DAta Mining & Exploration Program

Evolving Self Organizing Maps

User Manual

DAME-MAN-NA-0021

Issue: 1.2
Author: M. Brescia, F. Esposito

Doc. : ESOM_UserManual_DAME-MAN-NA-0021-Rel1.2

Index

TABLE INDEX

FIGURE INDEX

1 Introduction

The present document is the user guide of the data mining model Evolving Self Organizing Maps (ESOM), a data mining model that can be used to execute scientific experiments for clustering on massive data sets, formatted in one of the supported types: ASCII (columns separated by spaces), CSV (comma separated values), FITS-Table (numerical columns embedded into the fits file), VOTable, GIF, JPG and FITS-Image.

This manual is one of the specific guides (one for each data mining model available in the webapp) having the main scope to help user to understand theoretical aspects of the model, to make decisions about its practical use in problem solving cases and to use it to perform experiments through the webapp, by also being able to select the right functionality associated to the model, based upon the specific problem and related data to be explored, to select the use cases, to configure internal parameters, to launch experiments and to evaluate results.

The documentation package consists also of a general reference manual on the webapp (useful also to understand what we intend for association between functionality and data mining model) and a GUI user guide, providing detailed description on how to use all GUI features and options.
So far, we strongly suggest to read these two manuals and to take a little bit of practical experience with the webapp interface before to explore specific model features, by reading this and the other model guides.
All the cited documentation package is available from the address
http://dame.dsf.unina.it/dameware.html **, where there is also the direct gateway to the webapp.**

As general suggestion, the only effort required to the end user is to have a bit of faith in Artificial Intelligence and a little amount of patience to learn basic principles of its models and strategies.

By merging for fun two famous commercial taglines we say: "*Think different, Just do it!*"
(casually this is an example of *data* (*text) mining*...!)

2 ESOM theoretical overview

The goal of this guide is to show the use of the unsupervised model for clustering ESOM.

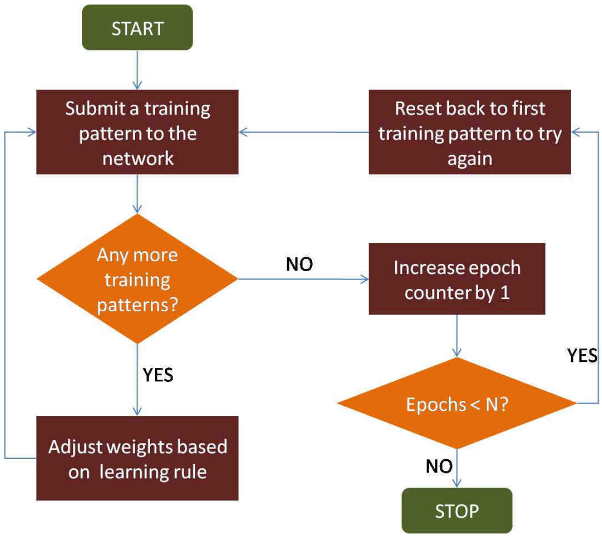


Figure 1 - Flow chart of a generic unsupervised neural network

The theory of neural network is based on computational models, introduced in 40s by McCulloch & Pitts (1943), which reproduced in a simplified way the behaviour of a biological neuron. The neural networks are self-adaptive computational models, based on the concept of learning from examples (supervised) or self-organizing (unsupervised).
The self-organizing neural networks are suitable for the solution of different problems in respect of networks with supervised training. The main use of these networks is precisely the data analysis in order to found groups having similarities (pre-processing and data clustering) or form classification (recognition of images or signals).
The supervised learning consists in the training of a network by input/target pairs that, obviously, are knows solutions of optimization problems in specific points of data space (parameters space) of problem itself (classification, approximation or functions regression). Sometimes there is not the possibility to have data relative to solution of problems but data to analyse without specific information on them (unsupervised training). A typical problem of such type is the research of class or groups of data with similar features within an unordered group of data (clustering).
Generally, clustering problems, needs the use of a competitive rule among the nodes of the network in which the winner is candidate to represents the input pattern. In the most well-known self-organizing neural network (SOM; Kohonen 2001) the nodes are placed on the top of a grid, forming a two- or three-dimensional topologically constrained space. The principal limitation of this type of structure is the static of the output layer that especially results a problem in case of on-line clustering in which is useful a network capable of evolve itself as new data are acquired.
During the years different solution were proposed. Fritzke (1994) propose the Growing Cell Structures (GCS) that introduced the incremental aspect of the network preserving a connection between nodes. One year later (Fritzke 1995) the Growing Neural Gas (GNG) remove also this aspect. In the same year, Bruske & Somemr (1995) introduced the Dynamic Cell Structures GCS (DCS-GCS) which differing from GNG slightly in the location of node insertion. However all the models described, each one of them for different reasons, implied additional computational time, which can be reduced as proposed in the Evolving Self Organizing Maps (Deng & Kasabov; 2003).

2.1 The model ESOM

The algorithm starts with a null network without any nodes. Nodes are created incrementally: when a new input pattern is presented, the prototype nodes in the network compete with each other and the connections of winner node are updated. In particular, if the two winners are not connected, a connection will be made between them. New node will be inserted into the network if none of existing nodes matches with the current input. In this case the new node also sets connections to the first two winners.
Let it be:

$x, \quad input\ pattern$
$W = \{w_1, w_2, \dots, w_N\}, \quad existing\ prototype\ set$
$\varepsilon, \quad minimum\ distance\ threshold$

A new node is inserted if:

$$\left\|w_j - x\right\| > \varepsilon, \ \forall\, w_j \in W \tag{1}$$

and it is initialized as

$$w_{N+1} = x \tag{2}$$

The eq. (2) show that a new node is inserted representing exactly the poorly matched input vector. This approach leads to a computational efficiency because other type of insertions, as the mid-point insertion used in GNG, takes a greater number of iterations. Although direct allocation in ESOM is sensitive to noise and may introduce some artefacts in clustering, this can be mitigated by automatic deletion of obsolete nodes.
When an input pattern matches well with some prototype, the activation of the winner node is defined as:

$$a_{j=} e^{\frac{-2\left\|x- w_j\right\|^2}{\varepsilon^2}} \tag{3}$$

In the ESOM model, the neighborhood of a node is defined as:

$$\Omega(i) = \{j \mid s(i,j) > 0\} \tag{4}$$

where $s(i,j)$ represent the weight of connection between nodes *i* and *j*, and the neighborhood function can be written as:

$$h_{i,b}(x) = \frac{a_i(x)}{\sum_k a_k(x)} \tag{5}$$

The weights update follow the formula:

$$\Delta\, w_i = \begin{cases} \gamma \frac{a_i}{\sum_k a_k}\,(x - w_i), & if\ i \in \Omega(b) \\ 0\ else \end{cases} \tag{6}$$

where:

$\gamma, \ learning\ rate\ typically\ constant, set\ to\ 0.05$
$b, \ BMU$

In eq. (6) clearly shows the strong analogy with classic Kohonen learning rule in which change only the definition of neighborhood.
The neighborhood concept used in ESOM, based on connection, result computationally less expensive than other methods, as the rank used in GNG, and allows to visualization of clustered structure of data (Figure 2).
In order to do this, a mechanism to delete the weak connection is required. After the presentation of an established number of pattern, the weakest connection is pruned, and this process goes on during the whole dataset.

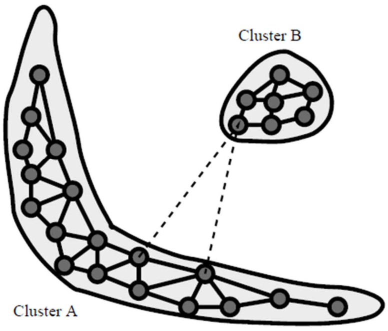


Figure 2 – Connected nodes reveals clusters

2.1.1 ESOM output layer visualization

The standard tool for visualization and interpretation of a SOM is the U-Matrix. For each node of Kohonen layer, a value will be computed according to its distance to adjacent nodes. This value can be visualized on a heat map in which light colours represents nearby nodes in the weights space, while dark colours represents distant nodes (Moutarde & Ultsch 2005). Typically, the map is represented on a greyscale as shown in Figure 3. In order to increase further the interpretability of U-Matrix is possible to overlay to each node BMU of some pattern, a colour that identify the relative cluster.

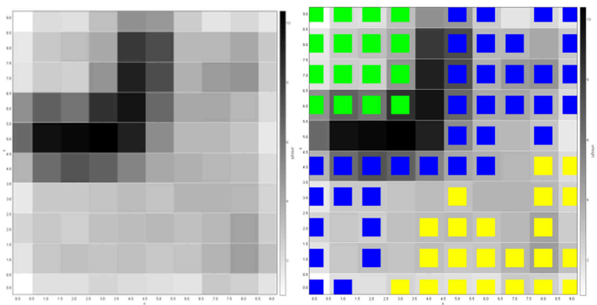


Figure 3 - Example of U-Matrix

Since in ESOM model neurons are not placed on a rigid structure such as the grid of Kohonen layer, it's evident the impossibility to use the classical U-Matrix as visualization tool. However, in order to provide a method to visualize the clustering results, a modified U-Matrix has been implemented. In this type of U-Matrix neurons are not arranged on the grid depending on the actual position in parameter space, but grouped by cluster membership.

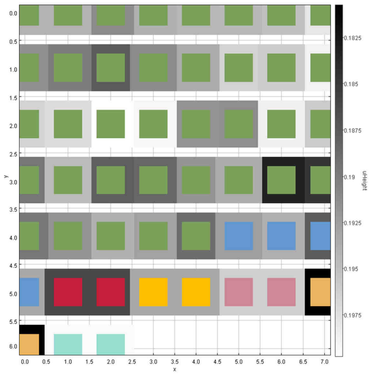


Figure 4 – Modified U-Matrix for ESOM model

As can be seen in Figure 4, the number of identified clusters is immediately clear as well as the number of nodes assigned to each of them.
As one of the peculiarity of this model is the presence of weighted connections between the output nodes, we have choose to show the average of weights of connections of each node, as gradient on gray scale. So, dark nodes are node without connections or with very weak ones.

2.2 SOM quality indicators

Good criteria to evaluate the quality of a SOM were proposed by Kiviluoto (1996):

i. What is the degree of continuity for the map topology?
ii. What is the resolution of the map topology?

A quantification of these two properties can be obtained by computation of quantization error and topographic error (Chi & Yang 2008). However the lack of classic grid of the output layer does not allow to evaluate the topographic error as it has been defined. Thus the ESOM model provides only the quantization error as a quality criterion.

2.2.1 Quantization error

The quantization error is used to the computation of similarity of pattern assigned to the same BMU, according to the following formula:

$$QE = \frac{1}{N}\sum_{i=1}^{N} \|\overrightarrow{w_{BMU\iota}} - \vec{x_\iota}\| \qquad \textbf{(7)}$$

where:

$\overrightarrow{w_{BMU\iota}}$, $\quad$ *weights vector of BMU i*
N, $\quad$ *number of pattern of dataset*
$\vec{x_\iota}$, *input vector i assigned to current BMU*

The equation (7) corresponds to the average of distance of each pattern form its BMU.

2.3 Clustering quality indicator

The results of clustering process can be evaluated using the Davies-Bouldin (DB) index. This index measures the ratio of intra-cluster and extra-cluster distances, measured from centroids (Davies & Bouldin, 1979).
The internal scatter of a cluster C_i can be written as:

$$S_{i,q} = \frac{1}{|C_i|}\sum_{\vec{x_i}\in C_i}\{|\vec{x_i} - \vec{z}|^q\}^{1/q} \ , i = 1 \ldots K \quad \textbf{(8)}$$

where $|C_i|$ is the number of pattern assigned to cluster i; x_i and z are respectively a pattern of cluster i and his centroid; q is an absolute value; K is total number of clusters.
The distance between two clusters can be written as:

$$d_{ij,t} = \left|\vec{z_i} - \vec{z_j}\right|_t = \left\{\sum_{s=1}^{D}\left|z_{si} - z_{sj}\right|^t\right\}^{1/t} \quad \textbf{(9)}$$

where z_i e z_j represent respectively centroids of clusters i and j; z_{si} e z_{sj} denotes the absolute value of the difference between vectors z_i and z_j computed on dimension s; D is the total number of pattern; t is an absolute value.
So the DB index can be written as:

$$DB = \frac{1}{k}\sum_{i=1}^{K}\max_{j,j\neq i}\left\{\frac{S_{i,q}+S_{j,q}}{d_{ij,t}}\right\} \quad \textbf{(10)}$$

Low values of this index indicate a better clustering. However, note that on non-linearly divisible dataset could not be objective.
A more objective evaluation can be obtained if the cluster of each input data is known. In such case is possible to computes the Index of Clustering Accuracy (ICA) and the Index of Clustering Completeness (ICC).
Let it be:

$NC_t, number\ of\ tehoretical\ clusters$
$NC_c, number\ of\ clusters\ found$
$NC_d, number\ of\ disjoint\ clusters$

Two theoretical clusters are disjoint if the intersection of the label assigned by clustering process in the two clusters is the empty set.

$$ICA = \frac{|NC_c - NC_t|}{NC_c + NC_t} \quad \textbf{(11)}$$

$$ICC = 1 - \frac{NC_d}{NC_t} \quad \textbf{(11)}$$

Low values of these indices reflects best results.

3 Use of the ESOM

For the user the ESOM offer three use cases:

- ***Train***
- ***Test***
- ***Run***

Additionally to use cases just described, is possible to perform a Train starting form a previously trained network. This use case is called ***Resume Training.***
A typical complete experiment consists of the following steps:

1. **Train** the network with a dataset as input; then store as output the final weight matrix (best configuration of trained network weights);
2. **Test** the trained network with a dataset containing both input and target features, in order to verify training quality;
3. **Run** the trained and tested network with new datasets. The Run use case implies the simple execution of the trained and tested model, like a generic static function.

3.1 Input

We also remark that massive datasets to be used in the various use cases are (and sometimes must be) different in terms of internal file content representation. Remind that it is possible to use one of the following data types:

- ASCII (extension .dat or .txt): simple text file containing rows (patterns) and columns (features) separated by spaces;
- CSV (extension .csv): Comma Separated Values files, where columns are separated by commas;
- FITS (extension .fits or .fit): fits files containing images and/or tables;
- VOTABLE (extension .votable): formatted files containing special fields separated by keywords coming from XML language, with more special keywords defined by VO data standards;
- JPEG (extension .jpg or .jpeg): image files;
- PNG (extension .png): image files;
- GIF (extension .gif): image files;

3.2 Output

In terms of output, the following file are obtained:

FILE	DESCRIPTION	REMARKS
E_SOM_Train_Network_Configuration.txt	File containing the parameters of a trained network.	Must be moved to File Manager tab to be used for test and run use cases
E_SOM_Train_Status.log E_SOM_Test_Status.log E_SOM_Run_Status.log	File containing details on the executed experiment	
E_SOM_Train_Results.txt E_SOM_Test_Results.txt E_SOM_Run_Results.txt	File that, for each pattern, reports ID, features, BMU, cluster and activation of winner node	
E_SOM_Train_Normalized_Results.txt E_SOM_Test_Normalized_Results.txt E_SOM_Run_Normalized_Results.txt	File with same structure of precedent described file, but with normalized features	The file is produced only if normalization of dataset was requested.
E_SOM_Train_Histogram.png E_SOM_Test_Histogram.png E_SOM_Run_Histogram.png	Histogram of clusters found	
E_SOM_Train_Validity_Indices.txt E_SOM_Test_Validity_Indices.txt E_SOM_Run_Validity_Indices.txt	File that reports the validity indices of the experiment.	Quantization error and DB index are always produced. ICA and ICC are produced

		only in Test use case.
E_SOM_Train_U_matrix.png E_SOM_Test_U_matrix.png E_SOM_Run_U_matrix.png	U-Matrix image	
E_SOM_Train_Output_Layer.txt E_SOM_Test_Output_Layer.txt E_SOM_Run_Output_Layer.txt	File that, for each node of output layer, reports ID, coordinates, clusters, number of pattern assigned and Uheight value.	The Uheight value is used to generate the U-Matrix. In ESOM model the coordinates are computed according to cluster membership
E_SOM_Train_Clusters.txt E_SOM_Test_Clusters.txt E_SOM_Run_Clusters.txt	File that, for each clusters, reports label, number of pattern assigned, percentage of association respect total number of pattern and its centroids.	
E_SOM_Train_Clustered_image.png E_SOM_Test_Clustered_image.png E_SOM_Run_Clustered_image.png	Image that show the effect of the clustering process	The file is produced only if input dataset is an image
E_SOM_Train_Clustered_image.txt E_SOM_Test_Clustered_image.txt E_SOM_Run_Clustered_image.txt	File that, for each pixel, reports ID, coordinates, features and cluster assigned	The file is produced only if input dataset is an image
E_SOM_Train_Datacube_image.zip E_SOM_Test_Datacube_image.zip E_SOM_Run_Datacube_image.zip	Archive that includes the clustered images of each slice of a datacube	The file is produced only if input dataset is a datacube

Table 1 – Output file list

3.3 Experiment parameter setup

There are several parameters to be set to achieve training, specific for network topology and learning algorithm setup. In the experiment configuration there is also the Help button, redirecting to a web page dedicated to support the user with deep information about all parameters and their default values.

We remark that all parameters labeled by an asterisk are considered as required. In all other cases the fields can be left empty (default values are used and shown in the help web pages).
The following table reports the web page addresses for all clustering models and related use cases, subject of this manual.

Functionality+Model	USE CASE	SETUP HELP PAGE
Clustering_ESOM	ALL	http://dame.dsf.unina.it/clustering_esom.html
	train	http://dame.dsf.unina.it/clustering_esom.html#train
	test	http://dame.dsf.unina.it/clustering_esom.html#test
	run	http://dame.dsf.unina.it/clustering_esom.html#run

Table 2 – List of model parameter setup web help pages available

4 Examples

This section is dedicated to show some practical examples of the correct use of the web application.
Not all aspects and available options are reported, but a significant sample of features useful for beginners of DAME suite and with a poor experience about data mining methodologies with machine learning algorithms.
In order to do so, very simple and trivial problems will be described.
Further complex examples will be integrated here in the next releases of the documentation.

4.1 First example: Iris Dataset

This example shows the use of the ESOM model applied to the dataset Iris.
The first step consists in the creation of a new workspace named for example e**somExp** and the input dataset, **iris.txt,** must be uploaded in the workspace just created.

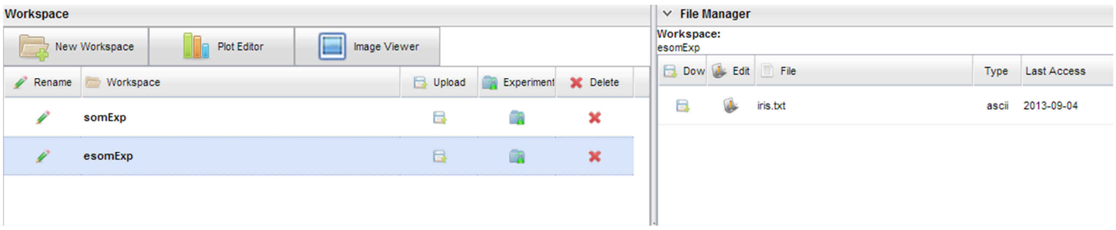


Figure 5 – The starting point, with a Workspace (esomExp) created and input dataset uploaded

4.1.1 Train Use Case

Let suppose we create an experiment named **EsomIris** and we want to configure it. After creation, the new configuration tab is open. Here we select **Clustering_E_SOM,** which indicates the functionality and the model. We select also **Train** as use case.

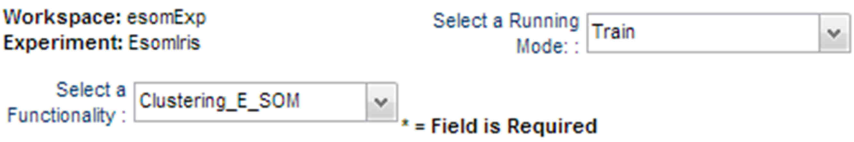


Figure 6 – Selection of functionality and use case

Now we have to configure parameters for the experiment. In particular, we will leave empty the not required fields (labels without asterisk).
As alternative, you can click on the Help button to obtain detailed parameter description and their default values directly from the web application.
We give **iris.txt** as training dataset, specifying:

- **dataset type:** 0, which is the value indicating an ASCII file
- **input nodes**: 4, because 4 are the columns in input dataset;
- **epsilon**: 0.5
- **pruning frequency**: 10

Note that the values of **epsilon** and **pruning frequency** can have a great influence on the results of the experiment. Unfortunately these values can be only set by a try & error process.

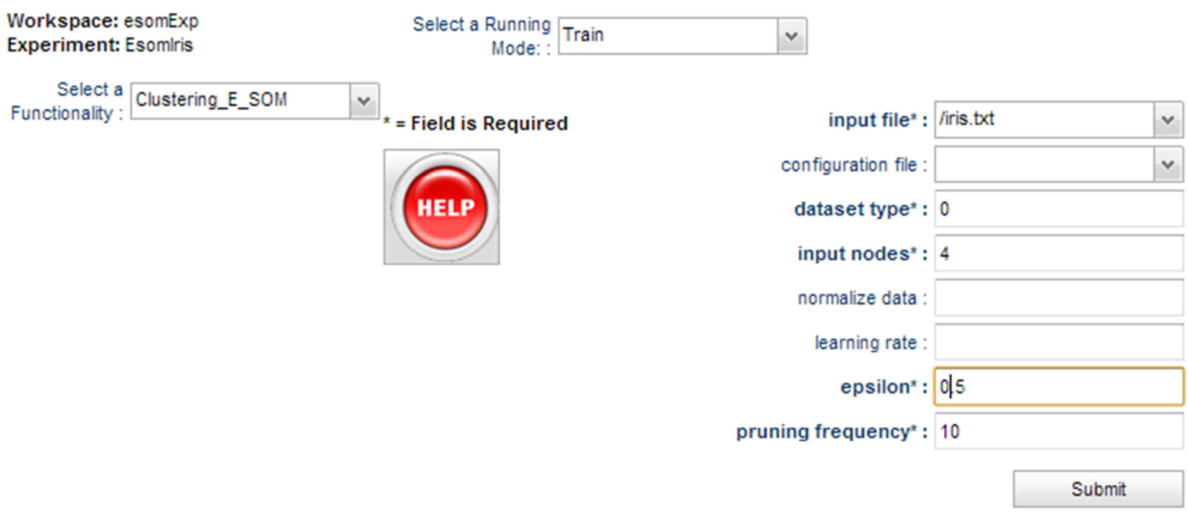


Figure 7 – The EsomIris experiment configuration tab

After submission, the experiment will be executed and a message will be shown when the execution is completed.

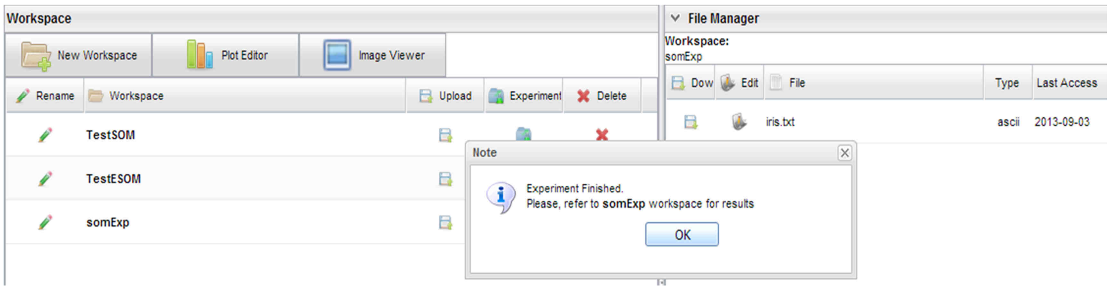


Figure 8 – Experiment finished message

The list of output files, obtained at the end of the experiment (available when the status is "ended"), is shown in the dedicated section. Each file can be downloaded or moved in the Workspace.

Figure 9 – List of output file produced

4.1.2 Test Use Case

In this paragraph is shown how execute a Test Use Case starting from a Train previously executed. Test use case is useful to evaluating the executed clustering by the indices described in paragraph **Errore. L'origine riferimento non è stata trovata.**. In order to do this, referring to the example shown above, we have to move the file **E_SOM_Network_Configuration.txt** in the Workspace. Moreover, in order to execute a Test, we need a file with one single column, with the target clusters of each pattern. Also this file must be uploaded in the Workspace.

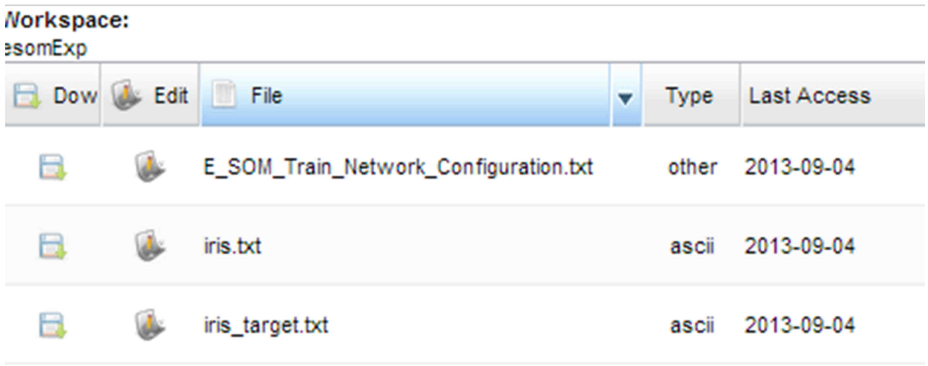


Figure 10 – Moving configuration file in the Workspace and uploading of target clusters file

Now we have to create a new experiment and choose the functionality, **Clustering_E_SOM**, and select **Test** as use case. For this model, test has only five mandatory parameters:

- **input file:** iris.txt
- **configuration file:** file produced by a Train use case, which contains experiment parameters
- **dataset target file:** file that report the cluster of each pattern present in the input dataset
- **dataset type:** 0, which indicates and ASCII input file

After submission, the experiment will be executed and will produced the output file expected.

5 Appendix – References and Acronyms

Abbreviations & Acronyms

A & A	Meaning	A & A	Meaning
AI	Artificial Intelligence	KDD	Knowledge Discovery in Databases
ANN	Artificial Neural Network	IEEE	Institute of Electrical and Electronic Engineers
ARFF	Attribute Relation File Format	INAF	Istituto Nazionale di Astrofisica
ASCII	American Standard Code for Information Interchange	JPEG	Joint Photographic Experts Group
BoK	Base of Knowledge	LAR	Layered Application Architecture
BP	Back Propagation	MDS	Massive Data Sets
BLL	Business Logic Layer	MLC	Multi Layer Clustering
CC	Connected Components	MLP	Multi Layer Perceptron
CSOM	Clustering SOM	MSE	Mean Square Error
CSV	Comma Separated Values	NN	Neural Network
DAL	Data Access Layer	OAC	Osservatorio Astronomico di Capodimonte
DAME	DAta Mining & Exploration	PC	Personal Computer
DAMEWARE	DAME Web Application REsource	PI	Principal Investigator
DAPL	Data Access & Process Layer	REDB	Registry & Database
DL	Data Layer	RIA	Rich Internet Application
DM	Data Mining	SDSS	Sloan Digital Sky Survey
DMM	Data Mining Model	SL	Service Layer
DMS	Data Mining Suite	SOFM	Self Organizing Feature Map
FITS	Flexible Image Transport System	SOM	Self Organizing Map
FL	Frontend Layer	SW	Software
FW	FrameWork	TWL	Two Winners Linkage
GRID	Global Resource Information Database	UI	User Interface
GSOM	Gated SOM	URI	Uniform Resource Indicator
GUI	Graphical User Interface	VO	Virtual Observatory
HW	Hardware	XML	eXtensible Markup Language

Table 3 – Abbreviations and acronyms

Reference & Applicable Documents

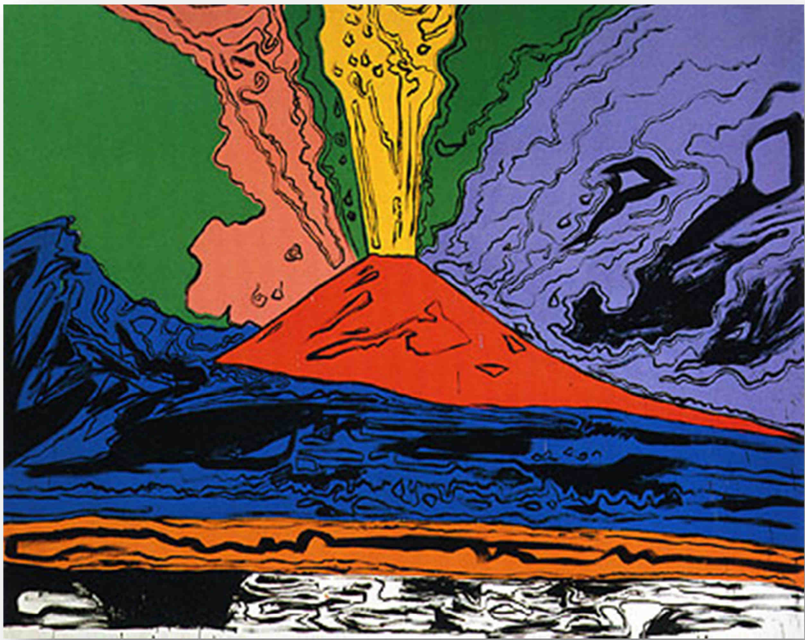
Title / Code	Author	Date
Dynamic cell structure learns perfectly topology preserving map. Neural Comput. 7 845–865	Bruske J., Sommer G.	1995
A Two-stage Clustering Method Combining Ant Colony SOM and K-means. Journal of Information Science and Engineering 24, 1445-1460	Chi S-C., Yang C-C.	2008
A cluster separation measure". IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 1, 224-227	Davies D.L., Bouldin D.W.	1979
On-line pattern analysis by evolving self-organizing maps. Neurocomputing 51, Elsevier, 87-103	Deng D., Kabasov N	2003
Improved interpretability of the unified distance matrix with connected components". Proceedings of the 2011 International Conference on Data Mining	Hamel L., Brown C.W	2011
Extending the Kohonen self-organizing map networks for clustering analysis". Computational Statistics & Data Analysis. Vol. 38, 161-180	Kiang M. Y	2001
Topology preservation in self-organizing maps". *Proceedings of the International Conference on Neural Networks.* 294-299	Kiviluoto K	1996
Self-Organizing Maps". 3rd ed., Springer	Kohonen T	2001
U*F Clustering: A new performant cluster-mining method on segmentation of self-organizing map". Proceedings of WSOM '05, September 5-8, Paris, France, 25-32	Moutarde F., Ultsch A.	2005
*Clustering with SOM: U*C.* Proc. Workshop on Self-Organizing Maps, Paris, France. 75-82	Ultsch, A	2005
Clustering of the Self-Organizing Map". IEEE Transactions on neural networks. Vol. 11, No. 3, 586-600	Vesanto J., Alhoniemi E	2000
A K-means clustering algorithm". Applied Statistics, 28, 100–108	Hartigan, J. A., Wong, M. A	1979

Table 4 – Reference Documents

ID	Title / Code	Author	Date
A1	SuiteDesign_VONEURAL-PDD-NA-0001-Rel2.0	DAME Working Group	15/10/2008
A2	project_plan_VONEURAL-PLA-NA-0001-Rel2.0	Brescia	19/02/2008
A3	statement_of_work_VONEURAL-SOW-NA-0001-Rel1.0	Brescia	30/05/2007
A4	mlpGP_DAME-MAN-NA-0008-Rel2.0	Brescia	04/04/2011
A5	pipeline_test_VONEURAL-PRO-NA-0001-Rel.1.0	D'Abrusco	17/07/2007
A6	scientific_example_VONEURAL-PRO-NA-0002-Rel.1.1	D'Abrusco/Cavuoti	06/10/2007
A7	frontend_VONEURAL-SDD-NA-0004-Rel1.4	Manna	18/03/2009
A8	FW_VONEURAL-SDD-NA-0005-Rel2.0	Fiore	14/04/2010
A9	REDB_VONEURAL-SDD-NA-0006-Rel1.5	Nocella	29/03/2010
A10	driver_VONEURAL-SDD-NA-0007-Rel0.6	d'Angelo	03/06/2009
A11	dm-model_VONEURAL-SDD-NA-0008-Rel2.0	Cavuoti/Di Guido	22/03/2010
A12	ConfusionMatrixLib_VONEURAL-SPE-NA-0001-Rel1.0	Cavuoti	07/07/2007
A13	softmax_entropy_VONEURAL-SPE-NA-0004-Rel1.0	Skordovski	02/10/2007
A14	*Clustering con Modelli Software Dinamici*. Seminario Dip. di Informatica, Università degli Studi di Napoli Federico II, http://dame.dsf.unina.it/documents.html	Esposito F.	2013
A15	dm_model_VONEURAL-SRS-NA-0005-Rel0.4	Cavuoti	05/01/2009
A16	DMPlugins_DAME-TRE-NA-0016-Rel0.3	Di Guido, Brescia	14/04/2010
A17	BetaRelease_ReferenceGuide_DAME-MAN-NA-0009-Rel1.0	Brescia	28/10/2010
A18	BetaRelease_GUI_UserManual_DAME-MAN-NA-0010-Rel1.0	Brescia	03/12/2010
A19	SOM and 2-stage clustering models Design and Requirements. som_DAME-SPE-NA-0014-Rel4.0	Esposito, Brescia	2013

Table 5 – Applicable Documents

__oOo__



DAME Program
“we make science discovery happen”