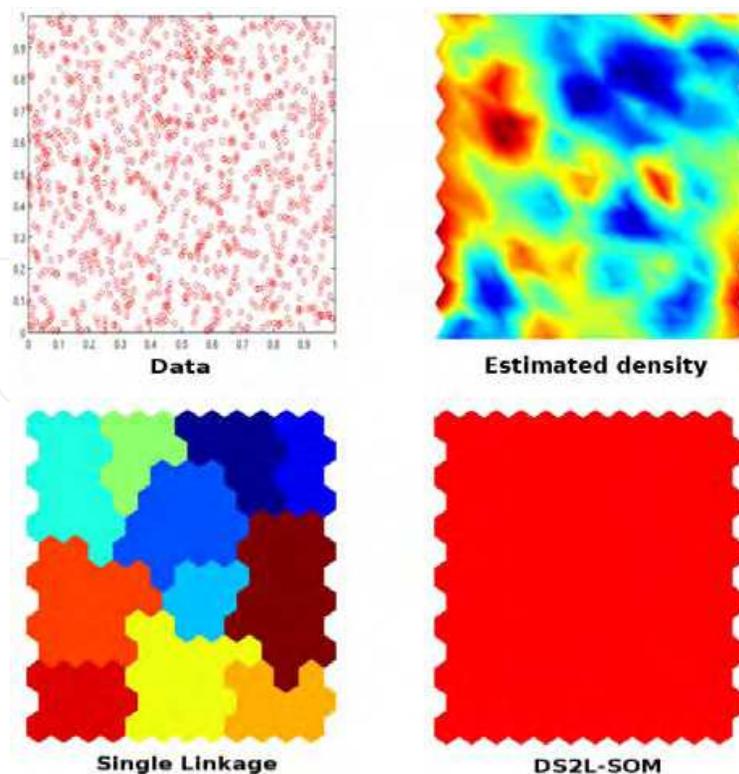


Fig. 9. Clustering of “Random” data. Data visualization is a two-dimensional projection of the “Random” database.

## 5. Conclusion

We proposed here a density-based simultaneous two-level clustering method. It uses SOM as dimensionality reduction technique and achieves an improved final clustering in the second level, using both distance and density information. The proposed algorithm DS2L-SOM locates regions of high density that are separated from one another by regions of low density. The performance of DS2L-SOM have been evaluated on a set of critical clustering problems, and compared to other two-level clustering algorithms. The experimental results demonstrate that the proposed clustering method achieves a better clustering quality than classical approaches. The results also demonstrate that DS2L-SOM is able to discover irregular and intertwined clusters, while conventional partitioning clustering algorithms can deal with convex clusters only. Finally, the number of clusters in our approach is determined automatically during the learning process, i.e., no a priori hypothesis for the number of clusters is required. In the future we plan to incorporate in the DS2L-SOM algorithm some plasticity property, to evaluate its impact on the cluster quality and stability.

## 6. Acknowledgment

This work was supported in part by the Sillages project (N° ANR-05-BLAN-0177-01) financed by the ANR (Agence Nationale de la Recherche).

## 7. References

- Aupetit, M. (2005). Learning topology with the generative gaussian graph and the EM algorithm. In *Neural Information Processing Systems (NIPS)* Vancouver, B.C., Canada.
- Ben-Hur, A.; Elisseeff, A. & Guyon, I. (2002). A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing*, 7, 6–17.
- Bohez, E. L. J. (1998). Two level cluster analysis based on fractal dimension and iterated function systems (ifs) for speech signal recognition. *IEEE Asia-Pacific Conference on Circuits and Systems*, (pp. 291–294).
- Cabanes, G. & Bennani, Y. (2007). A simultaneous two-level clustering algorithm for automatic model selection. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA'07)* Cincinnati, Ohio, USA.
- Davies, D. L. & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 1(2), 224–227.
- Halkidi, M.; Batistakis, Y. & Vazirgiannis, M. (2001). On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, 17(2-3), 107–145.
- Halkidi, M.; Batistakis, Y. & Vazirgiannis, M. (2002). Cluster Validity Methods. *SIGMOD Record*, 31(2,3), 40–45, 19–27.
- Hussin, M. F.; Kamel, M. S. & Nagi, M. H. (2004). An efficient two-level SOMART document clustering through dimensionality reduction. In *ICONIP* (pp. 158–165).
- Jain, A. K. & Dubes, R. C. (1988). *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.
- Kohonen, T. (1984). *Self-Organization and Associative Memory*. Berlin: Springer-Verlag.
- Kohonen, T. (2001). *Self-Organizing Maps*. Berlin: Springer-Verlag.
- Korkmaz, E. E. (2006). A two-level clustering method using linear linkage encoding. *International Conference on Parallel Problem Solving From Nature, Lecture Notes in Computer Science*, 4193, 681–690.
- Martinetz, T. (1993). Competitive hebbian learning rule forms perfectly topology preserving maps. In S. Gielen & B. Kappen (Eds.), *Proceedings of the International Conference on Artificial Neural Networks (ICANN-93)*, Amsterdam (pp. 427–434). Heidelberg: Springer.
- Ocsa, A.; Bedregal, C. & Cuadros-Vargas, E. (2007). DB-GNG: A constructive self-organizing map based on density. *Proceeding of International Joint Conference on Neural Networks*, (pp. 1506–1511).
- Pamudurthy, S. R.; Chandrakala, S. & Sakhar, C. C. (2007). Local density estimation based clustering. *Proceeding of International Joint Conference on Neural Networks*, (pp. 1338–1343).
- Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC.
- Utsch, A. (2005). Clustering with SOM: U\*C. In *proc. workshop on self-organizing maps* (pp. 75–82). paris, france.
- Vincent, L. & Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13, 583–598.
- Yue, S.-H.; Li, P.; Guo, J.-D. & Zhou, S.-G. (2004). Using greedy algorithm: DBSCAN revisited II. *Journal of Zhejiang University SCIENCE*, 5(11), 1405–1412.



## **Self-Organizing Maps**

Edited by George K Matsopoulos

ISBN 978-953-307-074-2

Hard cover, 430 pages

**Publisher** InTech

**Published online** 01, April, 2010

**Published in print edition** April, 2010

The Self-Organizing Map (SOM) is a neural network algorithm, which uses a competitive learning technique to train itself in an unsupervised manner. SOMs are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space and they have been used to create an ordered representation of multi-dimensional data which simplifies complexity and reveals meaningful relationships. Prof. T. Kohonen in the early 1980s first established the relevant theory and explored possible applications of SOMs. Since then, a number of theoretical and practical applications of SOMs have been reported including clustering, prediction, data representation, classification, visualization, etc. This book was prompted by the desire to bring together some of the more recent theoretical and practical developments on SOMs and to provide the background for future developments in promising directions. The book comprises of 25 Chapters which can be categorized into three broad areas: methodology, visualization and practical applications.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Guenael Cabanes and Younes Bennani (2010). Learning the Number of Clusters in Self Organizing Map, Self-Organizing Maps, George K Matsopoulos (Ed.), ISBN: 978-953-307-074-2, InTech, Available from: <http://www.intechopen.com/books/self-organizing-maps/learning-the-number-of-clusters-in-self-organizing-map>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen