SELF-ORGANIZING MAPS AS A TOOL FOR COMPARISON OF TWO CLASSIFICATIONS

Anna Bartkowiak[1], Niki Evelpidou[2], Andreas Vassilopoulos[2]
[1] University of Wrocław, Poland
[2] University of Athens, Greece

Abstract

Kohonen's self organizing maps are a widely recognized tool for data reduction and data visualization. They may be also used for visualization of results of classification of data vectors into several groups of data. Kohonen's maps may be also used for a graphical comparison of classification results obtained by two different algorithms.

The presented concepts are illustrated using the Sifnos erosion data (Gournelos et al., 2002).

Keywords: Self-organizing maps, erosion risk, Sifnos (Cyclades), visualization of multivariate data, classification.

1 Introduction

Visualization of multivariate data is a very important topic. We use for that purpose Kohonen's self organizing maps (SOMs). The method performs a clustering of the data with preserving their topology in the original data space. The constructed map is subdivided into smaller regular units (in our case: hexagons) which represent convex regions in the data space. The regions are called Voronoi regions. To obtain the correspondence between the data space regions and the map units, an unsupervised learning algorithm developed in the framework of neural networks is applied. Specifically, we use the method and algorithm proposed by Kohonen [3]. The considerations are illustrated using the erosion risk data gathered at the island Sifnos and described by Gournelos et al. [1].

The paper is organized as follows: In Section 2 we describe shortly the method, i.e. under which principles the SOM is constructed, and what kind of information can be shown in the map. We illustrate our considerations with maps constructed for the Sifnos data. In Section 3 we show how the basic SOM can display results of classification into several groups; also how classification results obtained by two different algorithms may be compared. Section 4 contains some concluding remarks.

2 Self organizing maps – Kohonen's SOMs

2.1 A real data set with 3 variables – the Sifnos erosion data

Let $\mathbf{X}$ denote a rectangular data table of size $n \times p$, with non-missing values. The data table comprises measurements of p variables (traits) over n objects. Each row (no. i) of the table comprises values of the observed variables recorded for the ith object. The values contained in the ith row will be referred to as the i-th data vector or the ith data point and denoted as $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$ $(i = 1, \ldots, n)$. Each data vector with its p components may be considered as a data point in the Cartesian space R^p.

In the following we will consider a real data set containing observations of $p = 3$ risk factors for $n = 123$ basins of the island Sifnos (Cyclades, Greece). The description of the risk factors and of the data may be found in the paper by Gournelos et al. [1]

and Gournelos [2]. The considered risk factors are: $x1$ - drainage density, $x2$ - slope (inclination), and $x3$ - vulnerability (called also erodibility). The values of each variable were normalized to be contained in the interval [0, 1]. This was achieved by dividing each variable by its maximal value. Because there are only 3 variables, the data set may be visualized in the form of a 3–dimensional spin plot. Such spin plot is shown in Figure 1. Each point in the plot represents a basin in the island Sifnos.

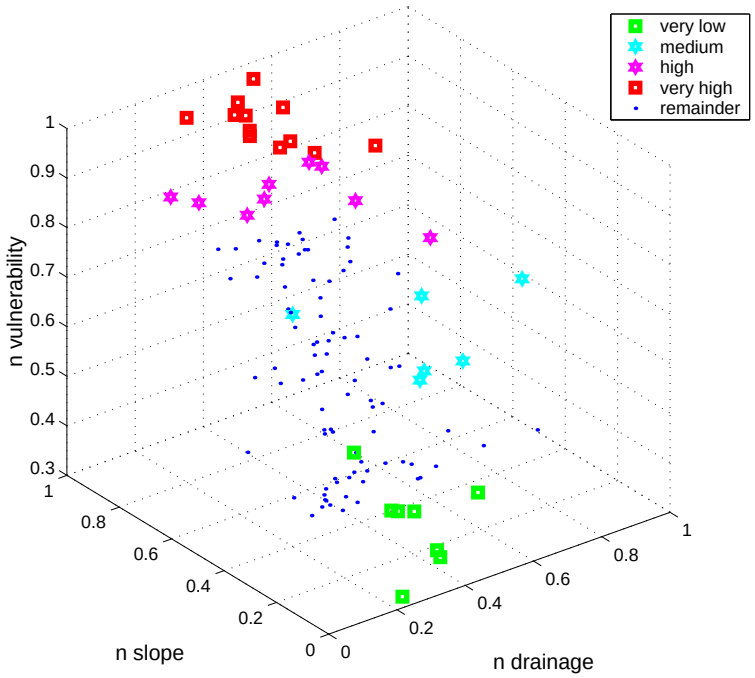

Figure 1: *Spin plot visualizing the 3 variables of the Sifnos data: normalized drainage, normalized slope and normalized vulnerability. Some data points denoting basins with very high, high, medium, and very low erosion risk are marked by special symbols. These points were used as training data for prediction of erosion risk.*

In Figure 1 we have exhibited the entire set of the Sifnos data comprising observations of $n = 123$ basins of that island. The majority of the points is marked by dots. Some of the points are marked with special symbols indicating category of erosion risk for the respective basins (very low, low, medium, high and very high). Together there are 35 such specially marked points – they were used as training data for a neural network to teach it how to predict erosion risk.

Figure 1 was constructed for 3–dimensional data. What about data with more then 3 dimensions?

2.2 The idea of Kohonen's self organizing maps (SOMs)

Kohonen [3, 6] had the bright ideas

- to quantify the data space into neighboring regions and establish one representative point for each region (the process is called *quantization of the data space*),
- to visualize the representative points in a low dimensional *map*, usually with 2 or 3 dimensions.

Kohonen proposed also algorithms realizing these ideas.

The regions obtained in the process of quantization of the data space are called Voronoi regions. The points representative for the Voronoi regions are called *codebook vectors* or *codebook points*. Each point $\mathbf{x}_i$ $(i = 1, \ldots, n)$ of the data space may be assigned to one of the Voronoi regions: namely that one containing the codebook point which is the closest to the point $\mathbf{x}_i$. Such codebook point found for a given $\mathbf{x}$ is called its BMU, i.e. its best matching unit.

The introduced concepts of Voronoi regions, codebook vectors and BMUs have important implications.

First of all, replacing the observed data points by their BMUs - playing the role of their representatives - permits for a substantial reduction of the data, because one codebook vector may represent many data points. Next: Making a projection of the codebook points onto a plane (generally: to a space of lower dimensions) permits to obtain a comprehensive view of the entire data cloud. The projection plane is called a *map*.

The map is a specific neural network, in which the neurons are exposed to a learning process by competition. The neural network learns – from the available training data – to reflect the multivariate structure of the data. Because it is an unsupervised learning, the obtained map is called a *self-organizing map*.

It is beyond the scope of the paper to go into details of the learning algorithm. The reader is referred to the quoted literature and the references therein [3, 6, 5].

2.3 Construction of the SOM for the Sifnos data

How to design the map exhibiting the projection of the Voronoi regions and codebook vectors? Generally, the map may have different shapes - the most popular shape is a sheet organized into hexagons or rectangles.

We have chosen for our visualization the hexagon lattice, which seems to be the most appealing and the most frequently used in applications.

Next we had to establish the size $m1 \times m2$ of the map.

Some practical advises given by Vesanto et al. [6] are: The number of the units of the map (denoted in the following by the symbol m) should be approximately equal to 5*sqrt(n), with n denoting the number of data points. The size of the map might be then based on the ratio between the two biggest eigenvalues of the covariance matrix of the given data. The side lengths of the map ($m1$ and $m2$) might then be set so that their product is as close to m as possible. Thus it should hold: $m1 \times m2 \approx m$. We should also take into account the quality of the constructed map, which means, that we should consider the errors of the representation of the analyzed data by the constructed map.

Generally two kinds of errors are considered:

1. Average quantization error (q) defined as the average distance - in the data space - from each data point to its BMU, i.e. to its closest codebook point.
2. Topographic error (t) defined as percentage of data points for which the BMU and the second BMU are not neighbor maps units.

According to these principles the number of units (m) of the map for our data should amount: $m \approx 55$.

The eigenvalues calculated for our data are: $\lambda_1 = 0.69068$, $\lambda_2 = 0.181643$, $\lambda_3 = 0.127678$.

Taking the above, we have considered lattices with the following side lengths: $11 \times 5 = 55$, $9 \times 6 = 54$ and $8 \times 7 = 56$. The lattice 8×7 was dropped, because it gave a positive topographic error, while for the remaining two lattices the topographic error was equal to zero. The lattices 11×5 and 9×6 gave similar quantization errors: $q = 0.083$ and $q = 0.084$. Taking into account all three eigenvalues we retained finally the structure 9×6 with an average quantization error $q = 0.08398$.

The map constructed for the Sifnos data is exhibited in Figure 2. In fact, that figure exhibits 4 maps, each displaying different information about the data.

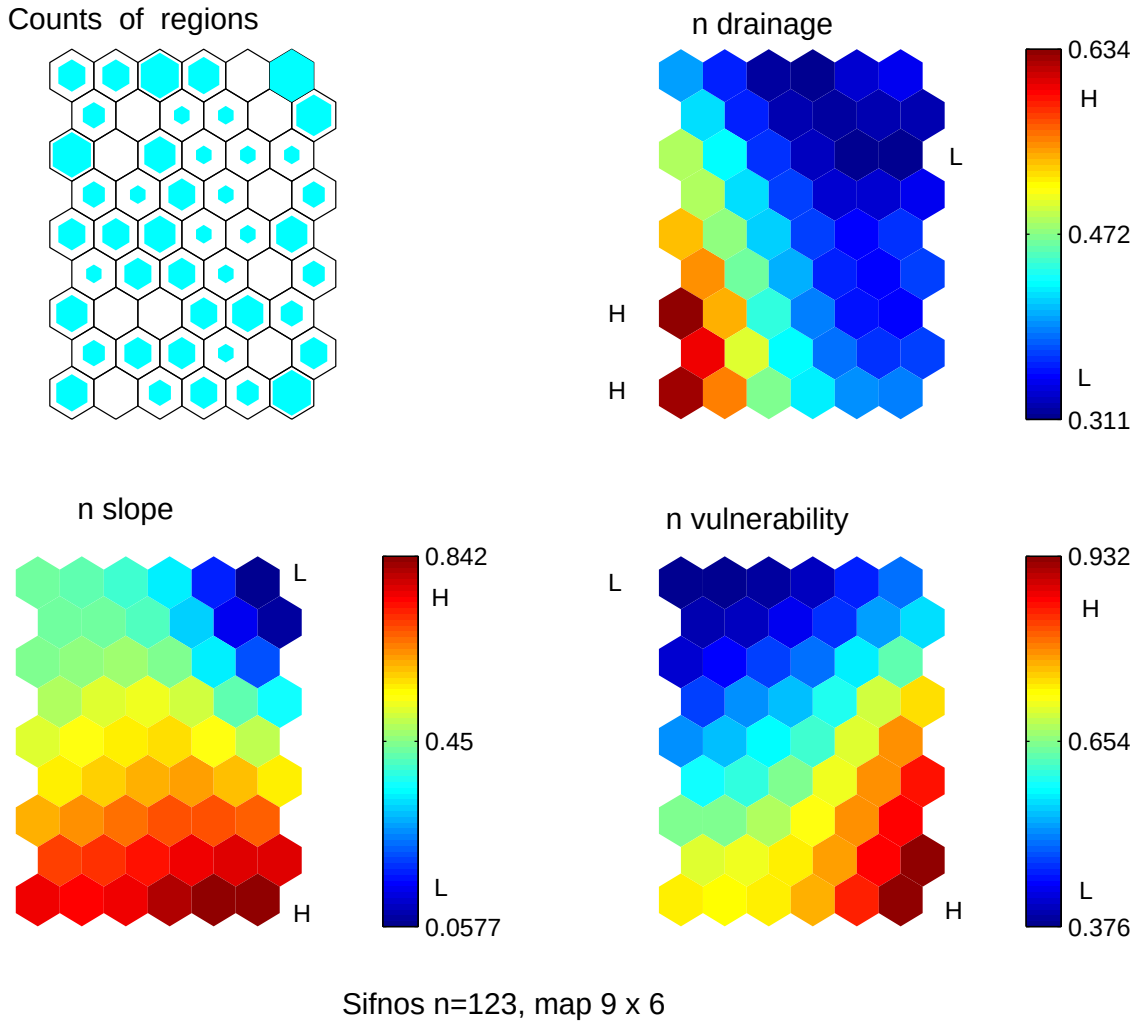

Figure 2: *Basic SOM with hits (top left) and 3 component planes. The hits show – by size of the painted area – how many data vectors are represented by subsequent hexagons. The three component planes show – by hue – the values of the three variables met in the codebook vectors connected with the subsequent hexagons.*

2.4 What kind of information may be displayed in a SOM

There is a variety of ways by which information about the analyzed data may be displayed in a SOM. We have used a few of them implemented in the package SOM Toolbox for Matlab 5 [5]. This package considers the constructed SOM as an object with various object properties.

The SOM object contains in first place the codebook vectors which constitute a new, reduced data table of size $m \times p$. The rows of that table represent the codebook vectors located in the data space and may have labels – which are set automatically by the program or manually by the user. We will use the labeling in Section 3 to indicate classification of the data into erosion risk classes.

2.4.1 Information on individual data points or their frequencies in the Voronoi regions

We know that each hexagon in the map represents a Voronoi region in the data space R^p. How many data vectors belong to one such Voronoi region? This may be exhibited in a map at least in three ways:

- writing directly on the top of the hexagon *the labels* of the data points (this is possible for relatively small data sets)
- writing on the top of the hexagons *the summary number* of points belonging to the corresponding Voronoi regions,
- drawing inside each hexagon another painted one with *surface proportional* to the number of points belonging to the corresponding Voronoi region.

We have chosen the 3rd alternative. The table of frequency counts is shown in Section 3, Table 1, under the heading: TOTAL SUM. The frequency counts appear in blocks of size $m1 \times m2$. When referred to map units, the frequency counts are called *hits*.

The maximum number of frequencies was $f_{<4,6>} = 8$, the minimum was 0 (for hexagons no. $< 1,5 >$, $< 2,2 >$, $< 2,5 >$, $< 3,2 >$, $< 4,5 >$, $< 6,5 >$, $< 7,2 >$, $< 7,3 >$, $< 8,5 >$, $< 9,2 >$). The number in the brackets $<,>$ indicate the row and column position of the hexagon (in the map). An empty hexagon means that the respective Voronoi region contains no points from the actually analyzed data set. An empty hexagon implies also that the data points are distributed unevenly over the data space.

2.4.2 Presentation of the codebook vectors

The codebook table is accessible as a normal data table. It is interesting to display various characteristics of the codebook vectors on top of the constructed map.

In Figure 2 we show the components of the codebook vectors component wise, i.e. each variable is visualized in a separate plot. Subsequent maps in Figure 2 present the distributions of the three analyzed variables. The magnitude of values of the variables is displayed using hue of the colormap ' jet ' [5]. A corresponding colorbar is shown at the right side of each map. When printed, the maps appear in shades of gray, therefore the poles of the hue are additionally annotated with L – for *lowest*, and H – for *highest* values.

One may note in Figure 2 the specific distribution of the three variables over the map. The growing of values of the variables occurs in each map in other direction, which means that the variables are practically independent.

It is also possible to gather together the information on all the p analyzed variables and exhibit it by a minuscule bar plot or line plot located on the top of each hexagon. Another interesting possibility is to draw a plot called 'umat' which shows, how far away are the codebook points - each from the other - when considering their true location in the data space. Because of lack of space we do not show these plots.

3 Classification of data vectors.

Kohonen's maps may be also used for visualizing the subdivision of the data into some classes or categories which were obtained by independent pattern recognition methods.

For our data 5 classes of erosion risk were considered: L - very low, l - low, m - medium, h - high, H - very high. The classification was done a) by experts using Boolean rules, b) by neural network using logistic regression and training data shown in Figure 1. The experts have considered all 5 classes. The neural network was trained to recognize only four classes of erosion risk – the very low and low classes were merged together and denoted jointly by l, while the m, h, and H classes remained the same.

We wanted to see how are the erosion risk classes distributed over the map constructed from the entire set of data. To achieve this we had to take each data class separately, find for it the corresponding hits, and next mark the number of hits in the appropriate hexagons. The blocks of hits – when considering the classification by both methods – are shown in Table 1 below.

Table 1: *Hits of classes of erosion risk established by experts (a), and neural network (b). Differences between a) and b) are marked by '*'. Part (c) shows hits of the entire data set without subdivision into classes.*

```
  (a) Hits of the L, l, m, h, H classes -- obtained by experts opinion
        0 0 0 0 0 6* 3 3 6 4 0 2* 0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0
        0 0 0 0 0 0  2 0 1 1 0 3  0 0 0 0 0 2  0 0 0 0 0 0  0 0 0 0 0 0
        0 0 0 0 0 0  6 0 4 1 0 0  0 0 0 0 1 1  0 0 0 0 0 0  0 0 0 0 0 0
        0 0 0 0 0 0  1*0*0 0 0 0  1*1*3 1 0 0  0 0 0 0 0 2  0 0 0 0 0 0
        0 0 0 0 0 0  1 0*0 0 0 0  2 3*4 1 1 0  0 0 0 0 0 4  0 0 0 0 0 0
        0 0 0 0 0 0  0 0 0 0 0 0  1 3 3 0 0 0  0 0 0 1 0 0  0 0 0 0 0 2
        0 0 0 0 0 0  0 0 0 0 0 0  3 0 0 0 0 0  1 0 0 3 4 0* 0 0 0 0 0 2*
        0 0 0 0 0 0  0 0 0 0 0 0  0 1 0 0 0 0  2 2 3 1 0 0  0 0 0 0 0 3
        0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0  4 0 2 3 2 0  0 0 0 0 0 6
 (c)  TOTAL SUM    (b) Hits of the (L+l), m, h, H classes -- by neural networks
   3 3 6 4 0 8       3 3 6 4 0 8* 0 0 0 0 0 0  0 0 0 0 0 0  0 0 0 0 0 0
   2 0 1 1 0 5       2 0 1 1 0 3  0 0 0 0 0 2  0 0 0 0 0 0  0 0 0 0 0 0
   6 0 4 1 1 1       6 0 4 1 0 0  0 0 0 0 1 1  0 0 0 0 0 0  0 0 0 0 0 0
   2 1 3 1 0 2       2*1*0 0 0 0  0*0*3 1 0 0  0 0 0 0 0 2  0 0 0 0 0 0
   3 3 4 1 1 4       1 1*0 0 0 0  2 2*4 1 1 0  0 0 0 0 0 4  0 0 0 0 0 0
   1 3 3 1 0 2       0 0 0 0 0 0  1 3 3 0 0 0  0 0 0 1 0 0  0 0 0 0 0 2
   4 0 0 3 4 2       0 0 0 0 0 0  3 0 0 0 0 0  1 0 0 3 4 1* 0 0 0 0 0 1*
   2 3 3 1 0 3       0 0 0 0 0 0  0 1 0 0 0 0  2 2 3 1 0 0  0 0 0 0 0 3
   4 0 2 3 2 6       0 0 0 0 0 0  0 0 0 0 0 0  4 0 2 3 2 0  0 0 0 0 0 6
```

On the basis of the blocks of hits given in Table 1 we constructed the maps exhibited in Figure 3 and Figure 4.

Figure 3 shows the classification done by experts. In particular: Black paint (in color print: dark red) in the bottom right corner of the right map indicates a very high risk (H). Enforced gray (in color print: dark green) at the upper right corner of the map denotes a region with very low erosion risk (L). Hexagons representing high (h) and low (l) erosion risk units are in the south and north of the map. The medium risk units are between them.

The maps presenting the classification done by neural network exhibits the same pattern. The increase of erosion risk goes from the south to the north of the maps.

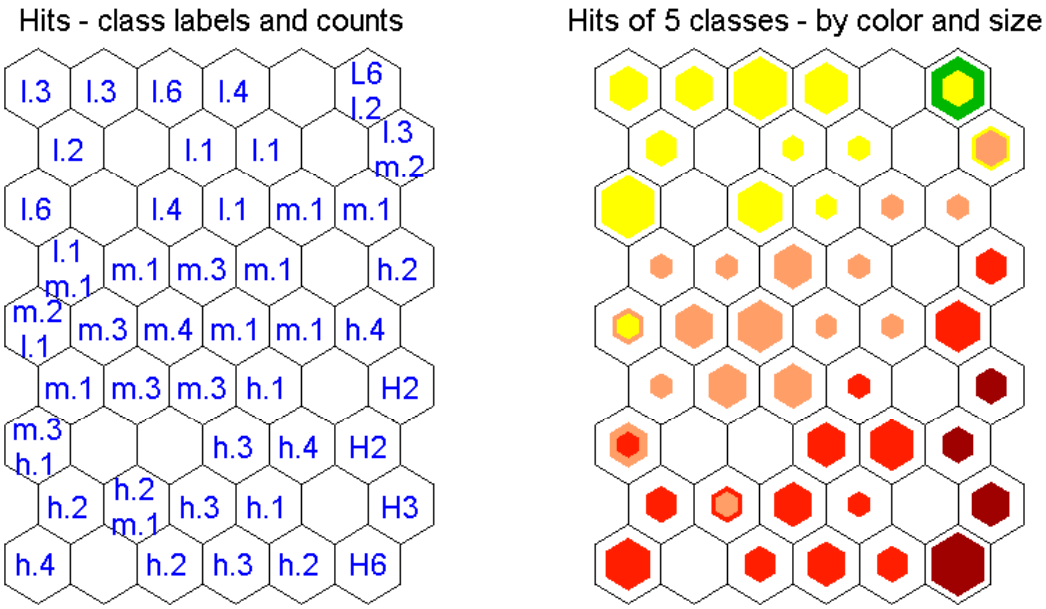

Figure 3. *Hits into 5 erosion risk classes (L, l, m, h, H) obtained by experts. LEFT: class labels and counts. RIGHT: hits of erosion risk classes marked by color and size. The increase of the gray hue (in color: yellow, green, red, dark red) indicates rising of erosion risk - except the top right hexagon, where dark gray (dark green) denotes a region with very low erosion risk.*

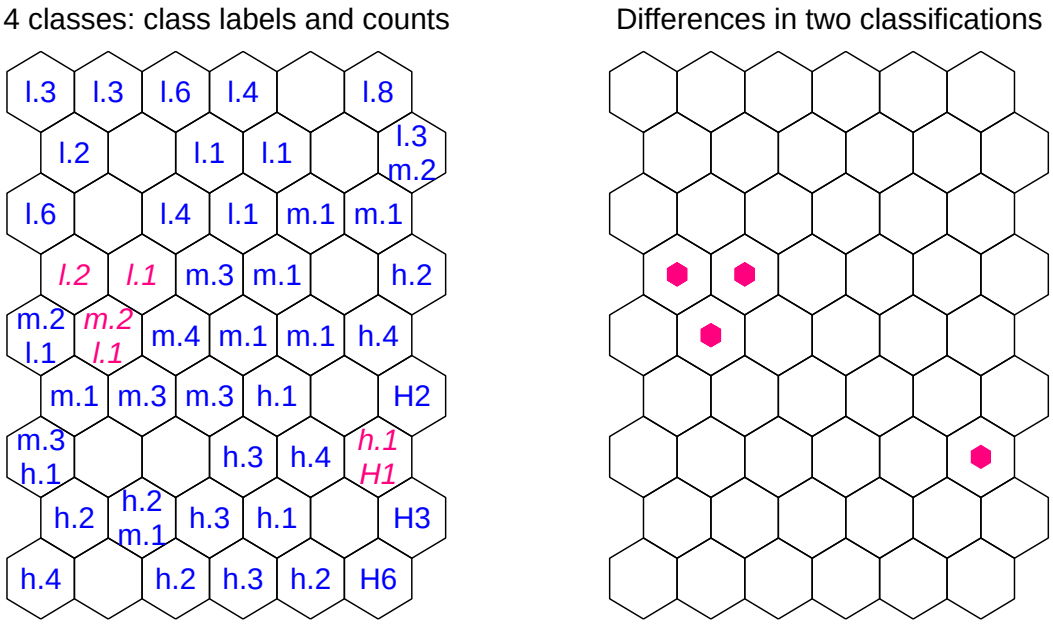

Figure 4. *LEFT: Subdivision of the data into 4 erosion risk classes (l=L+l, l, m, h, H) obtained by neural network. Only class labels and counts are shown. RIGHT: Placement of data vectors assigned by experts and neural network to different classes .*

The two maps differ only in 4 data vectors (basins). The hits for those basins are marked in the Table 1 by an asterisk. The map units containing the differing items are shown in Figure 4, right plot. The differently classified items are:

	classification			classification	
id (basin)	by experts	by neur. network	id (basin)	by experts	by neur. network
13 (9)	H	h	80 (71)	m	l
78 (69)	m	l	88 (79)	m	l

One may state that the four differing items were placed by neural networks in neighbor classes.

4 Closing remarks

We have demonstrated that self-organizing Kohonen's maps (SOMs) – obtained by applying an unsupervised self-learning technique – may be useful in representing data points from a multivariate data space. The constructed SOM can exhibit information not only on the neighborhood of the data points and their topology, but also on the (multivariate) density distribution of the analyzed data cloud.

An appropriately painted map may also serve as a tool for comparing graphically two multivariate distributions or results of classification. For the considered Sifnos data we have compared erosion risk calculated by two methods with a substantially different philosophy. The erosion risk classes estimated by the two methods differed only in four data vectors. When viewing the location of these four data vectors in the maps, it was stated, that the 4 items were located at the borders of neighbor risk classes.

During the construction of a SOM we perform a quantization of the data space into adjacent regions. All data points located in one such region are represented by one (central) point called codebook point or codebook vector. This provides a data reduction technique. Thanks to that technique the method of self-organizing maps can be easily applied to very large sets of data.

References

[1] Gournelos Th., Evelpidou N. and Vassilopoulos A., 2002, Developing an erosion risk map using soft computing, *Natural Hazards*, Submitted.

[2] Gournelos Th., 1980, *Contribution à l'étude geologique des Cyclades, L'ile de Siphnos*, Thèse de 3-éme cycle, Université de Paris VI, p. 182.

[3] Kohonen T., 1995, *Self-organizing Maps*, Springer, Berlin - Heidelberg.

[4] MapInfo Professional, 1999, MapInfo Corporation, Troy, New York.

[5] Vesanto J., Himberg J., Alhoniemi E. and Parhankangas J., 2000, *SOM Toolbox for Matlab 5*, Helsinki University of Technology, Finland, Libella Oy, Espoo 2000, 1-54. http://www.cis.hut.fi/projects/somtoolbox/

[6] Vesanto J., 1999, SOM-based data visualization methods. Intelligent Data Analysis, 3 (2), pp. 111-126.

Authors:

Prof. Anna Bartkowiak, University of Wrocław, Institute of Computer Science, Przesmyckiego 20, 51-151 Wrocław, Poland. e-mail: aba@ii.uni.wroc.pl

Dr. Niki Evelpidou and Dr Andreas Vassilopoulos, University of Athens, Geology Department, Remote Sensing Laboratory, Panepistimioupolis, Zografou, 157-84 Athens, Greece, e-mail: evelpidou@geol.uoa.gr