INFORMATICA, 2011, Vol. 22, No. 1, 115–134

Quality of Quantization and Visualization of Vectors Obtained by Neural Gas and Self-Organizing Map

Olga KURASOVA[1,2], Alma MOLYTĖ[1]

[1] *Vilnius University, Institute of Mathematics and Informatics*
Akademijos 4, LT-08663 Vilnius, Lithuania
[2] *Vilnius Pedagogical University*
Studentų 39, LT-08106 Vilnius, Lithuania
e-mail: olga.kurasova@mii.vu.lt, alma.molyte@gmail.com

Received: October 2009; accepted: October 2010

Abstract. In this paper, the quality of quantization and visualization of vectors, obtained by vector quantization methods (self-organizing map and neural gas), is investigated. A multidimensional scaling is used for visualization of multidimensional vectors. The quality of quantization is measured by a quantization error. Two numerical measures for proximity preservation (Konig's topology preservation measure and Spearman's correlation coefficient) are applied to estimate the quality of visualization. Results of visualization (mapping images) are also presented.

Keywords: self-organizing map, neural gas, multidimensional scaling, quantization error, proximity preservation, Konig's measure, Spearman's rho.

1. Introduction

A data analysis is one of the important factors for discovering new knowledge on the objects analysed. The term "object" can cover various things: people, equipment, produce of manufacturing, etc. Statistical analysis methods (classification, regression, cluster analysis, time series, etc.) are applied for a data analysis for a long time. Recently the other methods of data mining becomes the focus of attention of data analyzers (Fayyad *et al.*, 1996; Dunham, 2003; Cios *et al.*, 2007).

The following definitions of data mining and knowledge discovery in databases (KDD) are coined in Fayyad *et al.* (1996): KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data; data mining is a step in the KDD process that consists of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns (or models) over the data".

Statisticians and researchers in computer science often use different terms to describe the same objects, facts, and other things. Actually, both statistical analysis and data mining methods deal with the same or similar problems. A part of the problems are solved by the statistical methods successfully enough, however it is necessary to take into account particularity of data mining to solve some problems properly.

Data mining utilizes statistical methods, but there are more heuristics in data mining methods than in statistical ones that are defined more strictly mathematically. For this reason, when the new methods of data analysis are developed, it is useful to join the good properties of approaches of both groups. Combination of statistical analysis and data mining methods allows us to avoid the drawbacks the methods applied individually.

Often the object analysed can be described by some features. Denote features, characterized any set of objects, by $x_1, x_2, \ldots, x_n$. A combination of values of all features characterizes a particular object $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ from the whole set $X = \{X_1, X_2, \ldots, X_m\}$, where n is the number of features and m is the number of objects analysed. As the number n of the features often is more than two, we deal with multidimensional data. If the values of features are numerical ones, $X_1, X_2, \ldots, X_m$ are the n-dimensional vectors, and $x_{i1}, x_{i2}, \ldots, x_{in}$ are the components of vector $X_i,\ i = 1, \ldots, m$. Often the vectors $X_1, X_2, \ldots, X_m$ are interpreted as points in the n-dimensional space R^n, where n defines the dimensionality of the space. In fact, we have a matrix of numerical data $X = \{X_1, X_2, \ldots, X_m\} = \{x_{ij},\ i = 1, \ldots, m,$ $j = 1, \ldots, n\}$. The rows of this matrix are vectors $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, $i = 1, \ldots, m$, here x_{ij} is jth component of ith vector.

The data matrix X can be analysed by various data mining methods. This paper deals with two groups of methods: vector quantization and visualization. Here we investigate the quality of quantization and visualization of vectors obtained by quantization methods. Vector quantization is useful for reducing of an amount of the data. Visualization aims at integrating the human in the data analysis process, applying human perceptual abilities to the analysis of large data sets available in today's computer systems. Visualization methods used here are based in reducing dimensionality of multidimensional data. When combining vector quantization and visualization it is possible to get more knowledge than the methods used individually.

The rest of paper is organized as follows. In Section 2, a vector quantization is described and two methods (self-organizing map and neural gas) for vector quantization are introduced. The aim of visualization of multidimensional data is presented, and multidimensional scaling as one of visualization methods is described in more detail in Section 3. In Section 4, we introduce combinations of the neural gas and self-organizing map with a multidimensional scaling. Some results of experimental investigations and comparisons are presented in Section 5. Section 6 concludes the paper.

2. Vector Quantization

Vector quantization (VQ) is a classical signal-approximation method that usually forms a quantized approximation to the distribution of the input data vectors $X_l \in R^n$, $l = 1, \ldots, m$, using a finite number of so-called codebook (or reference) vectors $M_i \in R^n$, $i = 1, \ldots, N$. Once the codebook is chosen, the approximation of X_l means finding the codebook vector M_i closest to X_l, usually in the Euclidean metric (Kohonen, 2001).

The objective of vector quantization for a given data set X is to discover the optimal codebook, containing a predetermined number N of codebook vectors $M_i \in R^n$, $i = 1, \ldots, N$, which guarantee minimization of the chosen distortion metric (usually Euclidean distance) for all vectors from the data set X. Each codebook vector has an associated integer index used for referencing (Cios *et al.*, 2007). In other words, the aim of quantization methods is to change the values of codebook vectors so that they represent the properties of the analysed vectors $X_l,\ l = 1, \ldots, m$.

Vector quantization is used for data compression, missing data correction, classification, etc. It can be used for data clustering, too. In that case, the codebook vectors are representatives of clusters. Some methods for vector quantization are based on neural networks: self-organizing map (SOM), learning vector quantization (LVQ; Kohonen, 2001), neural gas (NG; Martinetz, 1991). Here the neurons correspond to the codebook vectors. The methods can be used instead of other well-known clustering methods as k-means (Šutienė *et al.*, 2010).

2.1. *Self-Organizing Map and Neural Gas*

The self-organizing map (SOM) is a class of neural networks that are trained in an unsupervised manner using a competitive learning (Kohonen, 2001). It is a well-known method for a vector quantization. Moreover, the SOM is used for mapping a high-dimensional space onto a low-dimensional one. The neural gas is a biologically inspired adaptive algorithm, proposed in Martinetz (1991). It sorts the input signals according to how far away they are. The algorithm was named "neural gas" because of the dynamics of the vectors during the adaptation process which distribute themselves like a gas within the data space.

An array of vectors (codebook) M is created in both the neural gas (NG) and the self-organizing map (SOM) methods. The codebook vectors are often called neurons. Array M is one-dimensional in the neural gas $M = \{M_1, M_2, \ldots, M_N\}$, where $M_i \in R^n$, $i = 1, \ldots, N$, N is the number of codebook vectors. There are two types of SOM topology: hexagonal and rectangular. The rectangular SOM is a two-dimensional array (grid) of neurons $M = \{M_{ij},\ i = 1, \ldots, \mathit{rows},\ j = 1, \ldots, \mathit{cols}\}$, where $M_{ij} \in R^n$, rows is the number of rows, cols is the number of columns, and the total number of neurons is $N = \mathit{rows} \times \mathit{cols}$ (Fig. 1).

The general scheme of the training algorithms is as follows. At first, the initial values are selected: the number N of codebook vectors; the initial values of codebook vectors; the number of training epochs $\hat{e}$ (each input vector is passed to the network $\hat{e}$ times, then the number of training steps $t_{\max} = \hat{e} \times m$).

In the neural gas method, the Euclidean distances between the input vector X_l and each codebook vector $M_i = (m_{i1}, m_{i2}, \ldots, m_{in}),\ i = 1, \ldots, N$, are calculated. The distances $\|M_1 - X_l\|, \ldots, \|M_N - X_l\|$ are sorted in an ascending order. A neuron set $W_1, W_2, \ldots, W_N$ is obtained, where $W_k \in \{M_1, M_2, \ldots, M_N\},\ k = 1, \ldots, N$:

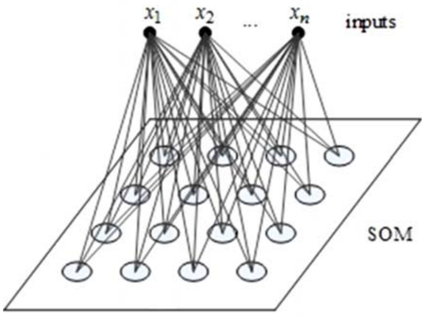


Fig. 1. Rectangular SOM structure.

$\|W_1 - X_l\| \leqslant, \ldots, \leqslant \|W_N - X_l\|$. The neuron W_1 is called a winner. The neuron $W_k,\ k = 1, \ldots, N$, is adapted according to the learning rule: $W_k(t+1) = W_k(t) + E(t)h_\lambda(X_l - W_k(t))$, where t is the order number of iterations, $E(t) = E_g(E_f/E_g)^{(t/t_{\max})},\ h_\lambda = e^{-(k-1)/\lambda(t)},\ \lambda(t) = \lambda_g(\lambda_f/\lambda_g)^{(t/t_{\max})}$. The values of the parameters $\lambda_g,\ \lambda_f,\ E_g,\ E_f$ are selected before learning.

In the SOM method, the Euclidean distances from the input vector X_l to each codebook vector $M_{ij} = (m_{ij}^1, m_{ij}^2, \ldots, m_{ij}^n),\ i = 1, \ldots, rows,\ j = 1, \ldots, cols$, are calculated, too. The vector $\hat{M}_c$ with the minimal Euclidean distance to X_l is designated as a winner, where $c = \arg\min_{i,j}\{\|X_l - M_{ij}\|\}$. The neuron (vector) M_{ij} is adapted according to the learning rule: $M_{ij}(t+1) = M_{ij}(t) + h_{ij}^c(t)(X_l - M_{ij}(t))$, where t is the order number of iterations, h_{ij}^c is the so-called neighbourhood function, $h_{ij}^c(t) \to 0$, as $t \to \infty$. For generality, notation M_i is used instead of M_{ij} below. Pseudo-codes of the training algorithms are presented in Figs. 2–3.

Then the networks are trained, the quantization error E_{QE} is calculated by the formula:

$$E_{\mathrm{QE}} = \frac{1}{m}\sum_{l=1}^{m} \|X_l - \hat{M}\|, \tag{1}$$

where $\hat{M}$ is a winner for the vector $X_l,\ \hat{M} = W_1$ in the neural gas method.

3. Visualization of Multidimensional Data

A large class of methods has been developed for multidimensional data visualization (Chen *et al.*, 2008; Dzemyda *et al.*, 2007). The visual presentation of the data allows us to see the data structure, clusters, outliers, and other properties of multidimensional data. The direct data visualization is a graphical presentation of the data set providing a quality understanding of the information contents in a natural and direct way. In the direct visualization methods (such as scatterplot matrices, parallel coordinates, Andrew's curves, Chernoff faces, stars, dimensional stacking, etc.), each feature of a multidimensional object is presented in a visual form (Hoffman and Grinstein, 2002). After the comprehensive

```
function NG_training $(X, M, t_{\max}, N, \lambda_g, \lambda_f, E_g, E_f)$
// input: $X$ – data set, $M$ – initial codebook, $t_{\max}$ – the number of training steps,
// $N$ – the number of neurons, $\lambda_g,\ \lambda_f,\ E_g,\ E_f$ – parameters (constants)
// output: $W$ – codebook
BEGIN
FOR $t = 0$ TO $t_{\max}$
  FOR $l = 1$ TO $m$ // the input vector $X_l$ is passed to the neural network
    FOR $i = 1$ TO $N$
      $\|M_i - X_l\| := \sqrt{\sum_{p=1}^{n}(m_{ip} - x_{lp})^2}$ // Euclidean distances are calculated
    END
    $\{W_1, W_2, \dots, W_N\} :=$ SORT_ASCENDING$(\|M_1 - X_l\|, \dots, \|M_N - X_l\|)$
      // here $W_k \in \{M_1, M_2, \dots, M_N\},\ k = 1, \dots, N,$
      // and $\|W_1 - X_l\| \leqslant, \dots, \leqslant \|W_N - X_l\|$
    $E(t) := E_g(E_f/E_g)^{(t/t_{\max})},\ \lambda(t) := \lambda_g(\lambda_f/\lambda_g)^{(t/t_{\max})}$
    FOR $k = 1$ TO $N$
      $h_\lambda := e^{-(k-1)/\lambda(t)},$
      $W_k(t+1) := W_k(t) + E(t)h_\lambda(X_l - W_k(t))$ // NG learning rule
    END
    END // the end of passing of the input vectors
END // the end of training
RETURN $W$
END
```

Fig. 2. Pseudo-code of the NG training algorithm.

```
function SOM_training $(X, M, \hat{e}, rows, cols)$
// input: $X$ – data set, $M$ – initial codebook, $\hat{e}$ – the number of training epochs,
// $rows,\ cols$ – the number of rows and columns
// output: $M$ – codebook
BEGIN
FOR $t = 1$ TO $\hat{e}$
  FOR $l = 1$ TO $m$ // the input vector $X_l$ is passed to the neural network
    FOR $i = 1$ TO $rows$
      FOR $j = 1$ TO $cols$
        $\|M_{ij} - X_l\| := \sqrt{\sum_{p=1}^{n}(m_{ij}^{p} - x_{lp})^2}$ // Euclidean distances are calculated
      END
    END
    $c := \arg\min_{i,j}\{\|X_l - M_{ij}\|\}$ // $\hat{M}_c$ – neuron-winner of vector $X_l$
    FOR $i = 1$ TO $rows$
      FOR $j = 1$ TO $cols$
        $M_{ij}(t+1) := M_{ij}(t) + h_{ij}^{c}(t)(X_l - M_{ij}(t))$// SOM learning rule
      END
    END
  END // the end of passing of the input vectors
END  // the end of training
RETURN $M$
END
```

Fig. 3. Pseudo-code of the SOM training algorithm.

analysis of these methods has been made, a conclusion is drawn: using these methods, it is complicated to comprehend the data structure, it is almost impossible, when large data sets or data of large dimensionality are analysed.

The results of visualization of the well known iris data taken from Asuncion and Newman (2007) by some direct visualization methods are presented in Fig. 4. Iris data set consists of 150 items: by 50 from each of three species of iris flowers (Setosa, Versicolor and Virginica). Four features of each flower were measured: x_1 is sepal length, x_2 is sepal width, x_3 is petal length, x_4 is sepal width. Four dimensional ($n = 4$) vectors

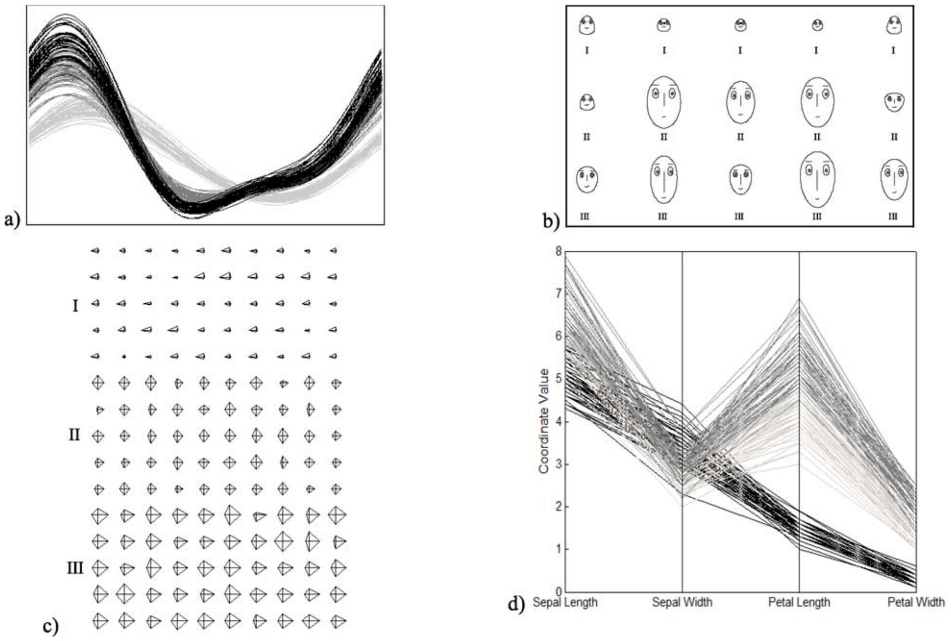


Fig. 4. Visualization of iris data by (a) Andrew's curves, (b) Chernoff faces (only five items of each iris species are presented), (c) stars, (d) parallel coordinates.

$X_1, X_2, \ldots, X_{150}$ are formed, where $X_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}),\ i = 1, \ldots, 150$. It is known that one class (Setosa) is linearly separable from the other two classes; the classes (Virginica and Versicolor) are not linearly separable from each other.

Another group of visualization methods is based on reduction of the dimensionality of data. There exist a lot of methods that can be used for reducing the dimensionality, and, particularly, for visualizing the n-dimensional vectors: principal component analysis (PCA; Jolliffe, 2002), multidimensional scaling (MDS; Borg and Groenen, 2005), locally linear embedding (LLE; Roweis and Soul, 2000), etc. The target of these methods is to represent the input data in a lower-dimensional space so that certain properties of the data set were preserved as faithfully as possible. If we have n-dimensional data set $X_1, X_2, \ldots, X_m$, where $X_i = (x_{i1}, x_{i2}, \ldots, x_{in}),\ i = 1, \ldots, m$, we desire to get d-dimensional data set $Y_1, Y_2, \ldots, Y_m$, where $Y_i = (y_{i1}, y_{i2}, \ldots, y_{id}),\ i = 1, \ldots, m$ and $d < n$. These methods can be used to visualize the data set, if a sufficiently small output dimensionality ($d = 2$ or $d = 3$) is chosen. In such cases, two or three dimensional vectors may be presented in a scatter plot. The result of visualization of the iris data set by the two principal components is presented in Fig. 5.

If some linear dependences between the features $x_1, x_2, \ldots, x_n$ exist, then the number of dimensions of data is decreased with small errors by the principal component analysis. However, in the general case, there exist some nonlinear dependences and the PCA cannot estimate them and this fact is a disadvantage of the method. The results of visualization, obtained by the PCA, depend on so-called outliers, because these points in the

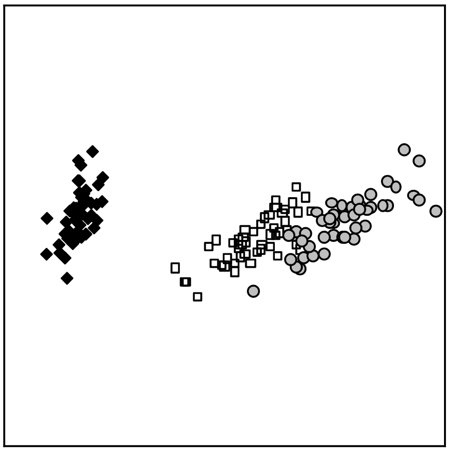


Fig. 5. Visualization of iris data by the principal component analysis.

n-dimensional space are distant from the other points, and this fact make a large influence to covariation matrix, as well as the principal components and the quality of mapping.

In this paper, one of the most popular methods of visualization, multidimensional scaling (Borg and Groenen, 2005) is used to visualize the codebook vectors obtained by a self-organizing map and neural gas. It is purposeful to visualize the codebook vectors obtained after a quantization in order to get more knowledge from the data set analysed. The multidimensional scaling is also superior to the principal component analysis in point of visualization, because not always two (or three) components are principal ones. It is taken into account all components of the visualized vectors (coordinates of points) in the multidimensional scaling. On the other hand, the principle component analysis is a linear method and the multidimensional scaling is a nonlinear one.

3.1. *Multidimensional Scaling*

Multidimensional scaling (MDS) refers to a group of the methods that are widely used for a dimensionality reduction and visualization of multidimensional data (Borg and Groenen, 2005). The starting point of the MDS is a matrix consisting of pairwise proximities (similarity or dissimilarity) of the data. In general, the proximities need not be distances in the mathematically strict sense.

The goal of multidimensional scaling (MDS) is to find lower-dimensional data Y_i, $i = 1, \dots, m$, such that the distances between the data in the lower-dimensional space were be as close to the original proximities as possible (Borg and Groenen, 2005). The objective function (stress) to be minimized can be written as

$$E_{\text{MDS}} = \sum_{i<j} w_{ij} \big(\delta(X_i, X_j) - d(Y_i, Y_j)\big)^2, \tag{2}$$

where $\delta(X_i, X_j)$ is the values of proximity between the n-dimensional data X_i and X_j, $d(Y_i, Y_j)$ is the distance (usually of the Euclidean metric) between the two-dimensional data Y_i and Y_j, $d(Y_i,\ Y_j) = \|Y_i - Y_j\|$; w_{ij} is the weight. If the proximity is the Euclidean distance, then $\delta(X_i, X_j) = d(X_i, X_j)$. If the neurons-winners obtained by the neural gas or the self-organizing map are mapped by MDS, X_i and X_j are changed to M_i and M_j in (2).

There exists a multitude of variants of MDS with different weights w_{ij} and optimisation algorithms: simple gradient descent method, conjugate gradient, quasi-Newton method, simulated annealing, combination of genetic algorithm, quasi-Newton's descent algorithm (Mathar and Žilinskas, 1993), two level minimization (Žilinskas and Žilinskas, 2007). Dimensionality of embedding space in multidimensional scaling is investigated in Žilinskas (2008), a relative MDS is researched in Bernatavičienė *et al.* (2007), etc. In this paper, we use the SMACOF (Scaling by MAjorization of a Complicated Function) algorithm for the stress function E_{MDS} (2) minimization, $w_{ij} = 1,\ \forall i, j$. It is one of the best optimization algorithms for this type of the minimization problem. This method is simple and powerful, because it guarantees a monotone convergence of the stress function (Borg and Groenen, 2005). This algorithm of MDS for finding of two-dimensional vectors can be summarized as follows:

1. Set the initial values of two dimensional vector Y, set $t = 0$.
2. Compute the value of the stress function $E_{\mathrm{MDS}}(Y(t))$; the two-dimensional vectors Y are set in $t = 0$.
3. Increase the iteration counter t by one.
4. Compute the *Guttman transform* $Y(t)$ by (3).
5. Compute $E_{\mathrm{MDS}}(Y(t))$ by (2).
6. If $E_{\mathrm{MDS}}(Y(t-1)) - E_{\mathrm{MDS}}(Y(t)) < \varepsilon$ or t is equal to maximum number of iterations, then stop (ε is a small positive constant), else go to Step 3.

Formula (3) is called the *Guttman transform*:

$$Y(t+1) = m^{-1} B(Y(t)) Y(t), \tag{3}$$

where $B(Y(t))$ has the elements

$$b_{ij} = \begin{cases} -\frac{d(X_i,X_j)}{d(Y_i,Y_j)}, & \text{for } i \neq j \text{ and } d(Y_i,Y_j) \neq 0 \\ 0, & \text{for } i \neq j \text{ and } d(Y_i,Y_j) = 0 \end{cases}, \quad b_{ii} = -\sum_{j=1,\ j\neq i}^{m} b_{ij}.$$

3.2. *Quantitative Criteria of Visualization*

When vectors are visualized, it is necessary to estimate the visualization quality. As one is desired to preserve proximity (for example, distance, topology, neighbourhood relationships, etc.) by the visualization methods, based on a dimensionality reduction, some numerical measures are used to estimate the preservation of the proximity. The problem of this estimation requires intensive researches (Bernatavičienė *et al.*, 2006; Estévez *et al.*, 2005; Goodhill and Sejnowski, 1996; Karbauskaitė and Dzemyda, 2009).

In this paper, we use two measures. First one is introduced by Konig (2000), applied in Estévez (2005), Karbauskaitė and Dzemyda (2009). Let us name it *Konig's measure*. The measure is a topology preservation measure. The second measure used is *Spearman's rho*. It is a statistical test for correlation between two rank-ordered scales. It yields a statement of the degree of interdependence of the scores of the two scales. *Spearman's rho* is used for estimation preservation of the inter-point distances when passing from n-dimensional space to d-dimensional space (Karbauskaitė *et al.*, 2007; Kurasova and Molytė, 2009b).

Konig's topology preserving measure is based on the assessment of rank order in n-dimensional and d-dimensional spaces. This measure has two control parameters – numbers of the nearest neighbours: μ and ν $(\mu < \nu)$. The neighbourhood is estimated by the Euclidean distances here.

Let us denote by:

- $X_{ij},\ j = 1,\dots,\mu,\mu$ is the nearest neighbours of the n-dimensional vector X_i, where the distances X_i and its neighbours satisfy the following inequality $\|X_i - X_{ij_1}\| < \|X_i - X_{ij_2}\|$ with $j_1 < j_2$;
- $Y_{ij},\ j = 1,\dots,\nu,\nu$ is the nearest neighbours of the d-dimensional vector Y_i;
- $r_X(i,j)$ a rank of the jth neighbour X_{ij} of the vector X_i, where the rank means the order number of X_{ij} in the data set analysed;
- $r_Y(i,j)$ a rank of the jth neighbour Y_{ij} of the vector Y_i, corresponding to X_i, where the rank means the order number of Y_{ij} in the data set analysed.

Konig's measure for the ith vector and the jth neighbour is calculated by (4):

$$E_{\mathrm{KM}}^{ij} = \begin{cases} 3, & \text{if } r_X(i,j) = r_Y(i,j), \\ 2, & \text{if } r_X(i,j) = r_Y(i,l),\ l \in (1,\dots,\mu),\ i \neq l, \\ 1, & \text{if } r_X(i,j) = r_Y(i,t),\ t \in (\mu+1,\dots,\nu),\ \mu < \nu, \\ 0, & \text{else.} \end{cases} \tag{4}$$

The general *Konig's measure* E_{KM} is calculated as follows:

$$E_{\mathrm{KM}} = \frac{1}{3\mu \times m} \sum_{i=1}^{\mu} \sum_{j=1}^{m} E_{\mathrm{KM}}^{ij}. \tag{5}$$

The range of E_{KM} is between 0 and 1, where 0 indicates a poor neighbourhood preservation, and 1 indicates a perfect one.

Spearman's rho is calculated by the formula:

$$\rho_{Sp} = 1 - \frac{6}{(m')^3 - m'} \sum_{k=1}^{m'} \left(r'_X(k) - r'_Y(k)\right)^2, \tag{6}$$

where r'_X and r'_Y are the ranks (order numbers) of pairwise distances calculated for the n-dimensional and d-dimensional data, respectively; $m' = m(m-1)/2$. As usual, $-1 \leqslant \rho_{Sp} \leqslant 1$. The best value of *Spearman's rho* is equal to one.

4. Combining Neural Gas and Self-Organizing Map with Multidimensional Scaling

After a large number of training steps, the neural gas network and the self-organizing map (SOM) have been organized and n-dimensional input vectors $X_1, X_2, \dots, X_m$ have been mapped – each input vector is related to the nearest neuron. Some neurons may remain unrelated with any vector of the set $X = \{X_1, \dots, X_m\}$, but there may occur neurons related with some input vectors. The neurons related with the input vectors are called neurons-winners. A number of neurons-winners r is usually less than that of all neurons N $(r \leqslant N)$.

In the case of the rectangular topology of the SOM, we can draw a table (grid) with cells corresponding to the neurons (Table 1). It can be considered as the location of n-dimensional points on the plane. The cells of the table (the order number of rows and columns) correspond to the location of the points on the plane. In Table 1, the numbers indicate the species (classes) of iris data. However, the table does not answer the question, how much the vectors of the neighbouring cells are close in the n-dimensional space. A natural idea arises to apply the distance-preserving method to an additional mapping of the neurons-winners in the SOM. MDS-type methods may be used for this purpose.

In the papers Bernatavičienė *et al.* (2006), Dzemyda and Kurasova (2006), some combinations of the SOM and MDS-type method are proposed and discussed. Neurons-winners obtained by the neural gas method may be visualized by MDS-type methods, too (Kurasova and Molytė, 2009a). Cross-entropy approach for a visualisation of the neurons-winners obtained by the neural gas is proposed in Estévez *et al.* (2005). There the method is compared with some other combinations. We have proposed a combination of the neural gas and self-organizing map with the multidimensional scaling on based SMACOF minimization. A scheme of a mapping (visualization) of the neurons-winners by the MDS is presented in Fig. 6.

Table 1

SOM table of iris data

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

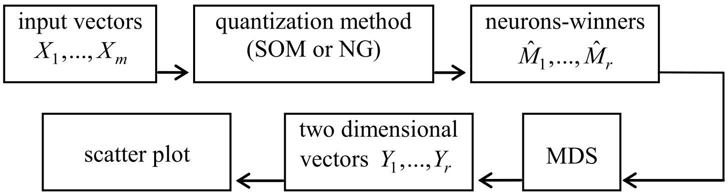


Fig. 6. Scheme of a visualization of the neurons-winners.

5. Experimental Results and Comparisons

5.1. *Data for Analysis*

Some data sets are used in the experimental investigations. Each data set has some specific characteristics. Short descriptions of the data sets are presented in Table 2. The c*hainlink*, *hepta*, and *target* data sets are taken from "Fundamental Clustering Problems Suite (FCPS)" (http://www.uni-marburg.de/fb12/datenbionik/data/), the *auto MPG* and *iris* data sets are taken from Asuncion and Newman (2007), and the last ones are generated by us.

5.2. *Numbers of Neurons-Winners*

The numbers of the neural gas (NG) and the self-organizing maps (SOM) are investigated. It is of interest to investigate whether more neurons become winners by the NG or by the SOM. The ratios between the number of neurons-winners and all the neurons of the NG and SOM are presented in Fig. 7. It is shown that the ratios of NG are larger than that of the SOM: about 80% of the NG neurons become winners. If the numbers of neurons are large, only about 50% of the SOM neurons become winners. This investigation shows that the SOM is more efficient than the NG for solving clustering problems.

Table 2

Data sets

No.	Name	m	n	Number of classes	Description
1	chainlink	1000	3	2	Two interlocking 3-dimensional rings, linear not separable
2	hepta	212	3	7	7 well separated clusters
3	target	770	2	6	Outliers
4	auto MPG	392	7	4	Linear not separable
5	iris	150	4	3	One class and a group of two other classes are linear separable, these two classes are not linear separable
6	rand_clust10	100	10	5	Random clusters
7	rand_clust5	100	5	5	Random clusters

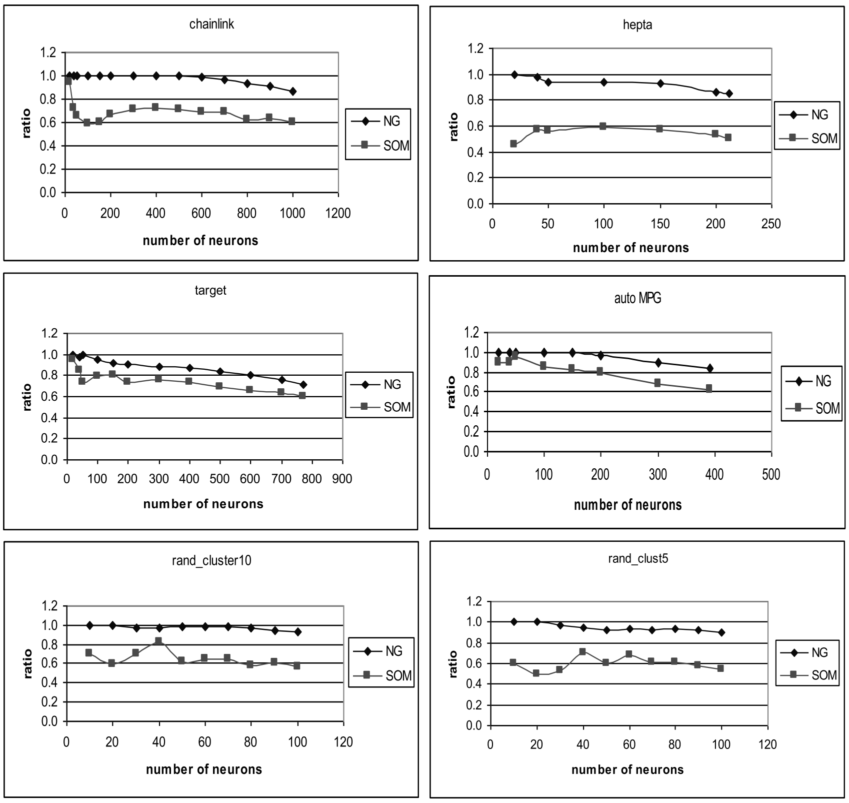


Fig. 7. The ratios between the number of neurons-winners and all the neurons of NG and SOM.

5.3. *Quality of Quantization*

At first, the input vectors $X_1, X_2, \dots, X_m$ are quantized by the neural gas and the self-organizing map. Quantization error E_{QE} (1) is calculated to estimate the quality of quantization. Quantization error shows the difference between the analysed vectors $X_1, X_2, \dots, X_m$ and quantised vectors (neurons-winners) $\hat{M}_1, \hat{M}_2, \dots, \hat{M}_r$, where r is the number of neurons-winners.

The dependence of the quantization error on the number of neurons-winners is presented in Fig. 8. The quantization error decreases then the number of neurons-winners is increasing. As we see in Fig. 8, the quantization errors of the NG are smaller significantly than that of the SOM then the number of neurons-winners is approximately equal. It means that the neural gas is more suitable for a vector quantization.

5.4. *Quality of Visualization*

After the quantization of the input vectors $X_1, X_2, \dots, X_m$ by the neural gas and the self-organizing map, neurons-winners $\hat{M}_1, \hat{M}_2, \dots, \hat{M}_r$ are visualised by the multidimensional scaling, and two-dimensional vectors $Y_1, Y_2, \dots, Y_r$ are obtained. Konig's

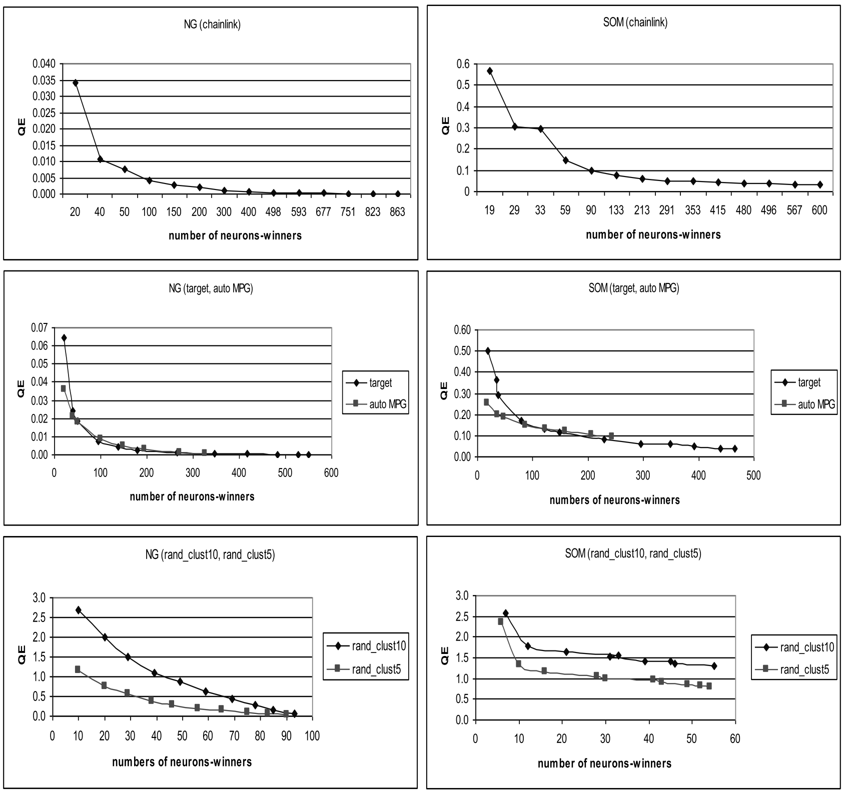


Fig. 8. The dependence of the quantization error (QE) on the number of neurons-winners of NG (*left*) and SOM (*right*).

topology preservation measure E_{KM} (5) and Spearman's rho ρ_{Sp} (6) are calculated to estimate the visualization quality. The number N of codebook vectors is selected so that the number of neurons-winners were equal to 100, 200, and 300 for the *chainlink* and *auto MPG* data sets, and to 50, 80, and 100 for the *rand_clust10* data set.

Since the results of SOM and NG depend on the initial values of codebook vectors, 40 experiments have been carried out for each input vector set with different initial values of codebook vectors. The values of the measures are calculated and averaged. The confidence intervals of the averages are also calculated (a probability is equal to 0.95).

When calculating Konig's topology preserving measure E_{KM}, it is necessary to select values of two parameters μ and ν. The parameter μ indicates a narrow round of neighbours, and the parameter ν indicates a wide round. In the experiments, $\mu = 4$, and ν is changed from 6 to 50. The averaged values of E_{KM} and the confidence intervals (CI) of the averages are presented in Fig. 9. We see that E_{KM} is larger, when the neurons-winners obtained by SOM are mapped, in all cases, except the *chainlink* data set, where the number of neurons-winners is equal to 100. We conclude that the topology is preserved precisely when the vectors-winners, obtained by SOM, are mapped by MDS.

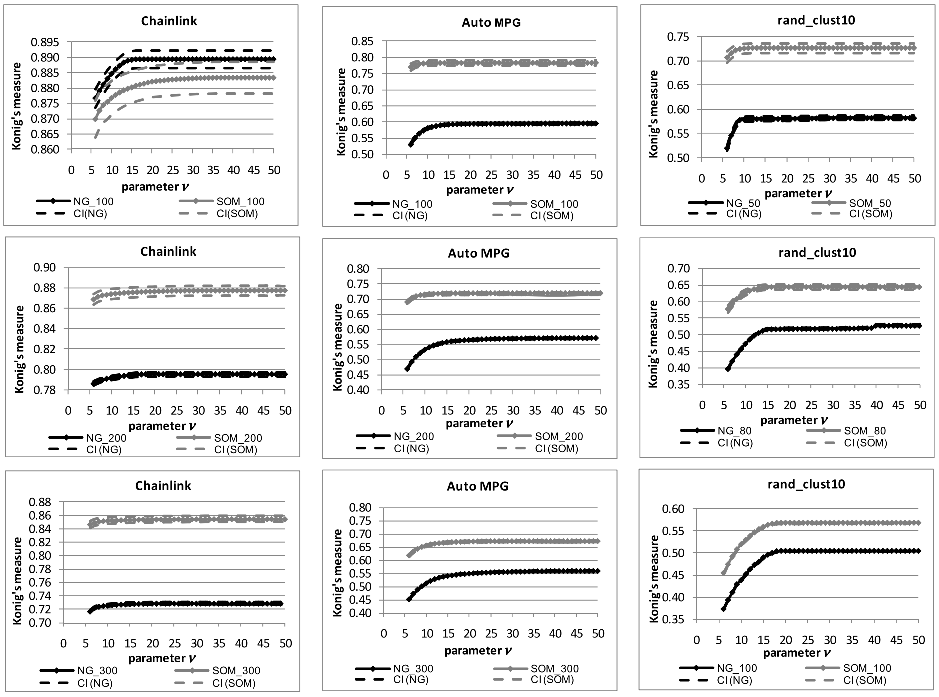


Fig. 9. The dependence of the averaged values of Konig's measure on parameter ν.

In an exceptional case, the confidence intervals are wide, they are overlapping, and therefore the results obtained are unreliable. When the number of neurons-winners is increasing, the confidence intervals are narrowing for all datasets. Naturally, for small values of the parameter ν, the values of E_{KM} are lower than that for higher ν, however starting a certain value of ν, the values of E_{KM} do not change at all or change but slightly.

The averaged values of Spearman's rho ρ_{Sp} and the confidence intervals (CI) of the averages are presented in Fig. 10. The values of Spearman's rho are higher, if the neurons-winners are obtained by NG for the *chainlink* data set, and by SOM for the *auto MPG* data set. The values of Spearman's rho are large enough ($\rho_{Sp} > 0.9$), which means that the mapping results are good in the sense of distance preserving, when passing from the n-dimensional space to a two-dimensional one. It is difficult to draw a conclusion on the mapping quality of the *rand_clust* dataset, because the values of Spearman's rho are varying, and the confidence intervals are wide and overlapping (see Fig. 10, right).

5.5. *Mapping Images*

When the neurons-winners that are n-dimensional vectors $\hat{M}_1, \hat{M}_2, \ldots, \hat{M}_r$ are mapped by multidimensional scaling, two-dimensional vectors $Y_1, Y_2, \ldots, Y_r$ are obtained. Here r is the number of neurons-winners. Two-dimensional vectors may be presented in a scatter plot. The mapping images of three artificial data sets (chainlink, target and hepta)

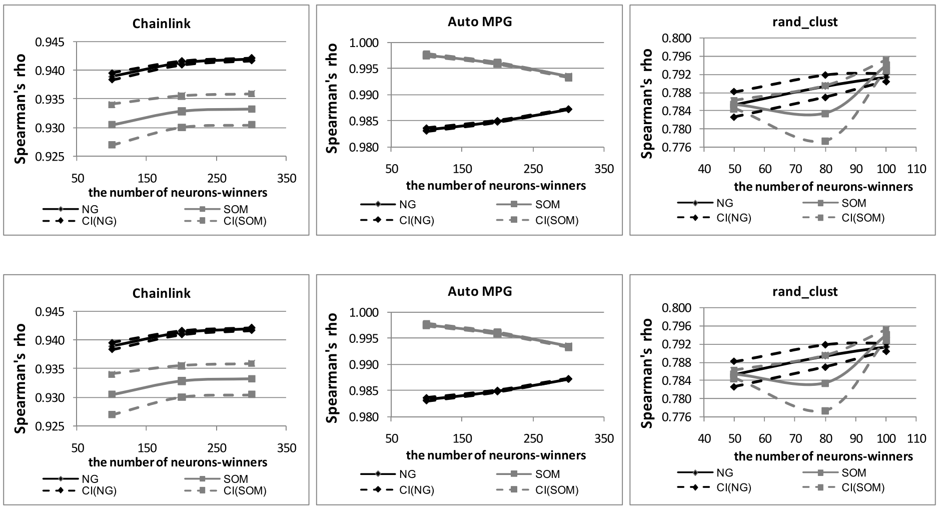


Fig. 10. The dependence of the averaged values of Spearman's of the number of neurons-winners.

are presented in Figs. 11–13. Numbers near points indicate order numbers of classes to which the points belong. Figures 11–13 show how the mapping images change when the number of neurons-winners is growing. The data structure is visible even when the number r of neurons-winners obtained by the NG is small enough. If the number r of neurons-winners obtained by the SOM is larger, the data structure is visible, as well.

In Fig. 14, the neurons-winners of the iris data set, obtained by the NG and the SOM, are visualized by the MDS. The points, corresponding to the items of the first species (Setosa), are marked by filled rhombi, the points, corresponding to the second species (Versicolor), are marked by filled squares and the points, corresponding to the third species (Virginica), are marked by filled circles. The points, corresponding to the neurons, that are the winners for both the second and the third species, are marked boxed circles. The quantization error of the SOM is much larger ($E_{\mathrm{QE}} = 0.3222$) than that of the NG ($E_{\mathrm{QE}} = 0.0379$). It means that the neurons-winners (quantized vectors) do not approximate the data analysed by the SOM precisely enough. We see that the points, obtained by the SOM, are clustered very much, but the points, obtained by the NG, are dispersed. The data structure is revealed better by the NG.

6. Conclusions

In this paper, two vector quantization methods (neural gas and self-organizing map) are used for reducing the number of the dataset items. The neurons-winners obtained are visualised (mapped) by multidimensional scaling. The number of neuron-winners, quantization and visualization qualities, and preservation of a data structure in the mapping image are investigated. Two criteria (Konig's topology preservation measure and Spearman's rho) are used to estimate the visualization quality. The dependence of the Konig's

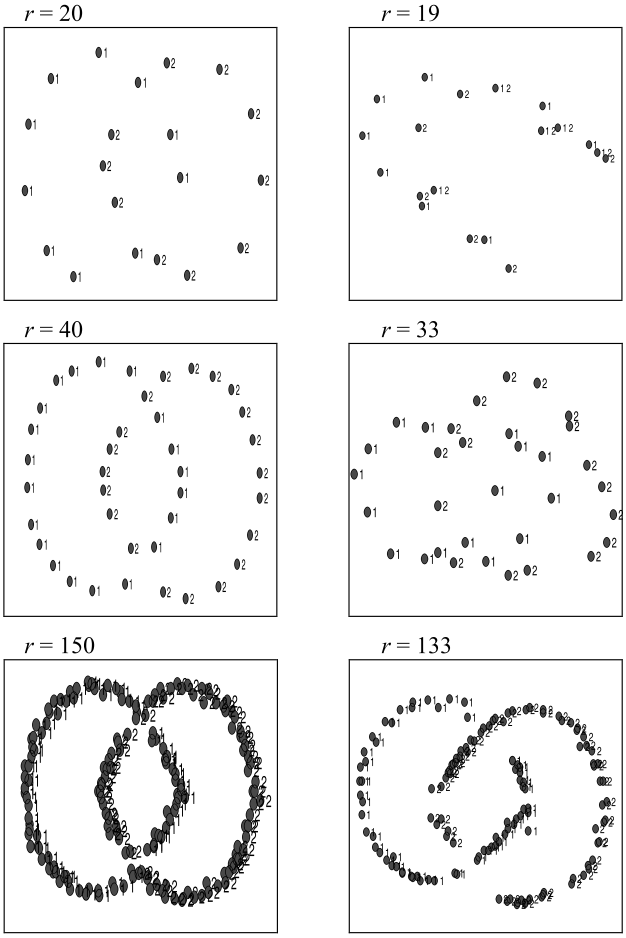
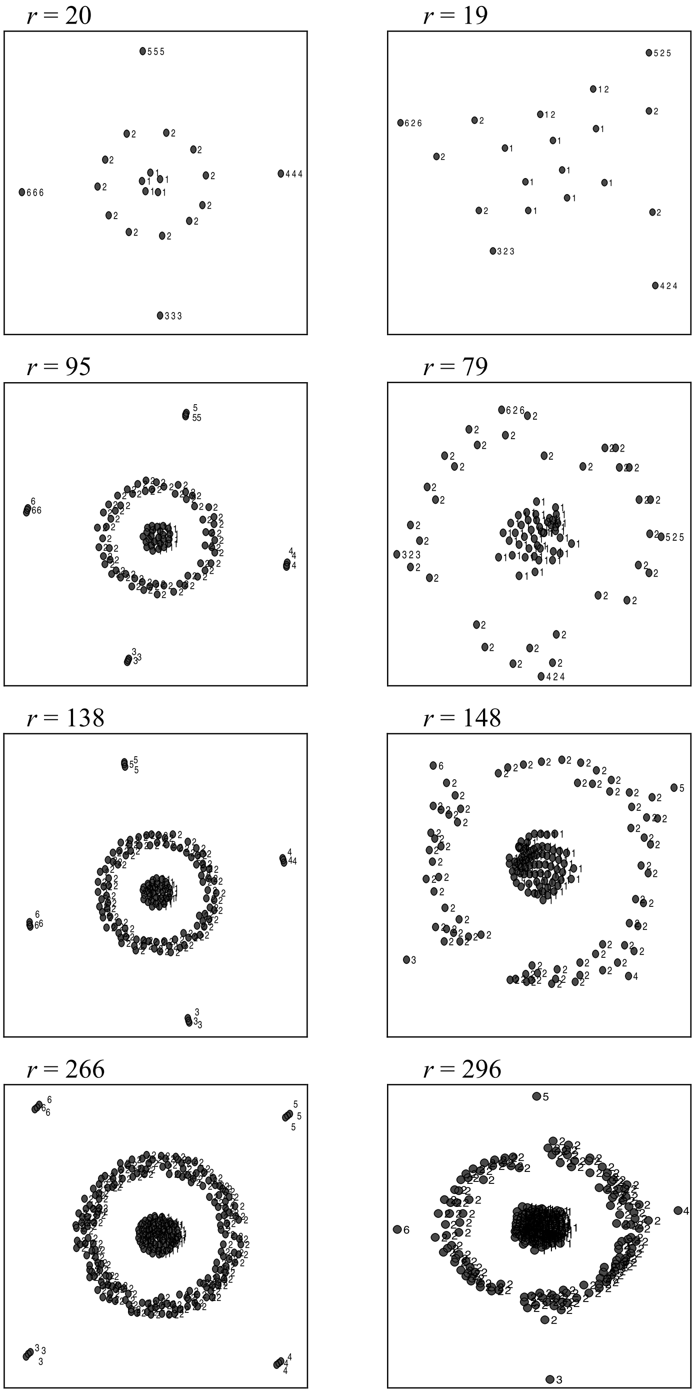


Fig. 11. Mapping images of *chainlink* data obtained by NG (*left*) and SOM (*right*).

measure on the parameter, which indicates a wide round of neighbours, has been investigated. The dependence of Spearman's rho on the number of neurons-winners has also been researched.

The experiments show that:

- The ratios between the number of neurons-winners and all the neurons of the neural gas (NG) are larger than that of the self-organizing map (SOM): about 80% of the NG neurons become winners. When the numbers of neurons are large, only about 50% of the SOM neurons become winners. It means that the SOM is more efficient then the NG for clustering.
- The quantization error decreases when the number of neurons is growing. The quantization errors of the NG are smaller than that of the SOM when the number of neurons-winners is approximately equal. It means that the neural gas is more suitable for vector quantization.



$r = 20$

$r = 19$

$r = 95$

$r = 79$

$r = 138$

$r = 148$

$r = 266$

$r = 296$

Fig. 12. Mapping images of *target* data obtained by NG (*left*) and SOM (*right*).

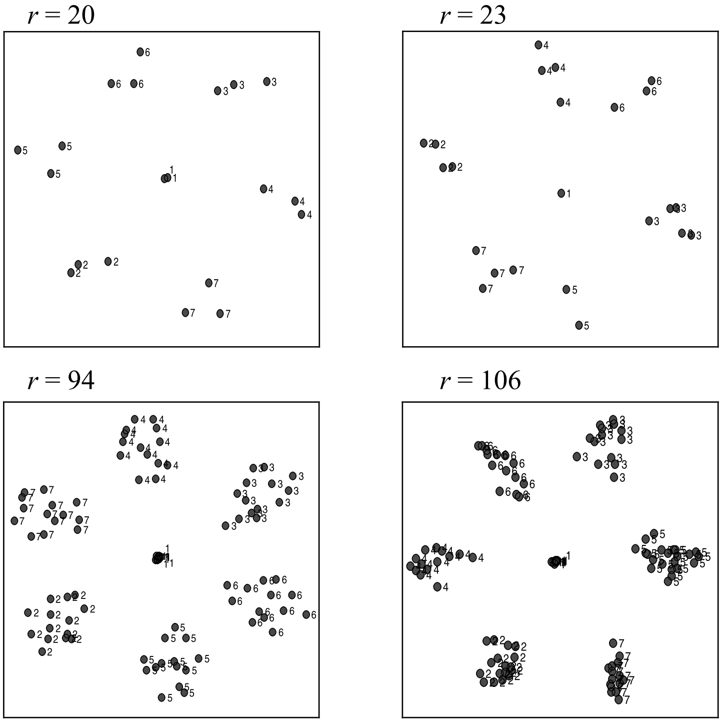


Fig. 13. Mapping images of *hepta* data obtained by NG (*left*) and SOM (*right*).

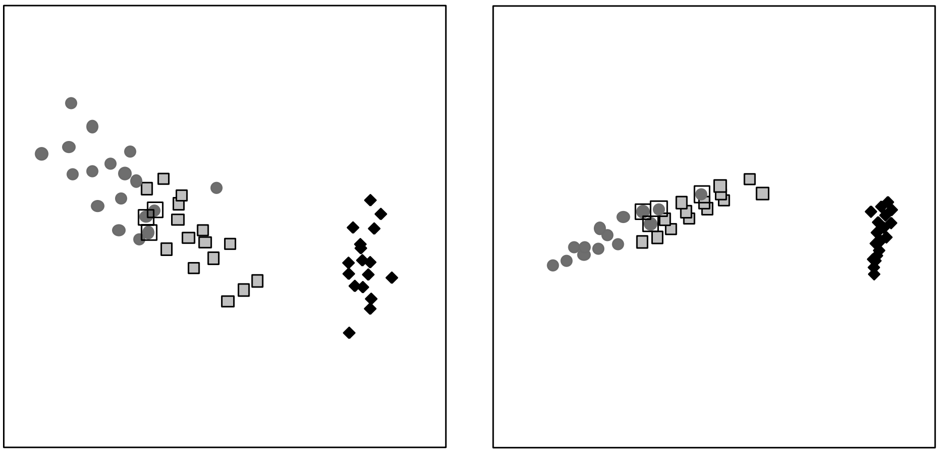


Fig. 14. Mapping images of *iris* data obtained by NG (*left*) ($E_{\mathrm{QE}} = 0.0379$) and SOM (*right*) ($E_{\mathrm{QE}} = 0.3222$).

- The topology is preserved more precisely in the sense of a Konig's measure, if the vectors-winners obtained by SOM are mapped by MDS. It means that the SOM preserves neighbourhood better.
- When the visualization quality is estimated by Spearman's rho, better results are obtained by NG for the *chainlink* dataset, and by SOM for the *auto MPG* dataset.
- The data structure is visible in the mapping image even when the number of neurons-winners of the NG is small enough. If the number of neurons-winners of the SOM is larger, the data structure is visible, as well.

Acknowledgement. This research was funded by a grant (No. COST-05/2010) from the Research Council of Lithuania and from the European Science Foundation through COST Action IC0702 "Combining Soft Computing Techniques and Statistical Methods to Improve Data Analysis Solutions (IC0702)". The authors' research was performed in cooperation with Vilnius University and Vilnius Pedagogical University.

References

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

Bernatavičienė, J., Dzemyda, G., Kurasova, O., Marcinkevičius, V. (2006). Optimal decisions in combining the SOM with nonlinear projection methods. *European Journal of Operational Research*, 173, 729–745.

Bernatavičienė, J., Dzemyda, G., Marcinkevičius, V. (2007). Conditions for optimal efficiency of relative MDS. *Informatica*, 18(2), 187–202.

Borg, I., Groenen, P. (2005). *Modern Multidimensional Scaling*. Springer, New York.

Chen, C., Hardle, W., Unwin, A. (2008). *Handbook of Data Visualization*. Springer, Berlin.

Cios, K.J., Pedrycz, W., Swiniarski, R.W., Kurgan, L.A. (2007). *Data Mining, a Knowledge Discovery Approach*. Springer, New York.

Dzemyda, G., Kurasova, O., Medvedev, V. (2007). Dimension reduction and data visualization using neural networks. In: Maglogiannis, I., Karpouzis, K., Wallace, M., Soldatos, J. (Eds.), *Emerging Artificial Intelligence Applications in Computer Engineering. Real World AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies. Frontiers in Artificial Intelligence and Applications*, Vol. 160. IOS Press, Amsterdam, pp. 25–49.

Dzemyda, G., Kurasova, O. (2006). Heuristic approach for minimizing the projection error in the integrated mapping. *European Journal of Operational Research*, 171(3), 859–878.

Dunham, M.H. (2003). *Data Mining Introductory and Advanced Topics*. Pearson Education, Prentice/Hall, Upper Saddle River/New York.

Estévez, P.A., Figueroa, C.J., Saito, K. (2005). Cross-entropy embedding of high-dimensional data using the neural gas model. *Neural Netwoks*, 18, 727–737.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17, 37–54.

Goodhill, G.J., Sejnowski, T. (1996). Quantifying neighbourhood preservation in topographic mappings. In: *Proceedings of the 3rd Joint Symposium on Neural Computation*. University of California, Los Angeles, pp. 61–82.

Hoffman, P.E., Grinstein, G.G. (2002). *A Survey of visualizations for high-dimensional data mining*. In: Fayyad, U., Grinstein, G. G., Wierse, A. (Eds.), *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, San Francisco.

Jolliffe I.T. (2002). *Principal Component Analysis*. 2nd edn., Springer Series in Statistics, Springer, New York.

Karbauskaitė, R., Dzemyda, G. (2009). Topology preservation measures in the visualization of manifold-type multidimensional data. *Informatica*, 20(2), 235–253.

Karbauskaitė, R., Kurasova, O., Dzemyda, G. (2007). Selection of the number of neighbours of each data point for the locally linear embedding algorithm. *Information Technology and Control*, 36(4), 359–364.

Kohonen, T. (2001). *Self-Organizing Maps*, 3rd ed., Springer Series in Information Science, Vol. 30. Springer, Berlin.

Konig, A. (2000). Interactive visualization and analysis of hierarchical neural projections for data mining. *IEEE Transactions on Neural Networks*, 11(3), 615–624.

Kurasova, O., Molytė, A. (2009a). Investigation of the quality of mapping vectors obtained by quantization methods. In: *Proceedings of XIII International Conference on Applied Stochastic Models and Data Analysis, ASMDA 2009*. Vilnius, Technika, pp. 269–273.

Kurasova, O., Molytė, A. (2009b). Combination of vector quantization and visualization. In: Perner, P. (Ed.), *Machine Learning and Data Mining in Pattern Recognition – MLDM 2009. Lecture Notes in Artificial Intelligence*, Vol. 5632. Springer, Heidelberg, pp. 29–43.

Martinetz, T.M., Schulten, K.J. (1991). A neural-gas network learns topologies. In: Kohonen, T., Mäkisara, K., Simula, O., Kangas, J. (Eds.), *Artificial Neural Networks*. Elsevier, Amsterdam, pp. 397–402.

Mathar, R., Žilinskas, A. (1993). On global optimization in two-dimensional scaling. *Acta Aplicandae Mathematicae*, 33, 109–118.

Roweis, S.T., Saul, L.K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500), 2323–2326.

Šutienė, K., Makackas, D., Pranevičius, H. (2010). Multistage K-means clustering for scenario tree construction. *Informatica* 21(1), 123–138.

Žilinskas, A., Žilinskas, J. (2007). Two level minimization in multidimensional scaling. *Journal of Global Optimization*, 38(4), 581–596.

Žilinskas, J. (2008). On dimensionality of embedding space in multidimensional scaling. *Informatica*, 19(3), 447–460.

O. Kurasova is a senior researcher of System Analysis Department at Institute of Mathematics and Informatics of Vilnius University as well as an associate professor at Information Technologies Department of Vilnius Pedagogical University. She received bachelor's degree in mathematics (1999) and master's degree in informatics (2001) from Vilnius Pedagogical University, PhD in informatics from Vytautas Magnus University and Institute of Mathematics and Informatics (2005). She is the author (or a co-author) of 40 scientific publications. Her research interests include data mining methods, artificial neural networks, multidimensional data visualization, dimensionality reduction, multi-criteria optimization.

A. Molytė is a PhD student of System Analysis Department at Institute of Mathematics and Informatics of Vilnius University. She received bachelor's degree in mathematics (1999) and master's degree in informatics (2001) from Vilnius Pedagogical University. Her research interests include artificial neural networks, dimensionality reduction and multidimensional data visualization.

Vektorių, gautų neuroninių dujų ir saviorganizuojančių žemėlapių metodais, vizualizavimo ir kvantavimo kokybės tyrimas

Olga KURASOVA, Alma MOLYTĖ

Šiame straipsnyje nagrinėjama vektorių kvantavimo, taikant saviorganizuojančius žemėlapius ir neuroninių dujų metodą, kokybė, o taip pat vektorių, gautų šiais kvantavimo metodais, vizualizavimo kokybė. Vektoriams vizualizuoti taikomas daugiamačių skalių metodas. Kvantavimo kokybė yra vertinama pagal kvantavimo paklaidą. Vizualizavimo kokybei įvertinti naudojami du panašumų išsaugojimo matai (Konigo topologijos išsaugojimo matas ir Spirmano koreliacijos koeficientas). Straipsnyje pateikiami kelių žinomų struktūrų duomenų aibių vizualizavimo rezultatai, kai pradžioje vektoriai yra kvantuojami, o paskui vizualizuojami tik kvantuoti vektoriai. Straipsnyje atlikta dviejų kvantavimo metodų lyginamoji analizė ir padarytos išvados apie jų jungimą su daugiamatėmis skalėmis.