ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2011, Vol.40, No.1

INTEGRATION OF THE SELF-ORGANIZING MAP AND NEURAL GAS WITH MULTIDIMENSIONAL SCALING

Olga Kurasova, Alma Molytė

Vilnius University, Institute of Mathematics and Informatics
Akademijos Str. 4, LT-08663, Vilnius, Lithuania
e-mail: Olga.Kurasova@mii.vu.lt, Alma.Molyte@gmail.com

crossref http://dx.doi.org/10.5755/j01.itc.40.1.188

Abstract. In the paper, two combinations (consecutive and integrated) of vector quantization methods (self-organizing map and neural gas) and multidimensional scaling (MDS) have been investigated and compared. The vector quantization is used to reduce the number of dataset items. The dataset with a smaller number of items is analyzed by multidimensional scaling in order to reduce the number of features of data (dimensionality of space) and to map them onto the plane, i.e., to visualize. Some ways of the initialization (at random, on a line, by PCs and by variances) of two-dimensional vectors in MDS have been investigated. Two ways of assignment of two-dimensional vectors in the integrated combinations of MDS and vector quantization methods have been examined, too.

Keywords: self-organizing map (SOM), neural gas (NG), multidimensional scaling (MDS), initialization.

1. Introduction

The analysis of multidimensional data $X_1, X_2,..., X_m$ is an important part of data analysis; here m is the number of data items. All items are described by the same set of features $x_1, x_2,..., x_n$; here n is the number of features. If the values of features are numerical ones, $X_1, X_2,..., X_m$ are points in the n-dimensional space R^n, where n defines the dimensionality of the space. For short, these points are called n-dimensional points. They can be conceived as position vectors (bound to the origin $(0, 0,\ldots, 0)$). The values $x_{i1}, x_{i2},..., x_{in}$ of features are the components of the vector (coordinates of point) X_i, $i=1,...,m$. We have a matrix of data $\boldsymbol{X} = \{ X_1, X_2,..., X_m\} = \{ x_{ij}, i = 1,..., m, j = 1,..., n\}$.

The numbers m and n can be large. It is purposeful to reduce these numbers in order to facilitate the data analysis and exploration, and interpretation of the results.

Two main problems are solved in this paper: (1) reducing the number of dataset items and their dimensionality and (2) investigation of a dependence of the multidimensional scaling results on the initial states.

2. Reducing the Number of Data Items and their Dimensionality

A lot of methods of different nature have been developed for reducing the number of data items and their dimensionality. Vector quantization can be used to reduce the number m of data items. Multidimensional scaling (MDS) as a dimensionality reduction method can be used to reduce n. MDS is useful to map data onto a plane, i.e., to visualize data, if the dimensionality is reduced to 2. The visualization allows us to comprehend data and processes [1, 2, 3, 4, 5, 6]. We have proposed to combine these two groups of methods in order to reduce the number of items and dimensionality [2, 7, 8, 9, 10, 11].

2.1. Vector Quantization: Self-organizing Map and Neural Gas

The objective of vector quantization (VQ) for a dataset $\boldsymbol{X}$ is to discover the optimal codebook, containing a predetermined number s of codebook (reference, prototype) vectors $M_i \in R^n$, $i=1,...,s$, which guarantees minimization of the chosen distortion metric (usually Euclidean) for all the vectors from $\boldsymbol{X}$. Each codebook vector has an associated index used for referencing. Thus, the aim of quantization is to change the vectors X_l, $l=1,...,m$, so that the new vectors M_i, $i=1,...,s$, $s \ll m$, represent the properties of the vectors X_l. Vector quantization is used for data clustering and compression. VQ can deal with missing data. In the clustering case, the codebook vectors are representatives of clusters. In the paper, we use two methods for vector quantization based on neural networks: self-organizing map (SOM) [12] and neural

gas (NG) [13]. Here the neurons correspond to the codebook vectors.

The self-organizing map (SOM) is a class of neural networks that are trained in an unsupervised manner using a competitive learning [12]. The neural gas is a biologically inspired adaptive algorithm [13]. 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. A codebook M is an array of vectors. The dimensionality of the vectors is such as that of the analyzed vectors X_l, $l=1,...,m$, i.e., equal to n. The array $M=\{M_1,M_2,...,M_s\}$ is one-dimensional in NG, $M_i \in R^n$, $i=1,...,s$, s is the number of codebook vectors. The rectangular SOM is a two-dimensional array (grid) of neurons $M=\{M_{ij}, i=1,\ldots, rows, j=1, ..., cols\}$, where $M_{ij} \in R^n$, $rows$ is the number of rows of the grid, $cols$ is the number of columns of the grid, and the total number of neurons is $s=rows\times cols$.

At first of the training algorithms, the initial values are selected: the number s of codebook vectors; the initial values of codebook vector components; the number of training epochs $\hat{e}$ (each analyzed vector is passed to the network $\hat{e}$ times, then the number of training steps $t_{\max}=\hat{e}\times m$).

In NG, the Euclidean distances between the input vector X_l and each codebook vector (neuron) M_i = 1, ..., s, are computed. The distances are sorted in an ascending order. A neuron set $W_1,W_2,...,W_s$ is obtained, where $W_k \in \{M_1,M_2,...,M_s\}$, $k=1,...,s$, and $\|X_l-W_1\| \le,...,\le \|X_l-W_s\|$. The neuron W_1 is called a winner. The neuron W_k, $k=1,...,s$, 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 λ_g, λ_f, E_g, E_f are predetermined.

In SOM, the Euclidean distances from the input vector X_l to each codebook vector M_{ij}, $i=1,\ldots,rows$, $j=1,\ldots,cols$, are also computed. The vector (neuron) $\hat{M}_c$ with the minimal Euclidean distance to X_l is designated as a winner, where c is a pair of indices, i.e., $c=\arg\min_{i,j}\{\| X_l-M_{ij}\|\}$. The neuron 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 a neighbourhood function, $h_{ij}^c(t)\to 0$, as $t\to\infty$. Some variants of h_{ij}^c are possible. We use $h_{ij}^c=\alpha/(\alpha\eta_{ij}^c+1)$, $\alpha=\max((\hat{e}+1-e')/\hat{e},0.01)$; η_{ij}^c is the neighbourhood order in the grid between the neurons M_{ij} and $\hat{M}_c$; $\hat{e}$ is the number of training epochs, e' is the order number of a current epoch $(e'\in\{1,...,\hat{e}\})$. The vector M_{ij} is recomputed if $\eta_{ij}^c \le \max[\alpha\max(rows,cols),1]$. For generality, notation M_i is used instead of M_{ij} below.

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

$$E_{QE}=\frac{1}{m}\sum_{l=1}^{m}\left\|X_l-\hat{M}_{c(l)}\right\|, \quad (1)$$

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

2.2. Multidimensional Scaling

The target of dimensionality reduction methods is to represent the input data in a lower-dimensional space so that certain properties of the dataset were preserved as faithfully as possible. If we have a dataset $\boldsymbol{X}=\{X_1,X_2,...,X_m\}$ in the n-dimensional space, where $X_i=(x_{i1},x_{i2},...,x_{in})$, $i=1,...,m$, we desire to get a dataset $\boldsymbol{Y}=\{Y_1,Y_2,...,Y_m\}$ in d-dimensional space, where $Y_i=(y_{i1},y_{i2},...,y_{id})$, $i=1,...,m$ and $d<n$. If a sufficiently small output dimensionality $d=2$ or $d=3$ is chosen, two or three dimensional vectors obtained may be presented in a scatter plot.

Multidimensional scaling (MDS) refers to a group of methods that are widely used for dimensionality reduction and visualization of multidimensional data [14]. The goal of multidimensional scaling (MDS) is to find lower-dimensional data Y_i, $i=1,...,m$, such that the distances between the data in the lower-dimensional space were as close to the original proximities (similarity or dissimilarity) as possible. The stress E_{MDS} must be minimized.

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

where w_{ij} is a weight; $\delta(X_i,X_j)$ is the value of proximity between the n-dimensional data X_i and X_j, $d(Y_i,Y_j)$ is the distance (usually, Euclidean) between the two-dimensional data Y_i and Y_j, $d(Y_i,Y_j)=\| Y_i-Y_j\|$. If the proximity is the Euclidean distance, then $\delta(X_i,X_j)=d(X_i,X_j)$. There exist a multitude of variants of MDS with different weights w_{ij} and

optimisation algorithms [15, 16, 17, 18], etc. In this paper, we use the SMACOF (Scaling by MAjorization of a COmplicated Function) algorithm for stress E_{MDS} (2) minimization, $w_{ij} = 1, \forall i, j$. This method guarantees a monotone convergence of the stress function [14]. The MDS based on SMACOF is as follows:

1. Set the initial values of two-dimensional vectors from the set $\boldsymbol{Y} = \{Y_1, Y_2, ..., Y_m\}$. Set $t = 0$.
2. Compute the value of the stress $E_{MDS}(\boldsymbol{Y}(t))$ by (2).
3. Increase the iteration counter t by one.
4. Compute $\boldsymbol{Y}(t)$ by the formula $\boldsymbol{Y}(t+1) = m^{-1}B(\boldsymbol{Y}(t))\boldsymbol{Y}(t)$, where $B(\boldsymbol{Y}(t))$ has the elements $b_{ij} = -d(X_i, X_j)/d(Y_i, Y_j), i \neq j,$ if $d(Y_i, Y_j) \neq 0$; and $b_{ij} = 0$, if $d(Y_i, Y_j) = 0$; $b_{ii} = -\sum_{j=1,\, j\neq i}^{m} b_{ij}$.
5. Compute $E_{MDS}(\boldsymbol{Y}(t))$ by (2). If $E_{MDS}(\mathbf{Y}(t-1)) - E_{MDS}(\boldsymbol{Y}(t)) < \varepsilon$ or t is equal to the maximum number of iterations, then stop (ε is a small positive constant), else go to Step 3.

A way of comparing the mapping results is to calculate and estimate the value of the stress function E_{MDS} (2). The normalized stress

$$\hat{E}_{MDS} = \sqrt{\frac{\sum_{i<j}(d(X_i, X_j) - d(Y_i, Y_j))^2}{\sum_{i<j}(d(X_i, X_j))^2}} \quad (3)$$

is used instead of E_{MDS}, because the inclusion of the normalized parameter gives a clear interpretation of the mapping quality that does not depend on the scale in an *n*-dimensional space. The computational complexity of MDS based on SMACOF is $O(m^2)$. If we analyze a large dataset, MDS is time consuming. Many techniques of reducing the computational time are proposed. Some ways are based on pre-processing: at first, the number *m* of dataset items is reduced and then a smaller data set is analyzed by MDS. The reduction of *m* can be done by clustering or vector quantization methods.

3. Integration of Vector Quantization and Visualization

After training the NG or SOM network, each input vector X_i, $i = 1, ..., m$, from $\boldsymbol{X}$ is related to the nearest neuron, called a neuron-winner. Some neurons may remain unrelated with any vector of the set $\boldsymbol{X}$, but there may occur neurons related with some input vectors. So, the neurons-winners represent some input vectors, and the number *r* of neurons-winners is smaller than that of input vectors ($r < m$). Thus, the number *m* of data items is reduced. A smaller dataset can be used by MDS and the time is saved. The consecutive combination of vector quantization methods and the multidimensional scaling (Figure 1) have been investigated in [7, 8, 9, 10, 11, 19].

So, the reason to use the combination is a desire to decrease the computation time without losing the quality of mapping (visualization).

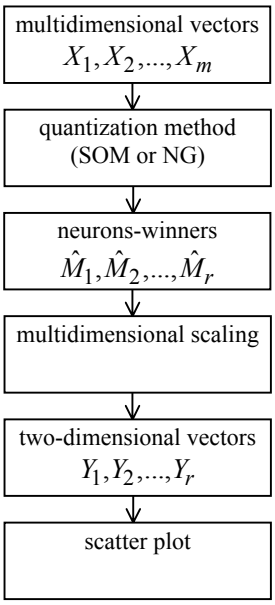

Figure 1. Scheme of a visualization of the neurons-winners (consecutive combination)

Another reason is based on improving the SOM visualization. As it is known, SOM itself has a visual presentation, e.g., u-matrix representation [12]. However, the SOM table does not answer the question, how much the vectors of the neighbouring cells are close in the *n*-dimensional space. It is reasonable to apply the distance-preserving method, like MDS, to an additional mapping of the neurons-winners in the SOM. A question arises: when the usage of MDS only is purposeful, and when its combination with vector quantization.

The computing time of MDS only, when all the items of the ellipsoidal dataset [20] ($m = 1338$, $n = 100$) have been analyzed is presented in Figure 2 (black line). The SOM learning has been repeated for several times with various numbers *s* of neurons. Various numbers *r* of neurons-winners have been obtained. The dependence of the SOM learning time on the number *r* of neurons-winners (blue curve), as well as of MDS on the number *r* of neurons-winners, when only they are analyzed by MDS (red curve), and the total time of the SOM and MDS combination (magenta curve) are presented in Figure 2. We see that if the number *r* of neurons-winners is smaller than 500, it is worth to use the combination in order to save the computational time comparing with MDS only. If NG is used instead of SOM, the similar results are obtained.

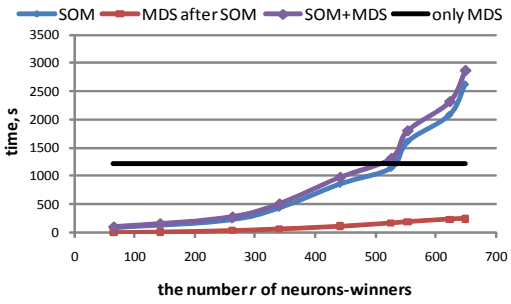

Figure 2. The computational time of MDS only and its combination with SOM

The visualization results of the ellipsoidal dataset when all data items ($m = 1338$) are mapped by MDS and only 259 neurons-winners ($r = 259$) of SOM are mapped by MDS are presented in Figure 3. We see that reduction of the number of data items does not aggravate the quality of visualization, however the computing time is saved essentially.

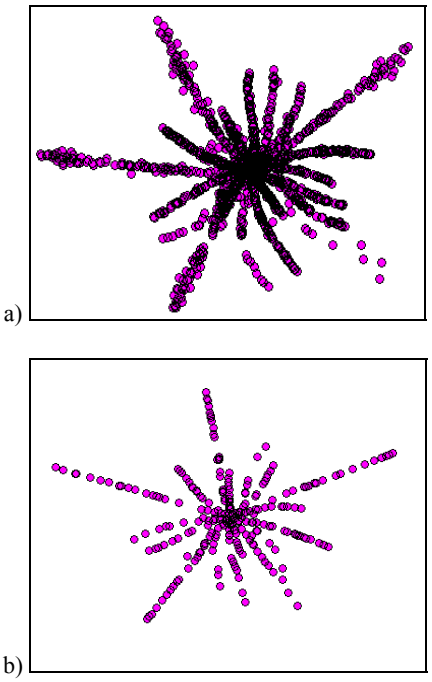

Figure 3. Mapping of an ellipsoidal dataset: (a) all data items are mapped by MDS, (b) only 259 neurons-winners of SOM are mapped by MDS

3.1. The Integrated Combination

Note that, if the MDS stress E_{MDS} is minimized in some iterative way, it is important to select the proper initial values of d-dimensional vectors $Y_1, Y_2, ..., Y_m$ (in our case, $d = 2$). The dependence of the MDS results on the initial values of these vectors remains a topical problem. We have proposed and investigated the integrated combination of SOM and MDS in [7, 8] as a new way of initialization of two-dimensional vectors. In this paper, we propose to use NG instead of SOM.

The idea of the integrated combination is as follows: n-dimensional vectors $X_1, X_2, ..., X_m$ are analyzed by using the MDS method, taking into account the process of SOM or NG training. Thus, the integrated combination consists of two parts: (1) SOM or NG training and (2) computing the two-dimensional points, corresponding to the neurons-winners of SOM or NG, by the MDS method. These two parts are performed alternately.

At first, some notation and definitions are introduced:

- Let the training set consist of n-dimensional vectors $X_1, X_2, ..., X_m$, ($X_i = (x_{i1}, x_{i2}, ..., x_{in})$, $i = 1, ..., m$). We need to get the two-dimensional vectors, called projections, $Y_1, Y_2, ..., Y_m$, ($Y_i = (y_{i1}, y_{i2})$, $i = 1, ..., m$)).
- The neural network (SOM or NG) is trained using $\hat{e}$ training epochs.
- All the $\hat{e}$ epochs are divided into equal training parts – blocks. Before starting the training of the neural network, we choose the number of blocks γ into which the training process will be divided. Each block contains v training epochs ($\hat{e} = v\gamma$). Denote by q a block of the training process consisting of v epochs ($q = 1, ..., \gamma$).
- Denote neurons-winners, obtained by the qth block of the training process, as $M_1^{(q)}, M_2^{(q)}, ..., M_{r_q}^{(q)}$ and two-dimensional projections of these neurons-winners, calculated by the MDS method, as $Y_1^{(q)}, Y_2^{(q)}, ..., Y_{r_q}^{(q)}$ | ($Y_i^{(q)} = (y_{i1}^{(q)}, y_{i2}^{(q)})$, $i = 1, ..., r_q$). Note that the number of neurons-winners r_q will be smaller than or equal to m.

We suggest the following way of integrating the SOM or NG and MDS methods:

Step 1: **network training begins** ($q = 1$). After the first v training epochs, the training is stopped temporally. The neurons-winners $M_1^{(1)}, M_2^{(1)}, ..., M_{r_1}^{(1)}$, obtained after the first block ($q = 1$) of the training process, are analyzed by MDS. The initial coordinates of two-dimensional vectors $Y_i^{(0)} = (y_{i1}^{(0)}, y_{i2}^{(0)})$, $i = 1, ..., r_1$, must be set for MDS. There are some possible ways. The initial coordinates $(y_{i1}^{(0)}, y_{i2}^{(0)})$ can be set:

(1) at random in the interval (0, 1);

(2) on a line: $y_{i1}^{(0)} = i + 1/3$, $y_{i2}^{(0)} = i + 2/3$;

(3) according to two largest principal components (PCs);

(4) according to the components whose variances are the largest ones.

After MDS has been performed, the two-dimensional projections $Y_1^{(1)}, Y_2^{(1)}, ..., Y_{r_1}^{(1)}$ of neurons-winners are obtained.

Steps from 2 **to** γ: **network training is continued** ($q = 2,..., \gamma$). The neurons-winners obtained after each qth block of the training process are analyzed by using MDS. The initial coordinates of two-dimensional vectors $Y_1^{(q)}, Y_2^{(q)}, ..., Y_{r_q}^{(q)}$ are selected for MDS taking into account the result of the (q–1) block. Note that $r_q \neq r_{q-1}$ in general. The way of selecting the initial coordinates is presented below. We must determine the initial coordinates of each two-dimensional vector $Y_i^{(q)}$ correspondent to the neuron-winner $M_i^{(q)}$, $i = 1,..., r_q$. The sequence of steps is as follows:

- Determine vectors from $\{X_1, X_2, ..., X_m\}$ that are related with $M_i^{(q)}$. Note that some vectors from $\{X_1, X_2, ..., X_m\}$ can be related with $M_i^{(q)}$. Denote these vectors by $X_{i_1}, X_{i_2}, ... \mid (X_{i_1}, X_{i_2}, ... \in \{X_1, X_2, ..., X_m\})$.
- Determine neurons-winners of the (q–1) block that were related with $X_{i_1}, X_{i_2}, ...$ Denote these neurons-winners by $M_{j_1}^{(q-1)}, M_{j_2}^{(q-1)}, ... \mid (M_{j_1}^{(q-1)}, M_{j_2}^{(q-1)}, ... \in \{M_1^{(q-1)}, M_2^{(q-1)}, ..., M_{r_{q-1}}^{(q-1)}\})$, and their two-dimensional projections, obtained as a result of MDS, by $Y_{j_1}^{(q-1)}, Y_{j_2}^{(q-1)}, ... \mid (Y_{j_1}^{(q-1)}, Y_{j_2}^{(q-1)}, ... \in \{Y_1^{(q-1)}, Y_2^{(q-1)}, ..., Y_{r_{q-1}}^{(q-1)}\})$,
- There are two possible ways of assignment:

  by proportion: the initial coordinates of $Y_i^{(q)}$ are set to be equal to the mean value of the set of vectors $\{Y_{j_1}^{(q-1)}, Y_{j_2}^{(q-1)}, ...\}$. In Figure 4 (*top*), two points $Y_{j_1}^{(q-1)}$ and $Y_{j_2}^{(q-1)}$ are coincident, a point $Y_i^{(q)} = \frac{1}{3}\left(Y_{j_1}^{(q-1)} + Y_{j_2}^{(q-1)} + Y_{j_3}^{(q-1)}\right)$ is closer to the points $Y_{j_1}^{(q-1)}$ than to $Y_{j_3}^{(q-1)}$.

  by midpoint: as the coincident vectors can be between the vectors $\{Y_{j_1}^{(q-1)}, Y_{j_2}^{(q-1)}, ...\}$, the initial coordinates of $Y_i^{(q)}$ are set to be equal to the mean value of the set of only the non-coincident points $\{Y_{j_1}^{(q-1)}, Y_{j_2}^{(q-1)}, ...\}$. In Figure 4 (*bottom*), $Y_i^{(q)} = \frac{1}{2}\left(Y_{j_1}^{(q-1)} + Y_{j_3}^{(q-1)}\right)$.

After the assignment, the two-dimensional vectors $Y_1^{(q)}, Y_2^{(q)}, ..., Y_{r_q}^{(q)} \mid (Y_i^{(q)} = (y_{i1}^{(q)}, y_{i2}^{(q)}),\ i = 1,..., r_q)$ of the neurons-winners are calculated using MDS.

The training of the neural network is continued until $q = \gamma$. After the γth block, we get two-dimensional projections $Y_1^{(\gamma)}, Y_2^{(\gamma)}, ..., Y_{r_\gamma}^{(\gamma)}$ of the n-dimensional neurons-winners $M_1^{(\gamma)}, M_2^{(\gamma)}, ..., M_{r_\gamma}^{(\gamma)}$ that are uniquely related with the vectors X_1, X_2, ..., X_m. The two-dimensional vectors $Y_1^{(\gamma)}, Y_2^{(\gamma)}, ..., Y_{r_\gamma}^{(\gamma)}$ obtained can be presented on a scatter plot (see Figure 5).

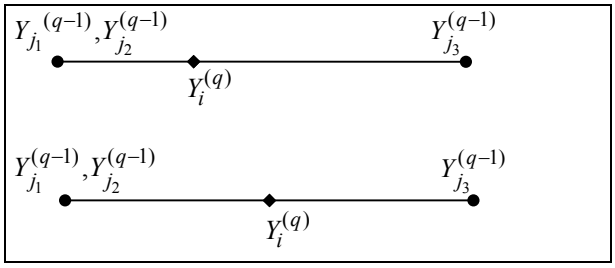

Figure 4. Two ways of assignment: by proportion (*top*), by midpoint (*bottom*)

4. Experimental Results

Some experiments have been carried out in order to ascertain: (1) which vector quantization method (SOM or NG) is more suitable to use in the combination with MDS; (2) which initialization way of two-dimensional points is most suitable in the consecutive combination of SOM or NG and MDS, as well as in the first block of the integrated combination (when the points are generated at random, on a line, according to two principal components (PCs), according to the components with the largest variances); (3) which way of assignment in the integrated combination is the most suitable one (by midpoint or by proportion).

The results of experimental investigation of some datasets are presented here: Iris (m = 150, n = 4) [21], hepta (m = 212, n = 3) [22], rand_data (m = 1500, n = 5) (here each component is generated at random in the interval (0,1)). SOM and NG are trained during 200 epochs ($\hat{e} = 200$). The training process is divided into γ = 2, 4, 8, 10, 25 blocks in the integrated combination, and ν = 100, 50, 25, 20, 8, respectively. 100 iterations are performed in MDS. The values of the normalized stress $\hat{E}_{MDS}$ (3) subject to the initialization and assignment ways for three datasets are presented in Tables 1–3. When choosing a random

initialization, ten experiments are done for each dataset and the averaged values are presented in Tables 1–3 and Figure 6. The smallest values are in an italic font and the most frequent values are in bold. The number s of neurons is set such that the same or a similar number r of neurons-winners were obtained by both vector quantization methods in order to compare the results obtained in the sense of the MDS stress.

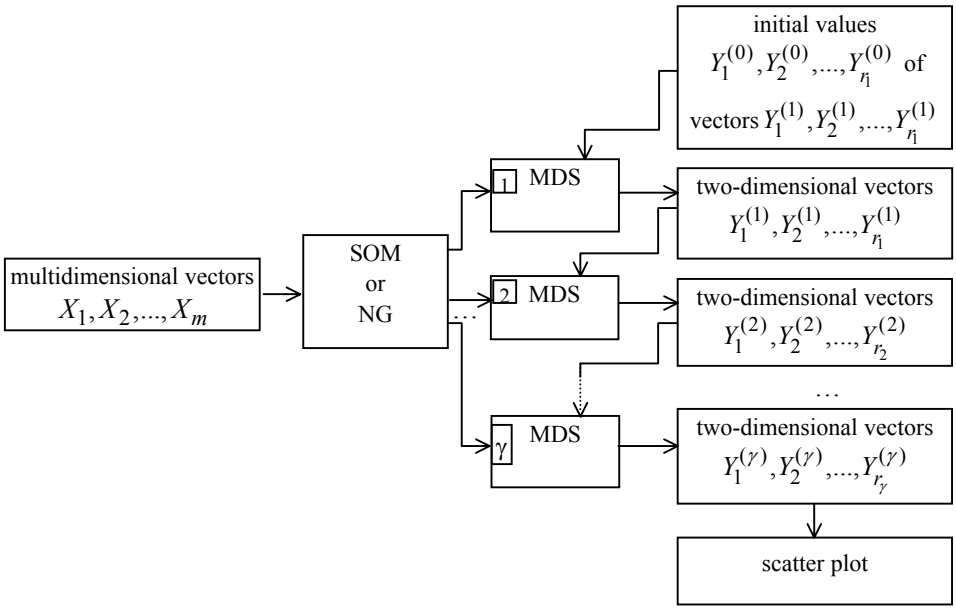

Figure 5. Scheme of the integrated combination of SOM or NG and multidimensional scaling

Table 1. The values of the MDS normalized stress subject to the initialization (*at random, on a line, by PCs, by variance*) and assignment (*by midpoint, by proportion*) ways for the iris dataset. SOM (a) (E_{QE} = 0.2225, r = 93) and NG (b) (E_{QE} = 0.0988, r = 94) are used in the consecutive and integrated combinations with MDS

a)

			at random		**on a line**		**by PCs**		**by variances**	
consecutive			0.0363		0.0366		0.0276		***0.0265***	
	γ	ν	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***
integrated	2	100	0.0385	0.0386	0.0484	0.0484	0.0395	0.0436	0.0438	0.0438
	4	50	0.0371	0.0373	***0.0265***	0.0271	0.0382	0.0269	0.0382	0.0382
	8	25	0.0335	0.0296	***0.0265***	***0.0265***	***0.0265***	***0.0265***	0.0347	***0.0265***
	10	20	0.0281	***0.0265***	0.0347	***0.0265***	***0.0265***	***0.0265***	***0.0265***	***0.0265***
	25	8	0.0298	0.0290	0.0347	***0.0265***	0.0347	***0.0265***	0.0347	***0.0265***

b)

			at random		**on a line**		**by PCs**		**by variances**	
consecutive			0.0489		0.0642		***0.0335***		0.0358	
	γ	ν	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***
integrated	2	100	0.0451	0.0452	0.0381	0.0561	***0.0335***	***0.0335***	***0.0335***	***0.0335***
	4	50	0.0399	0.0417	***0.0335***	***0.0335***	***0.0335***	***0.0335***	***0.0335***	***0.0335***
	8	25	0.0366	0.0363	***0.0335***	***0.0335***	***0.0335***	***0.0335***	***0.0335***	***0.0335***
	10	20	0.0392	0.0384	***0.0335***	***0.0335***	***0.0335***	0.0349	0.0349	0.0349
	25	8	0.0369	0.0388	0.0506	***0.0335***	***0.0335***	***0.0335***	***0.0335***	***0.0335***

When comparing the results, obtained by the consecutive and integrated combinations, smaller values of the normalized stress are obtained by the integrated combination in many cases. Thus, the integrated combination is superior to the consecutive one. It is quite evident, if the points are initiated on a line or at random (Figure 6). The values of the MDS normalized stress, obtained by the consecutive combination and the smallest values of the stress, obtained by the integrated combination, are presented in Figure 6.

In most cases, the normalized stress is slightly larger, if NG is used instead of SOM in combinations. However, the quantization error E_{QE} (1) is considerably smaller, therefore NG is more suitable in the combinations.

When the number γ of blocks of the integrated combination is increased, the normalized stress is rather fluctuating however it is no larger than that obtained by the consecutive combination.

The smallest value of the normalized stress for the iris dataset is obtained, if the initial values of two-dimensional points are set by variances, when SOM is used in the consecutive combination, $\hat{E}_{MDS} = 0.0265$, and by principal components, when NG is used $\hat{E}_{MDS} = 0.0335$. However, the same minimal value of the normalized stress is obtained by the integrated combination, when other initialization ways are used.

Table 2. The values of the MDS normalized stress subject to the initialization (*at random, on a line, by PCs, by variance*) and assignment (*by midpoint, by proportion*) ways for the hepta dataset. SOM (a) (E_{QE} = 0.3115, r = 86) and NG (b) (E_{QE} = 0.1765, r = 94) are used in the consecutive and integrated combinations with MDS

a)

			at random		**on a line**		**by PCs**		**by variances**	
consecutive			0.2182		0.2270		0.2042		0.2042	
	γ	ν	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***
integrated	2	100	0.2004	0.2066	***0.1994***	***0.1994***	***0.1994***	***0.1994***	***0.1994***	***0.1994***
	4	50	0.2078	0.2345	***0.1994***	***0.1994***	***0.1994***	**0.**2042	0.2270	0.2487
	8	25	***0.1994***	0.2109	***0.1994***	0.2270	***0.1994***	***0.1994***	***0.1994***	0.2270
	10	20	***0.1994***	0.2051	***0.1994***	0.2042	***0.1994***	***0.1994***	***0.1994***	0.2042
	25	8	***0.1994***	0.2081	***0.1994***	***0.1994***	***0.1994***	***0.1994***	***0.1994***	***0.1994***

b)

			at random		**on a line**		**by PCs**		**by variances**	
consecutive			0.2053		0.2115		**0.1964**		**0.1964**	
	γ	ν	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***
integrated	2	100	*0.1877*	*0.1877*	0.2043	0.2043	**0.1964**	**0.1964**	0.2043	0.2043
	4	50	0.2084	0.2084	0.2322	0.2322	0.2043	0.2043	0.2056	0.2056
	8	25	0.2194	0.2194	**0.1964**	**0.1964**	**0.1964**	**0.1964**	**0.1964**	**0.1964**
	10	20	0.2008	0.2052	**0.1964**	**0.1964**	**0.1964**	**0.1964**	**0.1964**	**0.1964**
	25	8	0.2115	0.2031	0.2115	**0.1964**	0.2115	**0.1964**	0.2115	**0.1964**

Table 3. The values of the MDS normalized stress subject to the initialization (*at random, on a line, by PCs, by variance*) and assignment (*by midpoint, by proportion*) ways for the rand_data dataset. SOM (a) (E_{QE} = 0.2139, r = 394) and NG (b) (E_{QE} = 0.1380, r = 400) are used in the consecutive and integrated combinations with MDS

a)

			at random		**on a line**		**by PCs**		**by variances**	
consecutive			0.3223		0.3189		0.3153		*0.3140*	
	γ	ν	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***
integrated	2	100	0.3244	0.3247	0.3252	0.3237	0.3241	0.3239	0.3241	0.3216
	4	50	0.3217	0.3225	0.3217	0.3220	0.3217	0.3220	0.3218	0.3229
	8	25	0.3176	0.3200	0.3178	*0.*3148	0.3176	*0.3142*	0.3177	0.3206
	10	20	0.3157	0.3155	0.3164	0.3162	0.3164	0.3164	0.3164	0.3167
	25	8	0.3159	**0.3161**	0.3162	**0.3161**	0.3160	**0.3161**	0.3162	**0.3161**

b)

			at random		**on a line**		**by PCs**		**by variances**	
consecutive			0.3202		0.3223		0.3119		*0.3103*	
	γ	ν	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***	***midpoint***	***proportion***
integrated	2	100	0.3192	0.3143	0.3179	0.3179	0.3125	0.3123	0.3140	0.3116
	4	50	0.3168	0.3159	0.3160	0.3160	0.3183	0.3187	**0.3115**	0.3140
	8	25	0.3129	0.3122	0.3132	0.3157	**0.3115**	**0.3115**	*0.3103*	**0.3115**
	10	20	0.3124	0.3131	0.3116	0.3223	0.3116	0.3119	**0.3115**	*0.3103*
	25	8	**0.3115**	**0.3115**	**0.3115**	0.3220	**0.3115**	**0.3115**	**0.3115**	**0.3115**

The smallest value of the normalized stress $\hat{E}_{MDS} = 0.1994$ for the hepta dataset is obtained by the integrated SOM and MDS combination independent of the initialization way. When NG is used, the most frequent value $\hat{E}_{MDS} = 0.1964$ is obtained by the consecutive combination, if the initial values are set by variances or PCs. The same value is obtained by the integrated combination, if the initial values are set on a line. If the random initialization is used, the smallest value $\hat{E}_{MDS} = 0.1877$ is obtained by the integrated combination, $\gamma = 2$. There is no value of the normalized stress that could be minimal and repeated for the rand_data dataset in contrast to the iris and hepta ones. However, the tendency of stress decline is shown in the integrated combination, when the

number γ of blocks is increased. When two ways of assignment (by midpoint and proportion) in the integrated combination are compared, no great differences were noticed.

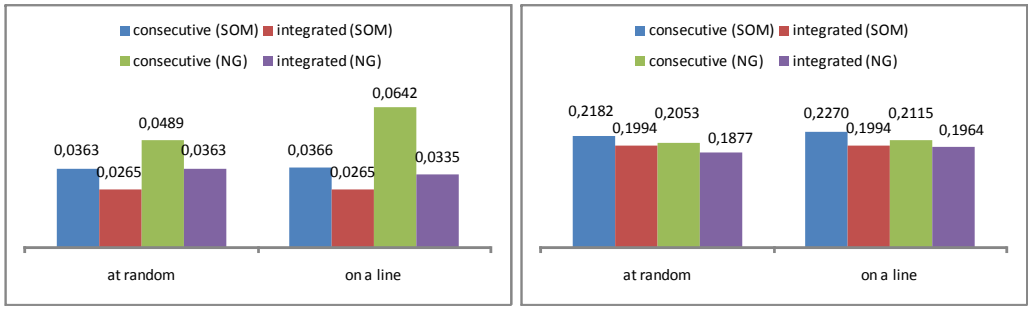

Figure 6. The values of the MDS normalized stress, obtained by the consecutive and integrated combinations, for the iris dataset (*left*) and the hepta dataset (*right*)

5. Conclusions

In the paper, two combinations (consecutive and integrated) of vector quantization and multidimensional scaling have been investigated. It is reasonable to use combinations, not only MDS with a certain number of neurons in order to save the computing time.

A conclusion on the usage of SOM and NG in combinations with MDS has been drawn: since the quantization error, obtained by NG, is considerably smaller than by SOM, if the number of neurons-winners is the same, it is reasonable to use NG in the combinations, though the MDS stress is slightly larger in these cases.

A conclusion on the assignment of the initial values of two-dimensional points in the integrated combination has been drawn: both proposed ways (by midpoint and by proportion) can be used, because any essential differences in the results obtained are not observed.

Some conclusions on the initialization of the values of two-dimensional points in the consecutive combination and the first block of the integrated combination have been drawn:

- If the initialization by the first two PCs or the components having the largest variances is used, rather a small MDS stress is obtained by the consecutive combination, however, sometimes it is possible to reduce it by the integrated combination.
- If the initialization, when the values are generated at random or set on a line, is used, the integrated combination is superior to the consecutive one in the sense of the MDS stress.

Acknowledgement

This research was funded by a grant (No. COST-05/2010) from the Research Council of Lithuania (COST Action IC0702). The authors' research was performed in cooperation with Vilnius University.

References

[1] **C. Chen, W. Hardle, A. Unwin**. Handbook of Data Visualization. *Springer, Berlin*, 2008.

[2] **G. Dzemyda, O. Kurasova, V. Medvedev**. 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, *S Press, Amsterdam*, 2007, 25–49.

[3] **P.E. Hoffman, G.G. Grinstein**. 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 Publishers, San Francisco,* 2002, 47–82.

[4] **A. Kačeniauskas, R. Pacevič, A. Bugajev. T. Katkevičius**. Efficient Visualization by Using Paraview Software on BalticGrid. *Information Technology and Control*, *Vol.*39, *No*. 2, 2010, 108–115.

[5] **R. Karbauskaitė, G. Dzemyda**. Topology Preservation Measures in the Visualization of Manifold-Type Multidimensional Data. *Informatica*, *Vol.*20(2), 2009, 235–253.

[6] **R. Karbauskaitė, O. Kurasova G. Dzemyda**. Selection of the Number of Neighbours of Each Data Point for the Locally Linear Embedding Algorithm. *Information Technology and Control*, *Vol.*36(4), 2007, 359–364.

[7] **J. Bernatavičienė, G. Dzemyda, O. Kurasova, V. Marcinkevičius**. Optimal Decisions in Combining the SOM with Nonlinear Projection Methods. *European Journal of Operational Research*, *Vol.*173, 2006, 729–745.

[8] **G. Dzemyda, O. Kurasova**. Heuristic Approach for Minimizing the Projection Error in the Integrated Mapping. *European Journal of Operat. Research, Vol.* 171(3), 2006, 859–878.

[9] **O. Kurasova, A. Molytė**. Investigation of the Quality of Mapping Vectors Obtained by Quantization Methods. *Proceedings of XIII International Conference on Applied Stochastic Models and Data Analysis, ASMDA* 2009, *Vilnius: Technika*, 2009, 269–273.

[10] **O. Kurasova, A. Molytė**. Combination of Vector Quantization and Visualization. *In: P. Perner (Ed.,) Machine Learning and Data Mining in Pattern Recognition – MLDM* 2009. *Lecture Notes in Artificial Intelligence*, *Springer Verlag, Heidelberg, Vol*.5632, 2009, 29–43.

[11] **O. Kurasova, A. Molytė**. Quality of Quantization and Visualization of Vectors obtained by Neural Gas and Self-Organizing Map. *Informatica*, *Vol.*22(1), 2011 (*in press*).

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

[13] **T.M. Martinetz, K.J. Schulten**. A Neural-Gas Network Learns Topologies. *Artificial Neural Networks (Kohonen, T., Mäkisara, K., Simula, O., and Kangas J. (eds.)*), 1991, 397–402.

[14] **I. Borg, P. Groenen**. Modern Multidimensional Scaling. *Springer, New York*, 2005.

[15] **J. Bernatavičienė, G. Dzemyda, V. Marcinkevičius**. Conditions for Optimal Efficiency of Relative MDS. *Informatica, Vol*.18(2), 2007, 187–202.

[16] **R. Mathar, A. Žilinskas**. On Global Optimization in Two-Dimensional Scaling. *Acta Aplicandae Mathematicae, Vol*.33, 1993, 109–118.

[17] **A. Žilinskas, J. Žilinskas**. Two Level Minimization in Multidimensional Scaling. *Journal of Global Optimization, Vol*.38(4), 2007, 581–596.

[18] **J. Žilinskas**. On Dimensionality of Embedding Space in Multidimensional Scaling. *Informatica*, *Vol*.19(3), 2008, 447–460.

[19] **P.A. Estévez, C.J. Figueroa, K. Saito**. Cross-Entropy Embedding of High-Dimensional Data Using the Neural Gas Model. *Neural Networks*, *Vol.*18, 2005, 727–737.

[20] **J. Handl**, **J. Knowles**. Cluster generators for large high-dimensional data sets with large numbers of clusters. *http://dbkgroup.org/ handl/generators/*.

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

[22] Fundamental Clustering Problems Suite (FCPS). *http://www.uni-marburg.de/fb12 datenbionik/ data//*

Received September 2010.