INFORMATICA, 2002, Vol. 13, No. 3, 275–286

Comparative Analysis of the Graphical Result Presentation in the SOM Software

Gintautas DZEMYDA, Olga KURASOVA
Institute of Mathematics and Informatics
Akademijos 4, LT-2021 Vilnius, Lithuania
e-mail: dzemyda@ktl.mii.lt

Received: May 2002

Abstract. In the paper, we analyze the software that realizes the self-organizing maps: SOM-PAK, SOM-TOOLBOX, Viscovery SOMine, Nenet, and two academic systems. Most of the software may be found in the Internet. These are freeware, shareware or demo. The self-organizing maps assist in data clustering and analyzing data similarities. The software differs one from another in the realization and visualization capabilities. The data on coastal dunes and their vegetation in Finland are used for the experimental comparison of the graphical result presentation of the software. Similarities of the systems and their differences, advantages and imperfections are exposed.

Key words: neural networks, SOM, mapping, visualization, correlation matrix, environmental parameters.

1. Introduction

The theory of artificial neural networks is developing very rapidly. The self-organizing map (SOM) proposed by Kohonen (2001) is a class of neural networks that are trained in an unsupervised manner using competitive learning. It is a well-known method for mapping a high dimensional space onto a low dimensional one. We consider here a mapping onto a two-dimensional grid of neurons. The method allows putting complex data into order, based on its similarity, and shows a map from which the features of the data can be identified and evaluated. Therefore, at the present time a lot of software that realizes the SOM is being created. Most of the systems can be found in the Internet. They are freeware, shareware or demo. It is necessary to analyze and compare the variety of these systems seeking to expose their advantages and imperfections. In this paper, the following systems have been analyzed:

N1. System **SOM-PAK** (Kohonen *et al.*, 1996). It has been developed at the Neural Networks Research Centre (NNRC), Laboratory of Computer and Information Science of Helsinki University of Technology.

N2. System **SOM-TOOLBOX** (Alhoniemi *et al.*, 2000). Like **N1** it has been developed at the NNRC.

N3. System **Viscovery SOMine** (2002).

N4. System **Nenet** (Hassinen, 1999).

N5. Academic system developed at the Groningen University in the Netherlands (Kleiweg, 1996).

N6. Academic system of correlation matrix analysis (Dzemyda, 2001a, b) that may be applied to the data of general nature as well (Dzemyda and Tiešis, 2001).

A certain analysis of several systems (SOM-PAK, SOM-TOOLBOX, Viscovery SOMine and Nenet) has been made by Deboeck (1999). The first difference of our study is on the application of real data (of ecological nature) for analysis. The second one is that we extend the analysis by two academic realizations. The third difference is that Deboeck (1999) put an attention on the system technical characteristics only. Our analysis tries to compare the graphical result presentation by different systems and the level of understability of the presentation.

2. The SOM

We present here some general details on the SOM. We consider here a mapping onto a two-dimensional grid of neurons. Let $Y_1, \ldots, Y_s \in R^n$ be a set of n-dimensional vectors for mapping. Usually, the neurons are connected to each other via a rectangular or hexagonal topology. Let us consider an example of the rectangular case because all ideas can be easily extended to the hexagonal one. The rectangular SOM is a two-dimensional array of neurons $M = \{m_{ij},\ i = \overline{1, k_x},\ j = \overline{1, k_y}\}$. Here k_x is the number of rows, and k_y is the number of columns. The total number of neurons is equal to $k_x \times k_y$. All neurons adjacent to a given neuron can be defined as its neighbours of a first order, then the neurons adjacent to a first-order neighbour, excluding those already considered, as neighbours of a second order, etc. The dimension of the vectors, that will be presented as inputs to train the network, is n. Each component of the input vector is connected to every individual neuron. Thus, there is a connection between the neuron of the network and every component of the input vector. The weights of these connections form an n-dimensional synaptic weight vector (the codebook vector, also called reference, model, or parameter vector, – see Kohonen (2001)). Thus, any neuron is entirely defined by its location on the grid (number of row i and column j) and by the codebook vector, i.e., we can consider a neuron as an n-dimensional vector $m_{ij} = (m_{ij}^1,\ m_{ij}^2, \ldots, m_{ij}^n) \in R^n$. In this way, each vector (neuron) m_{ij} represents a part of R^n.

The map is trained in an unsupervised manner using competitive learning. The learning starts from the vectors m_{ij} initialized randomly (other ways to initialize the vectors m_{ij} are possible, too). At each learning step, an input vector Y is drawn from the training set $\{Y_1, \ldots, Y_s\}$ and passed to the neural network. The Euclidean distance from this input vector to each vector m_{ij} is calculated and the vector (neuron) $m_c \in \{m_{ij},\ i = \overline{1, k_x},\ j = \overline{1, k_y}\}$ with the minimal Euclidean distance to Y is designated as a winner. Denote the row, where m_c is located, by i_c , and the column by j_c, i.e., c is a combination of two numbers: i_c and j_c. The components of the vector m_{ij} are adapted according to the rule $m_{ij} \leftarrow m_{ij} + h_{ij}^c(Y - m_{ij})$, where h_{ij}^c is the learning rate, which is maximal for the winning neuron, and decreases with the neighbourhood order and learning steps.

After a large number of learning steps, the network has been organized and n-dimensional input vectors $Y_1, \ldots, Y_s$ have been mapped – each input vector is related to the nearest neuron, i.e., the vectors are distributed among the elements of the map during training. Some elements of the map may remain unrelated with any vector from $\{Y_1, \ldots, Y_s\}$, but there may occur elements related with some vectors.

Using the SOM-based approach above we can draw a table with cells corresponding to the neurons. The cells corresponding to the neurons-winners are filled with the numbers of vectors $Y_1, \ldots, Y_s$. Some cells may remain empty. One can decide visually on the distribution of vectors $Y_1, \ldots, Y_s$ in the n-dimensional space R^n in accordance to their distribution among the cells of the table. The essential property of such visualization is clustering of the objects in accordance to their similarities.

3. Some Details of the Systems

There are some programs in system **N1 (SOM-PAK)**: **mapinit**, **randinit**, **lininit** initiate the map; **vsom** trains the map; **vfind** initiates the map and trains it; **vcal** labels the map units according to the samples in the input data file; **visual, planes**, **umat**, **sammon** visualize the map. All the programs run in *MS-DOS* or *UNIX*.

A set of functions created using *Matlab 5* is in system **N2 (SOM-TOOLBOX)**. *Matlab 5* (or higher) is necessary for this system. There are many functions in **SOM-TOOLBOX**. The basic functions are: ***som_read_data*** reads data from the ASCII file in **SOM_PAK** format; ***som_make*** creates, initializes and trains the SOM; ***som_autolabel*** automatically labels the map; ***som_show*** visualizes the map; ***som_show_add*** shows labels on the ***som_show*** visualization. Four kinds of SOM planes can be shown (see (Alhoniemi *et al.*, 2000) for details). The system has a graphical user interface for initialization, training, and visualization of the maps. The following functions may be used here: ***som_gui*** and ***som_show_gui***.

System **N3** (**Viscovery SOMine**) is a program that presents a friendly graphical interface to the user. It runs on *MS Windows*. This program can create a new map or edit the earlier created one. The user can choose components of reference vectors to train the SOM. **Viscovery SOMine** performs the clustering of data: lines may separate the clusters. In the demo version, a created map cannot be saved, but there is a possibility to export the map to a graphic file.

System **N4 (Nenet)** has its *MS Windows* interface, too. Creation of a new map consists of three parts: initialization of the map (*init*), its training (*train*) and testing (*test*). Before the initialization, it is necessary to point out SOM dimensions (the dimension is restricted to 10 in the demo version), the neighbourhood function, topology type, initialization type, random seed, and the initialization and training data file. Before training and testing, it is necessary to point out the file of training and testing data and some parameters. The map can be saved for later editing.

System **N5** includes the set of files: **koh.c** is the program of creation of Kohonen's maps, it should be compiled using *Borland C* or *Borland C++*; **koh.exe** is the compiled

koh.c program, it runs in *MS-DOS*; **kohview.cpp** is the program of visualization of the maps created using **koh.exe**, it runs in *MS-DOS*, too, and should be compiled using *Borland C 3.1*; **kohview.exe** is the compiled **kohview.cpp** program. The program **koh.exe** produces a lot of files: **file_name.ps, file_name.map, file_name.top, file_name.log, file_name.spn** (see (Kleiweg, 1996) for details). The program **kohview.exe** visualizes Kohonen's map and may be used in case the user has no tool to view *PostScript* files (example *GSView*).

System **N6** is a set of two programs written in *Fortran*. Sammon's algorithm and SOM are integrated in system **N6**. Some additional details on the realization are given in the papers by Dzemyda (2001a, b) devoted to the visual analysis and clustering of parameters by their correlation matrix. The table, filled out in accordance with the distribution of vectors $Y_1, \ldots, Y_s$ among the cells of the SOM, does not answer the question, how much the objects of the neighbouring cells are close in the n-dimensional space. In the case of the correlation matrix analysis, Dzemyda (2001a) has experimentally proved a general idea of Kaski (1997) that an especially useful combination first is to reduce the amount of data by SOM, and then to analyze the codebook vectors corresponding to non-empty cells by using Sammon's mapping, i.e., to visualize the relative distances between these codebook vectors of the SOM. Kohonen (2001) also notes that Sammon's projection is a useful tool in the preliminary data analysis as well as for monitoring the codebook vectors). In this case, the neural network performs some sorting (clustering) of data, and Sammon's algorithm presents the results visually to gain an additional insight.

The input data files of each system must be ordered in line with the requirements of that system. The data must be in text files (system **N3** allows *MS Excel* file, too). The dimension of vectors must be placed in the first line of these files (**N3** does not need this number). In system **N2**, the dimension may be set either in the input data file or in the argument of the data reading function ***som_read_data(filename, dimension)***. In system **N5**, the components of vectors must be written in one line or column, in the remaining systems each vector must be in a separate row. The names of vectors must appear before their components in system **N5**; the last column of the data files contains the names of vectors in systems **N1**, **N2**, **N4**; the names of vectors may be written in any column in system **N3**. In system **N3**, the neural network may be trained by a part of the selected components.

4. Results of Analysis

The data on coastal dunes and their vegetation in Finland (Hellemaa, 1998) have been analyzed. The following parameters $x_1 - x_{16}$ characterize the dunes:

- x_1 is the distance from the water line;
- x_2 is the height above the sea level;
- x_3 is the soil PH;
- x_4, x_5, x_6 , and x_7 are the contents of calcium (CA), phosphorous (P), potassium (K), manganese (Mn);

- x_8 and x_9 are the mean diameter and sorting of sand;
- x_{10} is the northernness in the Finnish coordinate system;
- x_{11} is the rate of land uplift;
- x_{12} is the sea level fluctuation;
- x_{13} is the soil moisture content;
- x_{14} is the slope tangent;
- x_{15} is the proportion of bare sand surface;
- x_{16} is the tree cover.

The correlation matrix $R = \{r_{x_i x_j},\ i, j = \overline{1, s}\}$ of these sixteen parameters ($s = 16$) is given in Table 1. Using the method developed by Dzemyda (2001), sixteen vectors $Y_1, \ldots, Y_s$ of unit length have been computed ($\|Y_l\| = 1,\ l = \overline{1, s}$), the dimension n of these vectors is the same as s, i.e., $n = s$. These vectors $Y_1, \ldots, Y_s$ correspond to the parameters $x_1, \ldots, x_{16}$. They are given in Table 2. Namely, they are used in the compared systems as data vectors for training and analyzing. The vectors $Y_l = (y_{l1}, \ldots, y_{ln}),\ l = \overline{1, s}$ are defined as follows: $y_{lk} = \sqrt{\lambda_k}\alpha_{lk}, k = \overline{1, n}$, where λ_k is the k-th eigenvalue of the matrix $K = \{r^2_{x_i x_j},\ i, j = \overline{1, s}\}, (\alpha_{1k}, \ldots, \alpha_{nk})$ is a normalized eigenvector corresponding to the eigenvalue λ_k.

The experiments with system **N4** have been performed using a smaller dimension of vectors $Y_1, \ldots, Y_s$ ($s = 16,\ n = 6$), because the examined demo version has limited capabilities (the maximal dimension of the vectors is restricted by 6). However, we can use such simplification, because the vectors $Y_1, \ldots, Y_s$ obtained in the above manner have a peculiarity that their components with higher order numbers have smaller absolute values and are less significant (see Table 2 to compare columns 1 and 16 visually). The reason is that the significance of components is characterized by the values of $\lambda_k,\ k = \overline{1, n}$, and the following equation holds: $\lambda_k \geqslant \lambda_l$ if $k < l$.

A graphic display method, called the U-matrix (Unified distance matrix), to illustrate the clustering of codebook vectors in the SOM has been developed by Ultsch and Siemon (1989), as well as Kraiijveld *et al.* (1992). They have proposed a method in which average distances between the neighbouring codebook vectors are represented by shades in a gray scale (or, eventually, pseudo-colour scales might be used). If the average distance of neighbouring neurons is small, a light shade is used; dark shades represent large distances: high values of the U-matrix indicate a cluster border; uniform areas of low values indicate the clusters themselves. A "cluster landscape" formed over the SOM then visualizes the classification. The interpretation is left to the reader: the clusters are indicated by light shades and border with darker shades, respectively (Kohonen, 2001, 2002). Systems **N1**–**N5** use various realizations of the idea of the U-matrix. The graphical results of systems **N1**–**N6** are presented in Figures 1–6 and Table 3. The rectangular SOM of dimensions 4×4 has been used in experiments with all the systems except **N3** which allows the hexagonal topology, only. Systems **N1** and **N2** provide additional rows and columns separating neurons, and systems **N4** and **N5** colourize the borders between any pair of the neighbouring neurons. System **N3** separates clusters by lines automatically. It is able to detect the number of clusters. Most of the results by other systems indicate four clusters in the data set. Therefore, in **N3** we used both the automatic detection of the

Table 1

Correlation matrix $R = \{r_{x_i x_j}, i, j = \overline{1, 16}\}$ of environmental parameters that describe the development of coastal dunes and their vegetation in Finland

$i \backslash j$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1.00	0.72	−0.60	−0.23	−0.02	−0.33	−0.38	−0.12	0.21	0.17	0.20	0.07	−0.02	0.02	−0.20	0.61
2	0.72	1.00	−0.36	−0.17	−0.09	−0.20	−0.22	−0.31	0.23	0.12	0.17	0.07	0.05	0.11	0.16	0.52
3	−0.60	−0.36	1.00	0.41	0.29	0.60	0.70	0.08	−0.36	−0.23	−0.22	−0.20	−0.26	−0.08	0.22	−0.39
4	−0.23	−0.17	0.41	1.00	0.20	0.79	0.70	−0.25	0.10	−0.42	−0.46	−0.26	−0.29	−0.10	−0.02	−0.07
5	−0.02	−0.09	0.29	0.20	1.00	0.17	0.47	0.35	−0.40	−0.31	−0.34	−0.29	−0.13	−0.36	0.01	0.02
6	−0.33	−0.20	0.60	0.79	0.17	1.00	0.69	−0.13	−0.02	−0.24	−0.28	−0.08	−0.19	−0.06	−0.02	−0.04
7	−0.38	−0.22	0.70	0.70	0.47	0.69	1.00	0.01	−0.20	−0.50	−0.52	−0.39	−0.47	−0.14	0.13	−0.06
8	−0.12	−0.31	0.08	−0.25	0.35	−0.13	0.01	1.00	−0.60	0.12	0.07	0.07	−0.05	−0.06	−0.15	−0.19
9	0.21	0.23	−0.36	0.10	−0.40	−0.02	−0.20	−0.60	1.00	0.27	0.30	0.25	0.30	0.02	−0.13	0.30
10	0.17	0.12	−0.23	−0.42	−0.31	−0.24	−0.50	0.12	0.27	1.00	0.96	0.91	0.69	0.18	−0.24	0.14
11	0.20	0.17	−0.22	−0.46	−0.34	−0.28	−0.52	0.07	0.30	0.96	1.00	0.76	0.64	0.21	−0.16	0.16
12	0.07	0.07	−0.20	−0.26	−0.29	−0.08	−0.39	0.07	0.25	0.91	0.76	1.00	0.67	0.15	−0.31	0.11
13	−0.02	0.05	−0.26	−0.29	−0.13	−0.19	−0.47	−0.05	0.30	0.69	0.64	0.67	1.00	−0.05	−0.06	−0.01
14	0.02	0.11	−0.08	−0.10	−0.36	−0.06	−0.14	−0.06	0.02	0.18	0.21	0.15	−0.05	1.00	−0.13	−0.02
15	−0.20	0.16	0.22	−0.02	0.01	−0.02	0.13	−0.15	−0.13	−0.24	−0.16	−0.31	−0.06	−0.13	1.00	−0.19
16	0.61	0.52	−0.39	−0.07	0.02	−0.04	−0.06	−0.19	0.30	0.14	0.16	0.11	−0.01	−0.02	−0.19	1.00

Table 2

Vectors corresponding to the parameters

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Y_1, x_1	0.208	0.541	0.573	−0.252	0.043	−0.052	−0.020	−0.025	−0.153	−0.089	−0.009	0.055	−0.458	−0.123	0.083	0.000
Y_2, x_2	0.135	0.402	0.599	−0.152	−0.003	−0.107	0.255	−0.210	−0.397	0.104	−0.168	−0.030	0.337	0.067	−0.046	−0.001
Y_3, x_3	0.364	0.593	0.010	−0.024	0.076	0.082	−0.477	0.301	−0.070	−0.238	0.177	0.064	0.103	0.126	−0.247	0.001
Y_4, x_4	0.466	0.422	−0.477	0.001	−0.152	−0.009	0.400	−0.126	0.119	0.088	−0.119	−0.056	−0.103	−0.164	−0.319	0.000
Y_5, x_5	0.215	0.069	−0.005	0.493	0.515	−0.104	−0.285	−0.536	0.064	0.173	−0.028	0.130	−0.060	0.058	−0.050	−0.002
Y_6, x_6	0.391	0.571	−0.478	−0.040	−0.113	0.027	0.201	0.015	0.050	−0.005	−0.066	0.344	0.023	0.231	0.240	0.000
Y_7, x_7	0.601	0.447	−0.342	0.020	0.095	−0.007	−0.218	−0.015	−0.063	0.080	0.084	−0.350	0.131	−0.226	0.237	0.000
Y_8, x_8	0.076	0.102	0.199	0.719	−0.291	0.030	0.283	−0.004	−0.143	0.002	0.489	−0.020	−0.034	0.022	0.018	0.002
Y_9, x_9	0.176	0.084	0.259	0.706	−0.226	0.009	−0.111	0.257	0.095	−0.111	−0.490	−0.012	0.008	−0.061	0.037	−0.002
Y_{10}, x_{10}	0.858	−0.438	0.106	−0.080	−0.015	−0.016	0.063	−0.068	0.031	−0.188	0.025	−0.017	0.000	0.044	0.004	−0.082
Y_{11}, x_{11}	0.803	−0.355	0.072	−0.056	0.004	−0.067	0.094	−0.103	0.056	−0.208	−0.048	−0.248	−0.091	0.274	0.004	0.055
Y_{12}, x_{12}	0.716	−0.447	0.144	−0.074	−0.007	0.063	−0.013	−0.012	−0.008	−0.099	0.078	0.346	0.148	−0.312	0.009	0.037
Y_{13}, x_{13}	0.573	−0.284	0.052	−0.048	−0.112	−0.075	−0.149	0.322	−0.148	0.632	−0.003	0.034	−0.099	0.091	−0.039	0.000
Y_{14}, x_{14}	0.059	−0.006	0.009	0.144	0.782	−0.039	0.418	0.432	−0.023	−0.029	0.008	−0.012	0.005	−0.017	0.008	0.000
Y_{15}, x_{15}	0.085	0.004	0.114	−0.006	0.087	0.975	0.040	−0.075	−0.019	0.090	−0.050	−0.042	−0.022	0.033	0.004	0.000
Y_{16}, x_{16}	0.108	0.319	0.561	−0.132	−0.014	−0.034	0.062	0.028	0.701	0.167	0.123	−0.029	0.113	0.014	0.021	0.000

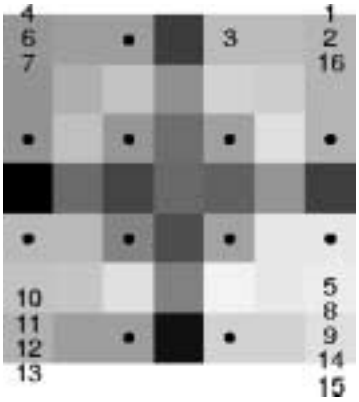

Fig. 1. Distribution of the parameters that characterze the dunes, 4×4 SOM, system **N1** (SOM-PAK).

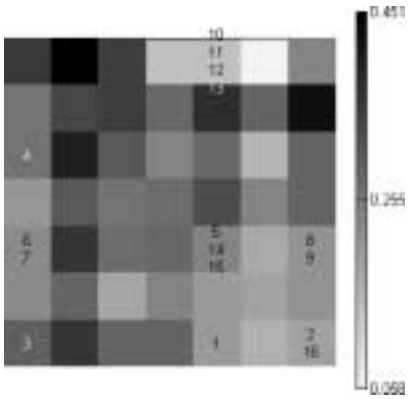

Fig. 2. Distribution of the parameters, 4×4 SOM, system **N2** (SOM-TOOLBOX).

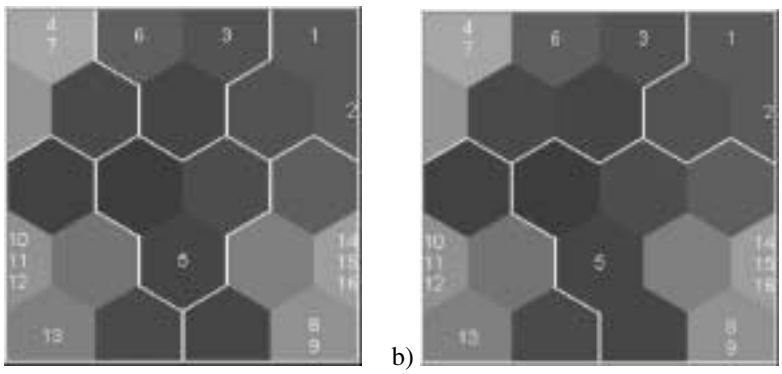

Fig.3. Distribution of the parameters, system **N3** (Viscovery SOMine):
a) automatically detected six clusters; b) by force into four clusters.

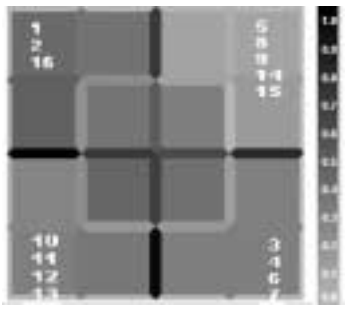

Fig. 4. Distribution of the parameters, 4×4 SOM, system **N4** (Nenet).

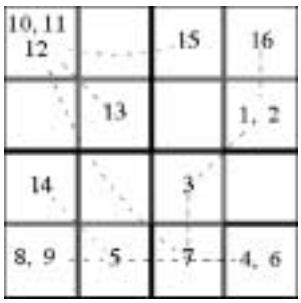

Fig. 5. Distribution of the parameters, 4×4 SOM, system **N5**.

Table 3

Distribution of the parameters, 4×4 SOM, system **N6**

4, 6		5	8, 9
7		15	14
3			
1, 2, 16		13	10, 11, 12

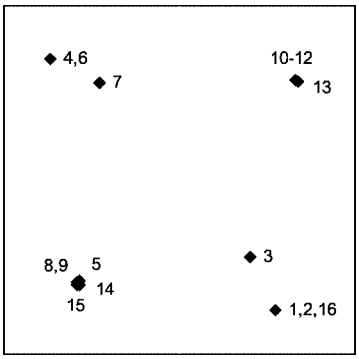

Fig. 6. Combination of SOM and Sammon's mapping.

Table 4

Clusters in the ecological data by different systems

System	Clusters
N1	Four clusters: $\{x_1, x_2, x_3, x_{16}\}, \{x_4, x_6, x_7\}, \{x_5, x_8, x_9, x_{14}, x_{15}\}, \{x_{10}, x_{11}, x_{12}, x_{13}\}$. The parameter x_3 has a tendency to form a separate cluster.
N2	The first cluster is $\{x_{10}, x_{11}, x_{12}, x_{13}\}$. We can suppose that there a second large cluster $\{x_1, x_2, x_5, x_8, x_9, x_{14}, x_{15}, x_{16}\}$. However, it may be divided into two subclusters: $\{x_1, x_2, x_{16}\}$, $\{x_5, x_8, x_9, x_{14}, x_{15}\}$. The third large cluster is $\{x_3, x_4, x_6, x_7\}$, however the parameters in this cluster are not so strong-related like in the previous ones.
N3	Four clusters: $\{x_1, x_2\}, \{x_3, x_4, x_6, x_7\}, \{x_{10}, x_{11}, x_{12}, x_{13}\}, \{x_5, x_8, x_9, x_{14}, x_{15}, x_{16}\}$. Six clusters: $\{x_1, x_2\}$,$\{x_3, x_6\}$,$\{x_4, x_7\}$,$\{x_{10}, x_{11}, x_{12}, x_{13}\}$,$\{x_5\}$,$\{x_8, x_9, x_{14}, x_{15}, x_{16}\}$.
N4	Four clusters: $\{x_3, x_4, x_6, x_7\}, \{x_5, x_8, x_9, x_{14}, x_{15}\}, \{x_{10}, x_{11}, x_{12}, x_{13}\}, \{x_1, x_2, x_{16}\}$.
N5	Four clusters: $\{x_3, x_4, x_6, x_7\}, \{x_5, x_8, x_9, x_{14}\}, \{x_{10}, x_{11}, x_{12}, x_{13}, x_{15}\}, \{x_1, x_2, x_{16}\}$.
N6	Four clusters: $\{x_4, x_6, x_7\}, \{x_5, x_8, x_9, x_{14}, x_{15}\}, \{x_{10}, x_{11}, x_{12}, x_{13}\}, \{x_1, x_2, x_3, x_{16}\}$. The parameter x_3 has a tendency to form a separate cluster.

number of clusters and forced the partition into four clusters. There is an order in the set of vectors that can be displayed in a minimal spanning tree, obtained by linking all the vectors together, using the smallest possible square difference between the linked vectors (Kohonen, 2001). System **N5** automatically provides the so-called minimal spanning tree that describes the similarity relations of the items of Table 2.

The simplest map is obtained by **N6**. In fact, we have here a quadratic table. The cells of this table are filled with the order numbers of vectors $Y_1, \ldots, Y_{16}$ (and the respective parameters $x_1, \ldots, x_{16}$, see Table 2). It is impossible to determine from the table, which vectors are closer and which are remote. Only some preliminary conclusions may be drawn. An application of some other visualization means is necessary here. Therefore, **N6** integrates Sammon's mapping to gain an additional insight. Fig. 6 has been drawn by means of *MS Excel*. The scales of data are not presented in Fig. 6, because the goal is to observe and evaluate visually the interlocation of points on a plane.

5. Conclusions

A variety of SOM-based systems and realizations have been examined in the paper on the basis of a set of ecological parameters. We can compare the graphical result presentation by different systems and intuitively evaluate the level of understandability of the presentations. This may influence our choice of the system to be used for data analysis. Each form of the visual presentation of results has advantages and disadvantages. However, deeper results may be obtained on the analyzed data set via a joint use of the systems. The results of visual analysis are summarized in Table 4, where various tendencies on similarities of the parameters and forming their clusters are pointed out.

References

Alhoniemi, E., J. Himberg, J. Parhankangas, J. Vesanto (2000). *SOM Toolbox for Matlab 5*. Helsinki University of Technology, Report A57, Libella Oy Espoo. http://www.cis.hut.fi/ projects/somtoolbox/index.shtml

Deboeck, G. **(**1999). Public domain versus commercial tools for creating Self-Organizing maps. *PC AI Magazine,* January/February, 27–30.

Dzemyda, G. (2001a). Visualization of a set of parameters characterized by their correlation matrix. *Computational Statistics and Data Analysis*, **36** (1), 15–30.

Dzemyda, G. (2001b). Visualization of environmental data via analysis of correlations. In A.Sano, S.Nishioka, H. Tamura (Eds.), *IFAC Workshop on Modeling and Control in Environmental Issues*. Keio University, Yokohama, Japan. pp. 79–83.

Dzemyda, G., V. Tiešis (2001). Visualization of multidimensional objects and the socio-economical impact to activity in EC RTD databases. *Informatica*, **12** (2), 239–262.

Hassinen, P., J. Elomaa, J. Rönkkö, J. Halme, P. Hodju (1999). *Neural Networks Tool – Nenet (Version 1.1).* http://koti.mbnet.fi/∼phodju/nenet/Nenet/General.html

Hellemaa, P. (1998). *The Development of Coastal Dunes and Their Vegetation in Finland*. Dissertation. Fenia 176: 1, Helsinki. ISSN 0015–0010.
http://hul.helsinki.fi/elbanco/julkaisut/mat/maant/vt/developm/index.htm

Kaski, S. (1997). *Data Exploration Using Self-Organizing Maps*. *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series*, No. 82. Espoo, Finish Academy of Technology.

Kleiweg, P. (1996). *Neurale netwerken: Een inleidende cursus met practica voor de studie Alfa-Informatica,* Master's thesis. Rijksuniversiteist Groningen. http://odur.let.rug.nl/∼kleiweg/kohonen/kohonen.html

Kohonen, T. (2001). *Self-Organizing Maps*, 3nd ed. *Springer Series in Information Sciences*, Vol. 30. Springer–Verlag.

Kohonen, T. (2002). Overture. In U. Seiffert, L.C. Jain (Eds.), *Self-Organizing Neural Networks: Recent Advances and Applications*. Heidelberg, New York: Physica–Verl., *Studies in Fuzziness and Soft Computing*,Vol. 78, pp. 1–11.

Kohonen, T., J. Hynninen, J. Kangas, J. Laaksonen (1996). *SOM_PAK: The Self-Organizing Map Program Package*. Helsinki University of Technology, Laboratory of Computer and Information Science, Neural Networks Research Centre. Technical Report A31, FIN-02150 Espoo, Finland. http://www.cis.hut.fi/research/som_lvq_pak.shtml

Kraaijveld, M.A., J. Mao, A.K. Jain (1992). A non-linear projection method based on Kohonen's topology preserving maps. In *Proceedings of 11ICPR, 11th International Conference on Pattern Recognition*. IEEE Computer Society Press, Los Alamitos, CA. pp 41–45.

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

Ultsch, A., H. Siemon (1989). *Exploratory Data Analysis: Using Kohonen Networks on Transputers*. Technical Report 329, Univ. of Dortmund, Dortmund, Germany.

Viscovery SOMine (2002). Viscovery SOMine 4.0 Plus. http://www.eudaptics.com/technology/somine4.html

G. Dzemyda graduated from Kaunas University of Technology, Lithuania, in 1980, and in 1984 received there the doctoral degree in technical sciences (Ph.D.) after postgraduate studies at the Institute of Mathematics and Informatics, Vilnius, Lithuania. In 1997 he received the degree of Doctor Habilius from Kaunas University of Technology. He was conferred the title of Professor (1998) at Kaunas University of Technology. He is a Deputy Director of the Institute of Mathematics and Informatics and a Principal Researcher at the Optimization Department of the institute. The main field of scientific interests is the interaction of optimization and data analysis. The interests include optimization theory and applications, multiple criteria decisions, neural networks, and data analysis.

O. Kurasova received the M.S. degree of informatics from Vilnius Pedagogical University, Lithuania, in 2001. Currently she is a doctoral student at the Department of Optimization of the Institute of Mathematics and Informatics, and a junior lecturer at the Department of Informatics of Vilnius Pedagogical University. Her research interests include neural networks, self-organizing maps, and databases.

Savireguliuojančių žemėlapių programų rezultatų grafinio pateikimo lyginamoji analizė

Gintautas DZEMYDA, Olga KURASOVA

Straipsnyje nagrinėjama savireguliuojančių žemėlapių programinė įranga: SOM-PAK, SOM-TOOLBOX, Viscovery SOMine, Nenet ir dvi akademinės sistemos. Visos jos arba jų ribotų galimybių bandomosios ar demonstracinės versijos yra platinamos nemokamai, didžioji dalis – Internete. Savireguliuojantys žemėlapiai yra neuroninių tinklų rūšis, naudojama duomenims klasterizuoti ir panašumams juose analizuoti. Esama programinė įranga skiriasi realizacijos ypatybėmis ir rezultatų vizualizavimo galimybėmis. Savireguliuojančių žemėlapių programų rezultatų grafinio pateikimo eksperimentinei lyginamajai analizei panaudoti duomenys apie Suomijos pakrantės kopas ir jų vegetaciją. Išryškinti sistemų panašumai ir skirtumai, privalumai ir trūkumai.