Visual analysis of self-organizing maps

Pavel Stefanovič, Olga Kurasova

Institute of Mathematics and Informatics, Vilnius University
Akademijos str. 4, LT-08663, Vilnius, Lithuania
pavel.stefanovic@mii.vu.lt; olga.kurasova@mii.vu.lt

Received: 5 April 2011 / **Revised:** 24 November 2011 / **Published online:** 7 December 2011

Abstract. In the article, an additional visualization of self-organizing maps (SOM) has been investigated. The main objective of self-organizing maps is data clustering and their graphical presentation. Opportunities of SOM visualization in four systems (NeNet, SOM-Toolbox, Databionic ESOM and Viscovery SOMine) have been investigated. Each system has its additional tools for visualizing SOM. A comparative analysis has been made for two data sets: Fisher's iris data set and the economic indices of the European Union countries. A new SOM system is also introduced and researched. The system has a specific visualization tool. It is missing in other SOM systems. It helps to see the proportion of neurons, corresponding to the data items, belonging to the different classes, and fallen in the same SOM cell.

Keywords: multidimensional data visualization, self-organizing map (SOM), visualization of SOM, u-matrix, SOM systems.

1 Introduction

A visualization of multidimensional data is a powerful tool helpful to analyze data, to interpret the results of the analysis, and to draw conclusions on a structure of the data analyzed. Two strategies for visualizing of multidimensional data can be distinguished: direct visualization methods and projection methods based on the reduction of dimension. Some projection methods can be based on artificial neural networks. Each feature, characterizing multidimensional data, is presented in a visual form by direct visualization methods. The methods can be classified into geometric techniques (scatter plots, Andrews curves, parallel coordinates, etc.), iconographic displays (Chernof faces, stars, etc.), and hierarchical techniques (dimensional stacking, "trellis" display, hierarchical parallel coordinates, etc.) [1]. Dimension reduction methods allow us to transform the points (vectors), corresponding to multidimensional data, to a space of lower (2 or 3) dimension, and to visualize two or three dimensional points in a scatter plot. The most popular methods are the principal component analysis, the linear discriminant analysis, multidimensional scaling [2], and locally linear embedding [3].

The artificial neural network (ANN) analysis tends to be an increasingly important branch of computer science. ANN's are helpful in solving various problems: classifica-

tion, forecasting, pattern recognition, etc. They are also used in search of multidimensional data projection onto a space of smaller dimension. ANN's realize some dimension reduction (projection) methods [4]. More popular methods are curvilinear component analysis (CCA), auto-associative neural networks, neural scales, SAMANN, and self-organizing maps (SOM). The CCA is a self-organizing neural network that performs two tasks: vector quantization of the submanifold in the data set (input space) and nonlinear projection of these quantizing vectors towards the output space, providing a revealing unfolding of the submanifold. After learning, the network acquires the ability to continuously map any new point from one space into another: forward mapping of new points in the input space, or backward mapping of an arbitrary position in the output space [5]. Auto-associative neural networks are feed-forward neural networks trained to produce an approximation of the identity mapping between network inputs and outputs using back-propagation or similar learning procedures. The key feature of an auto-associative network is a dimensional bottleneck between input and output [6]. Neural scales are based on radial basis function (RBF) neural networks. The implementation of this principle by a neural network is very simple. The RBF neural network is utilized to predict the coordinates of the data point in the transformed feature space. The locations of the feature points are indirectly determined by adjusting the weights of the network. The transformation is determined by optimizing the network parameters in order to minimize a suitable error measure that embodies the topographic principle [7]. Mao and Jain [8] have suggested a neural network implementation of Sammon's mapping as a method of multidimensional scaling. A specific back-propagation learning rule has been developed to allow a normal feed forward neural network to learn Sammon's mapping in an unsupervised way, called SAMANN.

The self-organizing maps as one type of the neural networks are commonly used for visualizing of multidimensional data, too. The SOM is applied not only to visualize, but also to cluster the data. Self-organizing maps can be combined with dimension reduction methods as a multidimensional scaling [9, 10]. The self-organizing maps are applied in various areas: medicine, financial, ecological, military, engineering, law enforcement, and other fields. For example in the financial area, if we analyze historical data, we can forecast the future stock prices. If we know the company or individual financial capabilities, the networks decide to give them credit or not [11]. Also SOM can be used for prediction of changes of bankruptcy classes [12].

The investigation of this article is focused on the opportunities to visualize the self-organizing maps. In the simplest case, a table is created by the SOM. Each cell corresponds to a neuron of the SOM. The SOM table requires an additional visualization because the simple table does not show how far the vectors, corresponding to the cells, are in the input multidimensional space. The possibilities to visualize the SOM table in four systems (NeNet, SOM-Toolbox, Databionic ESOM, and Viscovery SOMine) are analyzed here. We introduce a new SOM system, which has a specific visualization way. As the target of the SOM is data clustering and visualization, the systems are assessed according to their ability to cluster and visualize data correctly. Each system investigated has its own visualization way, so the main research object is to show various opportunities of the SOM visualization and to introduce a new system, having a new specific visualization tool.

2 Self-organizing maps

T. Kohonen began to explore self-organizing maps (SOM) in 1982. Almost thirty years have passed since that time, but SOM not lose their popularity. New extensions and modifications are developed constantly. The main objective of the SOM is to preserve the topology of multidimensional data, i. e., to get a new set of the data from the input data such that the new set preserves the structure (clusters, relationships, etc.) of the input data. The SOM is applied to cluster and visualize the data. The self-organizing map is a set of nodes, connected to one another via a rectangular or hexagonal topology. The rectangular topology of SOM is presented in Fig. 1. Here a circle represents a node. The connections between the inputs and the nodes have weights, so a set of weights corresponds to each node. The set of weights forms a vector $M_{ij}, i = 1, \ldots, k_x, j = 1, \ldots, k_y$ that is usually called a neuron or a codebook vector. In a rectangular SOM, k_x is the number of rows, and k_y is the number of columns. Usually the self-organizing map is called a self-organizing neural network. The dimension of the codebook vector is the same as that of the number of inputs, $M_{ij} = \{m_{ij}^1, m_{ij}^2, \ldots, m_{ij}^n\}$ [13].

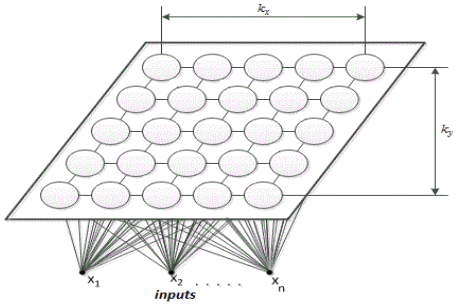

Fig. 1. Two-dimensional SOM (rectangular topology)

A self-organizing map (neural network) learns in an unsupervised manner. The learning starts from the components of the vectors M_{ij} initialized at random. If n-dimensional vectors $X_1, X_2, \ldots, X_m$ are needed to map, the components of these vectors $x_1, x_2, \ldots, x_n$ are passed to the network as the inputs, where m is the number of the vectors, n is the number of components of the vectors. At each learning step, an input vector $X_p \in \{X_1, X_2, \ldots, X_m\}$ is passed to the neural network. The vector X_p is compared with all the neurons M_{ij}. Usually the Euclidean distance $\|X_p - M_{ij}\|$ between the input vector X_p and each neuron M_{ij} is calculated. The vector (neuron) M_c with the minimal Euclidean distance to X_p is designated as a winner. The components of neurons are adapted according to the learning rule:

$$M_{ij}(t+1) = M_{ij}(t) + h_{ij}^c(t)\big(X_p - M_{ij}(t)\big). \tag{1}$$

Here t is a number of iterations. The learning is repeated until the maximum number of iterations is reached.

After training the SOM network, its quality must be evaluated. Usually two errors (quantization and topographic) are calculated. The quantization error E_{QE} shows how well neurons of the trained network adapt to the input vectors. Quantization error (2) is the average distance between the data vectors X_p and their neuron-winners $M_{c(p)}$.

$$E_{QE} = \frac{1}{m}\sum_{p=1}^{m} \|X_p - M_{c(p)}\|. \tag{2}$$

The topographic error E_{TE} shows how well the trained network keeps the topography of the data analyzed. The topographic error (3) is calculated by the formula:

$$E_{TE} = \frac{1}{m}\sum_{p=1}^{m} u(X_p). \tag{3}$$

If the neuron-winner of vector X_p is near the neuron, the distance from X_p to it is the smallest one, disregarding to the neuron-winner, then $u(X_p) = 0$, otherwise, $u(X_p) = 1$.

When the network is trained, all the input vectors are passed to the network once more and a set of the neurons-winners is found. Each neuron-winner corresponds to one or more input vectors, i.e., the input vectors are distributed among the elements of the map. In the simplest case, a table (grid) is created by the SOM (Table 1), the order numbers of the input vectors, labels of classes that they belong to, or other information about the input vectors are marked in the cells that correspond to their neurons-winners. Thus, multidimensional data are transformed to some discrete structure in the SOM. This fact can be regarded as a distribution of multidimensional data on the plane. Nodes of the grid (the number of rows and columns) indicate a location of data on the plane. Table 1 is obtained, when the iris data set, described in Subsection 3.1, is mapped in the SOM.

Table 1. Simple SOM table.

2,3		2	2		2,3	2
2,3		2		2	2	2
3	2,3	2		2	2	2
3			2			
3		2	2			
3	3	2				1
3	2	2			1	1

Such table is not highly informative, the table does not answer the question, how much the vectors of the neighboring cells are close in the n-dimensional space. Therefore it is necessary to seek ways for improving the quality of visualization which facilitates the interpretation of results. Several visualization techniques have been developed. The most popular one is based on the so-called unified distance matrix (u-matrix) [14]. The

u-matrix shows the relations between the neighboring neurons.

$$\textbf{u-matrix} = \begin{pmatrix} u_{11} & u_{11|22} & u_{12} & u_{12|13} & \cdots & u_{1k_y} \\ u_{11|21} & & u_{11|22} & & \cdots & u_{1k_y||2k_y} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{k_x1} & u_{k_x1||k_x2} & u_{k_x2} & u_{k_x2||k_x3} & \cdots & u_{k_xk_y} \end{pmatrix}. \quad (4)$$

The values of elements $u_{ij|i(j+1)}u_{ij|(i+1)j}$ are the distances between the neighboring neurons M_{ij} and $M_{i(j+1)}$ (M_{ij} and $M_{(i+1)j}$), respectively. The values of elements u_{ij} can be the average of neighboring elements of the u-matrix, e.g., if u_{ij} has four neighbors, then $u_{ij} = (u_{i(j-1)|ij} + u_{ij|i(j+1)} + u_{(i-1)j|ij} + u_{(i+1)j|i(j+1)})/4$, if the number of the neighbors is smaller, the average is computed with a smaller number of elements.

The SOM is colored by the values of u-matrix elements. If the grey scale is used, a dark color between the neurons corresponds to a large distance. A light color between the neurons signifies that the codebook vectors are close to each other in the input space. Light areas can be thought as clusters and dark areas as cluster separators. Not only the grey scale is used. In many systems, other color scales are applied. The SOM, represented by the u-matrix, is presented in Fig. 2. The color scale is displayed near the SOM. The number denotes the values of u-matrix elements and that of the distances between neighboring neurons.

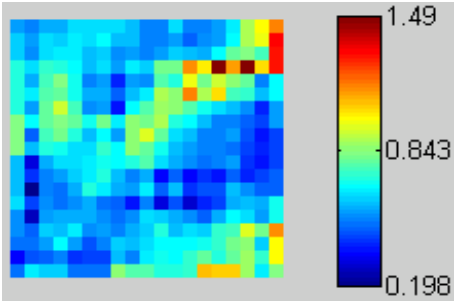

Fig. 2. U-matrix representation in SOM.

3 Possibilities of SOM visualization

Currently, various systems of self-organizing maps have been designed. Each of them has its own specific visualization techniques [15]. A comparative analysis of the graphical result presentation in the SOM software is presented in [16]. In this article, other possibilities are focused. In [16], SOM-Toolbox and Viscovery SOMine were analyzed, too. However, these systems are upgraded so far, some features have been added which are analyzed in this article. Here we analyze the possibilities of SOM visualization in four systems: NeNet, SOM-Toolbox, Databionic ESOM and Viscovery SOMine and a new system is introduced.

3.1 Data sets analyzed

In order to demonstrate the possibilities for SOM visualization in various systems, two data sets (Fisher's iris data and economic indices of the European Union countries) are used. The iris data set consists of the values of some features of three species of flowers: Iris Setosa, Iris Versicolor and Iris Virginica [17]. Four features are measured: x_1 is sepal length, x_2 is sepal width, x_3 is petal length, and x_4 is petal width. Four-dimensional vectors $X_1, X_2, \ldots, X_{150}$ are formed, where $X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$, $i = 1, \ldots, 150$. The vectors $X_1, X_2, \ldots, X_{50}$ represent class I (Iris Setosa), vectors $X_{51}, X_{52}, \ldots, X_{100}$ represent class II (Iris Versicolor) and vectors $X_{101}, X_{102}, \ldots, X_{150}$ represent class III (Iris Virginica).

The economic data set consists of the values of some features of the European Union (EU) countries and countries, that strive to become EU members. We have chosen economic indices of the EU countries in 2009 [18]: x_1 is a compensation of employees, x_2 is the final consumption expenditure of households and non-profit institutions serving the households, x_3 is the final consumption expenditure of general government, x_4 is gross fixed capital formation (investments), x_5 is goods and services, imports and exports, x_6 is labor productivity per person employed. Six-dimensional vectors $X_1, X_2, \ldots, X_{31}$ are formed, where $X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6})$, $i = 1, \ldots, 31$. The vectors $X_1, X_2, \ldots, X_6$ represent the countries, that established the European Union (Belgium, German, France, Italy, Luxemburg and the Netherland) (class I), vectors $X_7, X_8, \ldots, X_{15}$ represent the countries, that joined EU in 1957–1995 (Denmark, Ireland, Greece, Spain, Austria, Portugal, Finland, Sweden, United Kingdom) (class II), vectors $X_{16}, X_{17}, \ldots, X_{27}$ represent the countries, that joined EU in 2004–2007 (Czech Republic, Estonia, Cyprus, Latvia, Lithuania, Hungary, Malta, Poland, Slovenia, Slovakia, Bulgaria, Romania) (class III), $X_{28}, X_{29}, \ldots, X_{31}$ represent the countries, that are striving to be EU members (Macedonia, Turkey, Iceland, Croatia) (class IV).

The first data set (iris) is commonly used for testing various data mining methods, as well as visualization methods, because the structure of the data is known. Usually, the vectors of class I form a cluster, and the vectors of classes II and III form another cluster. The background of the other data set is economic. We want to see how the SOM clusters these data, how the clusters differentiate in the map.

3.2 Visualization in NeNet

In the system NeNet, SOM visualization is based on the u-matrix [19]. Here shades represent the u-matrix values. A light shade of cells and borders of the map cells denotes that the codebook neighboring vectors are near to each other in the input space, while the dark shades shows that they are far away. After the SOM has been trained using the iris data set, the vectors of class I split off from that of classes II and III (Fig. 3). We can see dark shades between the two clusters. There are no significant differences between the vectors of classes II and III. If the vectors from two different classes fall in to the same cell, it means that their features are similar.

If the economic data set is presented in the SOM, we can see a lot of vectors of classes II, III and IV in one map side. They are separated from the vectors of class I by

dark shades. In Fig. 4, we see the abbreviations of the names of countries. The numbers near the abbreviations correspond to the numbers of the classes.

Two other views of u-matrix representation are possible in system NeNet (Fig. 4). In the left map, we can see the class numbers not inside of the cells, but on the nodes. The image in the right map looks like that, presented in Fig. 3, only there are no colored borders and other shades are applied. We can see the same clusters of the data mapped.

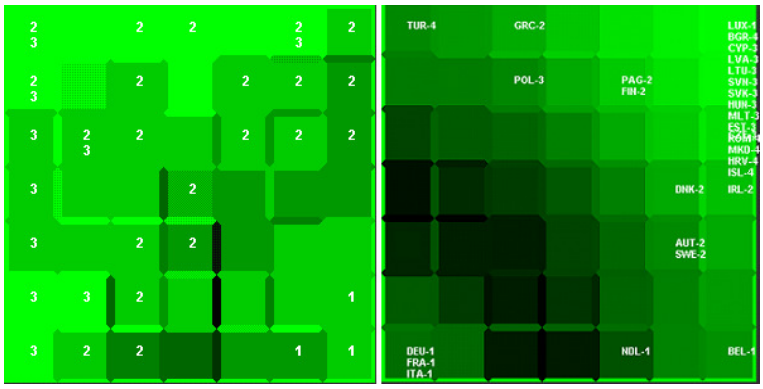

Fig. 3. Iris data set (in the left) and economic data set (in the right) in 7 × 7 SOM, created by system NeNet.

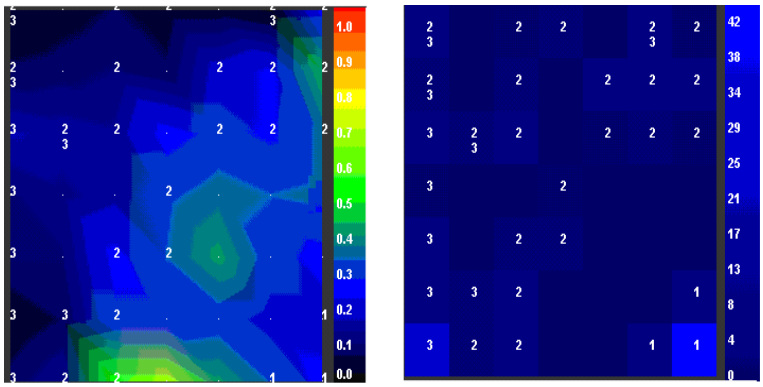

Fig. 4. Iris data set in 7 × 7 SOM (optional visualization), created by system NeNet.

3.3 Visualization in SOM-Toolbox

Some representations of the u-matrix are possible in SOM-Toolbox, too [20]. SOM-Toolbox is an additional toolbox of Matlab and it includes many functions for training various topologies of SOM with various learning parameters, computing various errors of qualities, visualizing SOM using u-matrices, component planes, computing correlation,

clustering, etc. SOM-Toolbox also has some features of other data analysis methods related to vector quantization (k-means, learning vector quantization), dimension reduction (principal component analysis, Sammon's projection), etc.

As distinct from the system NeNet, the cells of the map, obtained using SOM-Toolbox, are separated not by borders, but by the cells of the same size. So if we choose to train the map of size 7×7, we get the map of size 13×13 as a result. The various u-matrix shades can be used. The values of the u-matrix are represented in the scale near the map (Fig. 5). We can see that the vectors in class I are distinguished from the vectors from classes II and III by light shades (red, yellow and green), which means that they are far away from one another. The dark shades (blue) show that the vectors are near. When the economic data are mapped, a majority of the vectors from classes I and II falls on in the right corner of the map, the vectors from classes III and IV falls on the other side.

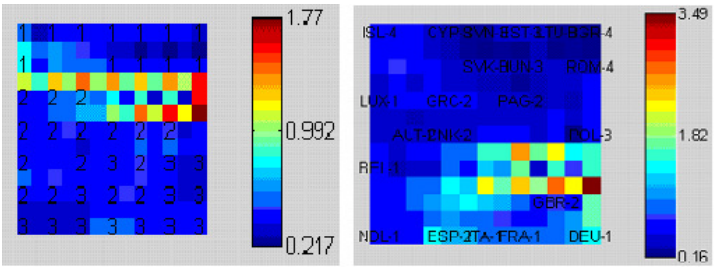

Fig. 5. Iris data set (in the left) and economic data set (in the right) in 7×7 SOM, created by SOM-Toolbox.

In Fig. 6, another view of the u-matrix is presented. We do not see the cells, but only the nodes, like in the system NeNet. The clusters are divided by green, red and yellow zones.

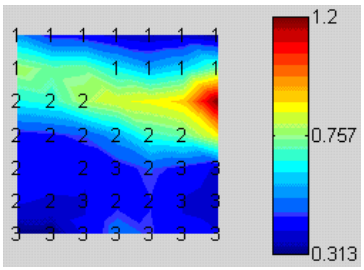

Fig. 6. Iris data set in 7×7 SOM (another view of u-matrix), created by SOM-Toolbox.

In this system, we can see component planes. The u-matrix, obtained according to the features of data (a component of the vector), is represented in each component plane. The component planes of the iris data set are presented in Fig. 7. The first map represents the sepal length, the second one represents the sepal width, the third one represents the

petal length and the fourth one represents the petal width. Component planes show which feature has a more significant influence on the results of clustering.

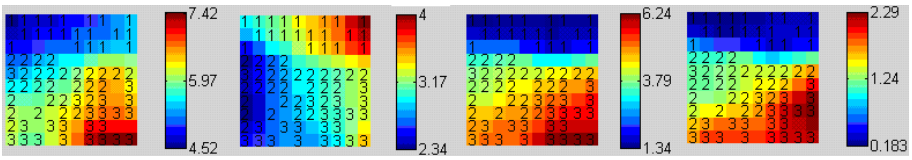

Fig. 7. Iris data set in 7×7 component planes, created by SOM-Toolbox.

3.4 Visualization in Databionic ESOM

The topology preservation of the SOM projection is of little use when using small maps. Emergent phenomena involve by definition a large number of individuals, where large means at least a few thousand. This is why Databionic use large SOM and this system is called Databionic Emergent Self-Organizing Maps (ESOM) [21]. System is created for large data sets, but also can be trained and using small data sets. The map results can be visualized in various ways: p-matrix, u-matrix, Components planes, TwoMatch, Gap, Opinion, Random and The Smoothed Data Histograms.

When the iris data set is mapped using Databionic ESOM (Fig. 8), we see clusters similar to that, obtained by NeNet and SOM-Toolbox. The vectors in class I are separated from that of classes II and III. As distinct from NeNet and SOM-Toolbox, there are no borders of cells, and we cannot see the cells, because they are small enough, as a large map (50×50). In this map, clusters are divided with different landscapes. For example, brown areas shows different clusters are separated by the "mountains". Blue areas show "water", which means that the vectors are near to each other. The distance scale which represents all colors we can find in system options.

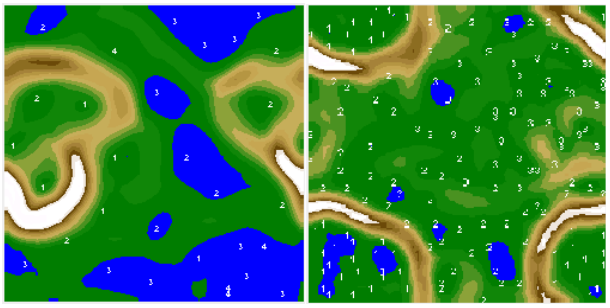

Fig. 8. Iris data set (in the left) and economic data set (in the right) in 50×50 SOM, created by system Databionic ESOM.

In this system, the class labels must to be written in numerical values, so in order to map the economic data set, only the class numbers without abbreviations are presented in Fig. 8. The brown arcs separate the vectors of classes I and II from that of classes

III and IV, like “mountains”. The vectors in classes III and IV are placed in the same blue area, which looks like “water”. Such a cluster extraction method is unusual. The similarities and differences between the data are expressed distinctly. Some other visual representations of SOM are presented in Fig. 9.

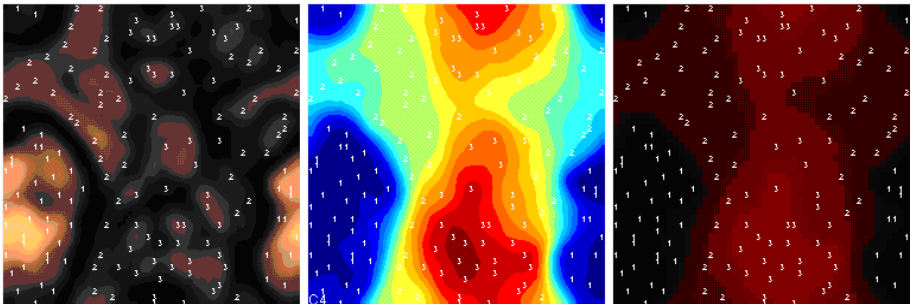

Fig. 9. Iris data set in 50×50 SOM (left – p-matrix, centre – components planes, right – opinion), created by system Databionic ESOM.

3.5 Visualization in Viscovery SOMine

The system Viscovery SOMine is a commercial program [22]. The system is updated constantly. In this system, a user can find a lot of various possibilities. Due to capacity of the system, its usage is much more complex than that of the system, investigated before. After the SOM training, it is possible to see the u-matrix of each parameter (component planes), but the user can configure the views how he (she) wants. When the iris data set is analyzed, the system sets three clusters automatically (Fig. 10), but the user can select the number of the clusters manually. In Fig. 10, we see only the vectors from class I in the red cluster, only the vectors of classes III in the yellow cluster, and the vectors of classes II and III in the light blue cluster.

When the economic data set is mapped, the system sets seven clusters automatically. If we set four clusters as we have four classes, the result obtained is presented in Fig. 10. The

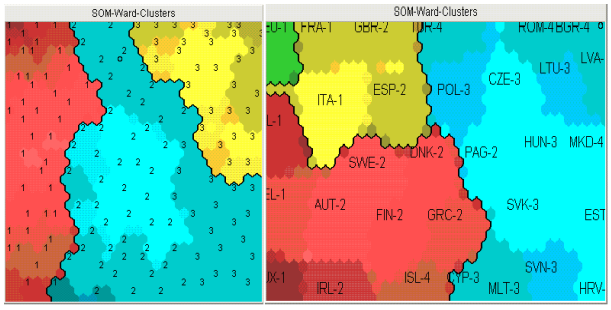

Fig. 10. Iris data set (in the left) and economic data set (in the right) in SOM, created by system Viscovery SOMine.

majority of vectors from classes III, IV is assigned to the light blue cluster. The vectors from classes I, II are assigned to the red and yellow clusters, and a vector from class I (Germany) is assigned to the green cluster. There is no distance scale between vectors, but the system allows us a clear view of clusters. Viscovery SOMine has some optional visualization views: shaded clusters, flat clusters, global shading and u-matrix (Fig. 11).

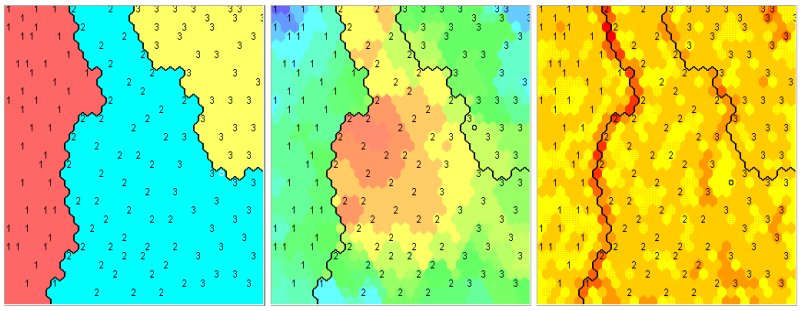

Fig. 11. Iris data set in SOM (left – flat clusters, middle – global shading, right – u-matrix), created by system Viscovery SOMine.

4 A new system of SOM

The analysis of SOM systems has showed that the systems have many possibilities to visualize SOM's. However systems have a common disadvantage. If the classes, to which the data belong to, are known, and the labels of the classes are displayed in the map, it is difficult to understand how much the vectors from one or other class correspond to a cell (neuron), because usually only different (but not the same) labels are shown. Especially it is important, whether the vectors from different classes fall into a cell. We do not know how many vectors are from of the same class or of different class, and what their proportions are.

In order to solve the problems, some ways are developed in the systems. For example, a histogram map is implemented in the system NeNet (Fig. 12). The histogram shows how many vectors fall into a cell, but it is not obvious how many vectors are from one or another class. It is possible to create a map, where all the labels (not only different) are shown, in SOM-Toolbox (Fig. 13). The view of such a map is very complicate, because the labels are overlapping, it is not clear which label corresponds to which cell. The systems Databionic ESOM and Viscovery SOMine do not even have such abilities.

We have developed a new SOM system, which allows us to see how many vectors and which classes they belong to fall into the same cell of a map. Besides a new way for visualizing SOM, a neighborhood function, different than in other systems is implemented in the new system. Usually, Gaussian and bubble neighborhood functions are used. However other neighborhood function (5) is implemented in the new system [23].

$$h_{ij}^{c} = \frac{\alpha}{\alpha\eta_{ij}^{c} + 1}, \quad \text{where } \alpha = \max\left(\frac{e + 1 - \hat{e}}{e}, 0.01\right). \tag{5}$$

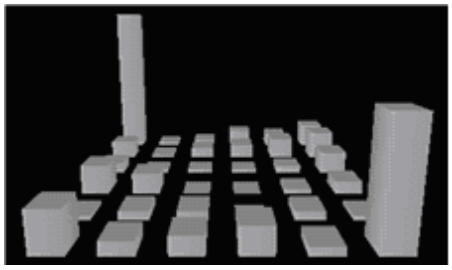

Fig. 12. Histogram map, obtained by the system NeNet.

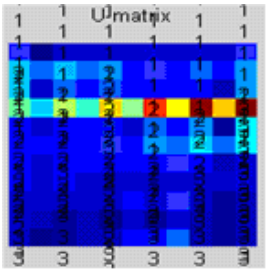

Fig. 13. SOM, obtained by SOM-Toolbox, where all labels are shown.

Here e is the number of epochs, set before training, $\hat{e}$ is the number of the current epoch, η_{ij}^c is the neighboring rank between M_c and M_{ij}. The neurons M_{ij} are recalculated in each epoch, if inequality (6) is valid:

$$\eta_{ij}^c \leq \left[\alpha \max(k_x, k_y; 1)\right]. \tag{6}$$

The system is implemented in Matlab. The graphical user interface of the system is presented in Fig. 14. Before training, the map size (the numbers of rows and columns) has

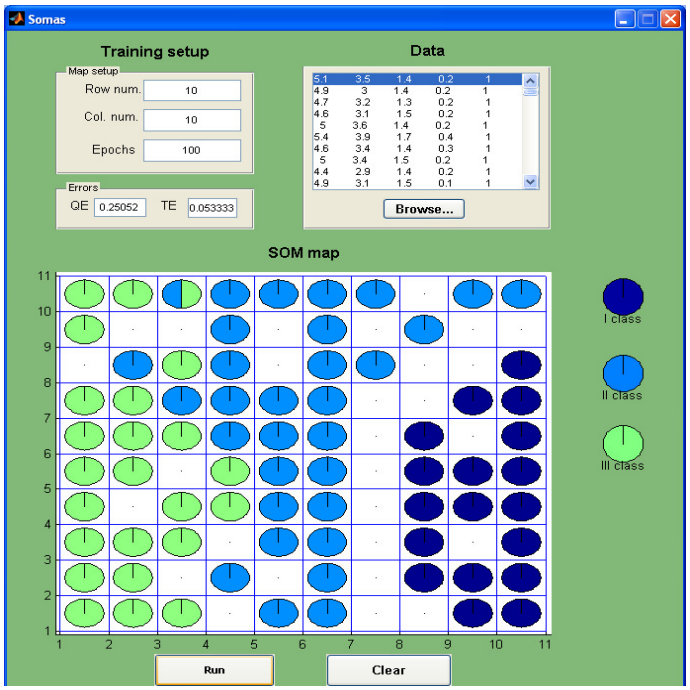

Fig. 14. Iris data set in 10×10 SOM, created by the new system.

to be chosen. We also have to choose the number of epochs, i.e., how many times vectors will be presented to the network and load a data set. The data set have to be written in txt file. Data must be written in rows, the last column is for labels of data classes.

After the iris data has been analyzed, a map, obtained by the new system, is presented in Fig. 14. Little pie charts are shown in the cells of SOM. The pie shows a proportion between the vectors that belong to the different classes and fall into a cell. The blue pie and a slice of pie correspond to the vectors of class I. The light blue pie and a slice of pie correspond to the vectors of class II. The green pie and a slice of pie correspond to the vectors from class III. The vectors from class I fall into cells in the right corner at the bottom of the map. The vectors of classes II and III are distributed in the other part of the map. There are some empty cells between the cluster of class I and the vectors from classes II and III. It means that these clusters are far away from one another. There is no distance scale in this system, so we cannot say how far neuron-winners are. Some pies are formed from two slices of different colors. If one vector from class II and two vectors from class III fall into a cell, the light blue slice takes 1/3 of the pie, and the green slice takes 2/3 of the pie.

After the economic data set has been analyzed, the map obtained is presented in Fig. 15. Only the vectors of class I are mostly distributed on the right side of the map. The vectors of class II are distributed in the center. The vectors of classes III and IV are mostly distributed on the left corner at the top. Some vectors from different classes also fall into the same cell, so some pies are formed from the slices of different colors.

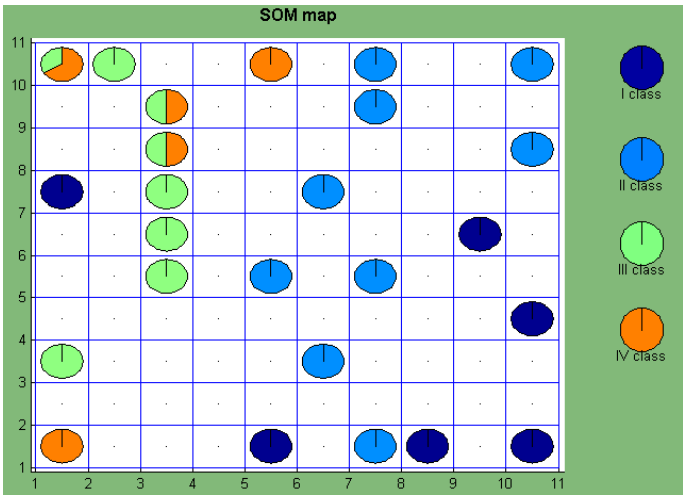

Fig. 15. Economic data set in 10×10 SOM, created by new system.

Visualization by proportions also has its own drawbacks – what if there is a much larger number of classes and a much larger number of instances, especially if the numbers of different instances, which belong to different classes, vary a lot (e.g., from 1 to 100) –

the visualization can become very complex and confusing. So, the new system introduced can be used when the data with some classes (not more five) are analyzed. When the data with more than five are analyzed, the most of clustering methods confronts with difficulties. In the other hand, usually the number of classes of the data from the practical problems is not exceeding five. Thus, the new system can be applied successfully in a lot of applications.

The proposed system distinguishes by possibility to show the proportion of neurons, corresponding to the data items, belonging to the different classes, and fallen in the same SOM cell. The simple way for SOM visualization, implemented in the system, solves the problem that occurred in the other SOM systems analyzed.

5 Comparison of systems

All self-organizing map systems have the advantages and disadvantages. In order to compare all systems seven criteria were selected. The results are presented in Table 2: the plus sign means that system has this possibility and the minus sign means that the possibility is missing.

As we can see in the Table 2, the system NeNet has the most limited usages, so it is useful only for beginners. The system Databionic ESOM and Viscovery SOMine have some disadvantages which are not so important for data analysis. The most universal systems are SOM-Toolbox and the new proposed SOM system. Both them can be the perfect choice for deep analysis of any data set.

Table 2. System comparison by the criteria selected.

	Possibilities						
System	Amount of data items and dimentions	Represent class names of all neuron winners in the same cell of a map	Represent proportion of different classes of neuron winners in the same cell of a map	Easy data file preparation	Represent distance between neutrons in a map	Can be used some neighboring function and any learning rate	Can be used different learning step (epochs or iterations)
NeNet	−	−	−	+	+	−	−
SOM-Toolbox	+	+	−	+	+	+	+
Databionic ESOM	+	−	−	−	+	−	−
Viscovery SOMine	+	−	−	+	+	−	−
New system of SOM	+	+	+	+	−	+	+

6 Conclusions

The self-organizing maps have possibilities not only to cluster, but also to visualize multidimensional data. However the simple SOM table is not enough in order to reveal the characteristics of the data analyzed. It is necessary to supplement the table with extra information, i.e., to visualize the table additionally.

In this article, some ways for visualization of self-organizing maps have been investigated. Various ways are implemented in the popular systems. Four systems (NeNet, SOM-Toolbox, Databionic ESOM and Viscovery SOMine) have been analyzed here. All of them have the possibility to visualize SOM by the u-matrix, but they use different shades. Other visualization ways (e.g., p-matrix, component planes, etc.) are implemented in some of the systems. In order to demonstrate the abilities of the systems, the iris and economic data sets are mapped in SOM. The experiments have showed that all the systems cluster and visualize both data sets rather similarly. The clusters are expressed in some systems quite differently than in the other ones. We recommend using several systems to obtain some various maps. They help to draw conclusions on the data sets analyzed. The systems are compared by some criteria. The conclusions are done that the system NeNet is useful only for beginners and for educational purposes, the most universal systems are SOM-Toolbox and the new proposed system.

The new system for SOM visualization has been introduced and investigated here. The new way of visualization of data with the known classes are implemented in this system. The system differs from the others in the fact that the map shows the proportions between the data belong to the different classes and fall into a cell. This feature is missing in the other systems. The new system allows us to investigate properties of SOM. Different learning rates, neighboring functions and learning steps (epochs and iteration) are implemented in the system. Such extensions help to analyze the data sets in different ways.

References

1. P.E. Hoffman, G. Grinstein, A survey of visualizations for high-dimensional data mining, in: U. Fayyad, G.G. Grinstein, A. Wierse (Eds.), *Information Visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann Publishers Inc., 2002, pp. 47–82.
2. I. Borg, P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, Springer-Verlag, 2005.
3. S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, **290**, pp. 2323–2326, 2000.
4. G. Dzemyda, O. Kurasova, M. Medvedev, Dimension reduction and data visualization using neural networks emerging, in: I. Maglogiannis, K. Karpouzis, M. Wallace, J. Soldatos (Eds.), *Artificial Intelligence Applications in Computer Engineering*, Vol. 160, IOS Press, 2007, pp. 25–49.
5. P. Demartines, J. Hérault, Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets, *IEEE Trans. Neural Networks*, **8**(1), pp. 148–154, 1997.

6. M.A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, *AIChE J.*, **37**(2), pp. 233–243, 1991.

7. D. Lowe, M.E. Tipping, Feed-forward neural networks and topographic mappings for exploratory data analysis, *Neural Comput. Appl.*, **4**, pp. 83–95, 1996.

8. J. Mao, A.K. Jain, Artificial neural networks for feature extraction and multivariate data projection, *IEEE Trans. Neural Networks*, **6**, pp. 296–317, 1995.

9. O. Kurasova, A. Molytė, Integration of the self-organizing map and neural gas with multidimensional scaling, *Information Technology and Control*, **40**(1), pp. 12–20, 2011.

10. O. Kurasova, A. Molytė, Quality of quantization and visualization of vectors obtained by neural gas and self-organizing map, *Informatica*, **22**(1), pp. 115–134, 2011.

11. E. Merkevičius, G. Garšva, R. Simutis, Neuro-discriminate model for the forecasting of changes of companies financial standings on the basis of self-organizing maps, in: *ICCS'07 Proceedings of the 7th international conference on Computational Science, Part II*, Lect. Notes Comput. Sci., Vol. 4488, Springer Verlag, Heidelberg, 2007, pp. 439–446.

12. E. Merkevičius, G. Garšva, Prediction of changes of bankruptcy classes with neuro-discriminate model based on the self-organizing maps, *Information Technology and Control*, **36**(1A), pp. 145–151, 2007.

13. T. Kohonen, *Self-Organizing Maps*, 3rd edition, Springer Ser. Inf. Sci., Springer-Verlag, Berlin, 2001.

14. A. Ultsch, H. Siemon, Exploratory data analysis: Using Kohonen networks on transputers, Technical Report, No. 329, Univ. of Dortmund, Dortmund, Germany, 1989.

15. O. Kurasova, P. Stefanovič, Saviorganizuojančių neuroninių tinklų sistemų lyginamoji analizė, *Informacijos mokslai*, **50**, pp. 334–339, 2009 (in Lithuanian).

16. G. Dzemyda, O. Kurasova, Comparative analysis of the graphical result presentation in the SOM software, *Informatica*, **13**(3), pp. 275–286, 2002.

17. A. Asuncion, D.J. Newman, UCI Machine Learning Repository, University of California, School of Information and Computer, Irvine, CA, 2007, `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

18. Eurostat, 2010, `http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home`.

19. P. Hassinen, J. Elomaa, J. Rönkkö, J. Halme, P. Hodju, Neural Networks Tool – NeNet, 1999, `http://koti.mbnet.fi/~phodju/nenet/Nenet/General.html`.

20. J. Vesanto, J. Himberg, E. Alhoniemi, J. Parhankangas, SOM Toolbox for Matlab 5, 2005, `http://www.cis.hut.fi/projects/somtoolbox/about.shtml`.

21. A. Ultsch, F. Moerchen, ESOM-Maps: Tools for clustering, visualization, and classification with emergent SOM, Technical Report Dept. of Mathematics and Computer Science, University of Marburg, Germany, No. 46, 2005, `http://databionic-esom.sourceforge.net`.

22. Viscovery SOMine 5.0, 2010, http://www.viscovery.net/self-organizing-maps.

23. P. Stefanovič, O. Kurasova, Influence of learning rates and neighboring functions on self-organizing maps, in: J. Laaksonen, T. Honkela (Eds.), *Advances in Self-Organizing Maps, WSOM 2011*, Lect. Notes Comput. Sci., Vol. 6731, Springer Verlag, Heidelberg, 2011, pp. 141–150.