

# Efficient Data Projection for Visual Analysis of Large Data Sets Using Neural Networks

Viktor MEDVEDEV, Gintautas DZEMYDA, Olga KURASOVA,  
Virginijus MARCINKEVIČIUS

*Vilnius University, Institute of Mathematics and Informatics  
Akademijos 4, LT-08663 Vilnius, Lithuania  
e-mail: {viktor.medvedev, gintautas.dzemyda, olga.kurasova,  
virginijus.marcinkevicius}@mii.vu.lt*

Received: September 2011; accepted: December 2011

**Abstract.** The most classical visualization methods, including multidimensional scaling and its particular case – Sammon’s mapping, encounter difficulties when analyzing large data sets. One of possible ways to solve the problem is the application of artificial neural networks. This paper presents the visualization of large data sets using the feed-forward neural network – SAMANN. This back propagation-like learning rule has been developed to allow a feed-forward artificial neural network to learn Sammon’s mapping in an unsupervised way. In its initial form, SAMANN training is computation expensive. In this paper, we discover conditions optimizing the computational expenditure in visualization even of large data sets. It is shown possibility to reduce the original dimensionality of data to a lower one using small number of iterations. The visualization results of real-world data sets are presented.

**Keywords:** large multidimensional data sets, SAMANN, neural network, visualization.

## 1. Introduction

It is often desirable to visualize a data set which items are described by more than three features. Therefore, we have multidimensional data and our goal is to make some visual insight into the analyzed data set. For human perception, the data must be represented in a low-dimensional space, usually of two or three dimensions. The goal of the visualization methods is to represent the input data items in a lower-dimensional space so that certain properties (e.g., clusters, outliers) of the structure of the data set were preserved as faithfully as possible. Such visualization is highly important in data mining, because recent applications produce large amount of data that need specific means for the knowledge discovery and dimensionality reduction or visualization methods are recent techniques to discover knowledge hidden in large multidimensional data sets.

This paper focuses on dimensionality reduction (or visualization) methods as the tool for the analysis of large multidimensional data sets. Today’s large multidimensional data sets contain huge amount of data that becoming almost impossible to manually analyse them and to extract valuable information. As data sets become increasingly large, we

require more effective ways to display, analyse and interpret the information contained within them. A deep review of the dimensionality reduction methods (classical projection methods, artificial neural networks based methods, nonlinear manifold learning methods, etc.) may be restored from Borg and Groenen (2005), Dzemyda *et al.* (2007), Jolliffe (2004), Karbauskaitė and Dzemyda (2009).

Classical projection methods include the linear principal component analysis (PCA) and the multidimensional scaling (MDS). PCA seeks (Jackson, 2003) to reduce the dimensionality of the data by finding a orthogonal linear combinations (the principal components, PCs) of the original variables (features) with the largest variance. The interpretation of the PCs can be difficult at times. PCA can be used to display the data as a linear projection on such a subspace of the original data space that best preserves the variance in the data. The PCA cannot embrace nonlinear structures, consisting of arbitrarily shaped clusters or curved manifolds, since it describes the data in terms of a linear subspace. An alternative perspective on dimensionality reduction is offered by MDS (Borg and Groenen, 2005). MDS is another classical nonlinear approach that maps the original high dimensional data set to a lower dimensional data set, but does so in an attempt to preserve distances of the corresponding data points.

In Fig. 1 visualization results of Housing data set (Asuncion and Newman, 2007; the number of items (objects) is 506, the number of features is 14) using two different methods (PCA and MDS) have been presented.

Projection accuracy of PCA is limited by its global linearity. MDS, which does a nonlinear projection of data, attempts to avoid from such a drawback. However, the computational complexity of the most classical metric MDS methods is  $O(m^2)$ , where  $m$  is the number of items (objects) forming the data set. Thus, the MDS method is unsuitable for large data sets: it requires much computational resources. Therefore, the combinations of different data visualization methods are under a rapid development. The combination of different methods can be applied to make more efficient data analysis, while minimizing the shortcomings of individual methods [see, e.g., integration of the self organizing map and some particular case of MDS (Sammon's mapping) in Dzemyda and Kurasova

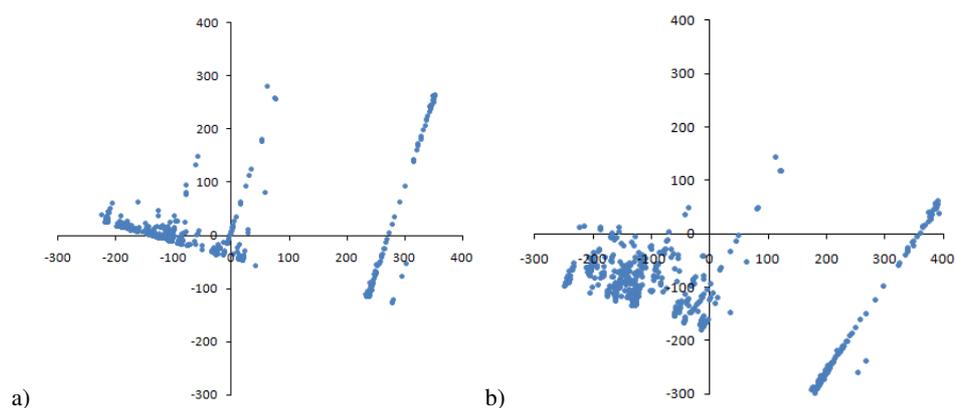


Fig. 1. Visualization results of Housing data set using: (a) PCA and (b) MDS.

(2006); integration of the self-organizing map and neural gas with MDS in Kurasova and Molytè (2011)].

Artificial neural networks may be used for dimensionality reduction and data visualization. The MDS got some attention from neural network researchers (Lowe and Tipping, 1996; Mao and Jain, 1995; van Wezel and Kusters, 2004). The back propagation-like learning rule, SAMANN, has been developed to allow a feed-forward artificial neural network to learn Sammon's mapping in an unsupervised way (Mao and Jain, 1995; de Ridder and Duin, 1997; Ivanikovas *et al.*, 2008; Medvedev and Dzemyda, 2006). Sammon's mapping is a particular case of the MDS. In its initial form, SAMANN training is computation expensive. In this paper, we discover conditions optimizing the computational expenditure in visualization even of large data sets. It is shown possibility to reduce the original dimensionality of data to a lower one using a small number of iterations.

## 2. The Neural Network Implementation for Data Visualization

The goal of projection in the metric MDS is to optimize the representations so that the distances between the points in the lower-dimensional space would be as close to the original dissimilarities (distances) as possible. Let the initial dimensionality is  $n$ : the analysed data set is  $\{X_i = (x_{1i}, \dots, x_{ni}), i = \overline{1, m}\}$ . The resulting dimensionality (denote it by  $d$ ) is 2 or 3: the resulting data set is  $\{Y_i = (y_{1i}, \dots, y_{di}), i = \overline{1, m}\}$ . Denote the distance between the points  $X_i$  and  $X_j$  in the feature (input) space  $R^n$  by  $d_{ij}^*$ , and the distance between their projections  $Y_i$  and  $Y_j$  in the projected (output) space  $R^d$  by  $d_{ij}$ . The metric MDS tries to approximate  $d_{ij}$  by  $d_{ij}^*$ . Ideally, the distances between points in the projected data space should be identical to the distances between corresponding points in the reduced data space. A particular case of the metric MDS is Sammon's mapping (Sammon, 1969). It tries to optimize a projection error that describes how well the pairwise distances in a data set are preserved:

$$E_S = \frac{1}{\sum_{\substack{i,j=1 \\ i < j}}^m d_{ij}^*} \sum_{\substack{i,j=1 \\ i < j}}^m \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*}. \quad (1)$$

In the error function  $E_S$ , the smaller the error value, the better projection we get. Sammon's mapping places greater emphasis on smaller distances rather than on larger distances. In contrast to PCA, Sammon's mapping is a nonlinear transformation of the data set  $X_i = (x_{1i}, \dots, x_{ni}), i = \overline{1, m}$ . Sammon (1969) applied a gradient steepest descent procedure to find a configuration of  $m$  points in the  $d$ -dimensional space that minimizes  $E_S$ .

The application of original Sammon's mapping (and MDS in general) becomes impractical for large data sets. Another disadvantage of Sammon's mapping is that when a new data point has to be mapped, the whole mapping procedure has to be repeated. To solve this problem, some methods have been proposed: triangulation algorithm (Lee, 1977), standard feed-forward neural network (de Ridder and Duin, 1997).

Mao and Jain (1992, 1995) have suggested a neural network implementation of Sammon's mapping which is able to project new points after training. It has been concluded in Mao and Jain (1995) that SAMANN neural network preserves the data structure, cluster shape, and interpattern distances well. A drawback of using SAMANN is that it is rather difficult to train and the training process is extremely slow.

The architecture of the SAMANN network is a multilayer perceptron where the number of input nodes is set to be the input space dimensionality,  $n$ , and the number of output nodes is specified as the projected space dimensionality,  $d$ .

The classical way of training the SAMANN network is described in Mao and Jain (1995). The general weights updating rule that minimizes Sammon's error for all hidden layers,  $l = 1, \dots, L - 1$  and for the output layer ( $l = L$ ) is:

$$\Delta w_{jt}^{(l)} = -\eta \frac{\partial E_S(\mu, \nu)}{\partial w_{jt}^{(l)}}, \quad (2)$$

where  $w_{jt}^{(l)}$  is the weight between the unit  $j$  in the layer  $l - 1$  and the unit  $t$  in the layer  $l$ ,  $\eta$  is the learning rate.

The sigmoid activation function  $g(x)$  whose range is (0.0, 1.0) is used for each unit:

$$g(x) = \frac{1}{(1 + e^{-kx})}, \quad (3)$$

$k$  is the slope parameter of the sigmoid function. By varying the parameter  $k$ , it is possible to obtain sigmoid functions of different slopes.

The network takes a pair of input points each time in the training. The outputs of each neuron are stored for both points. The distance between the neural network output points can be calculated and a projection error measure can be defined in terms of this distance and the distance between the points in the input space. The weights of neural network are updated according to the update rule (2) using the error measure  $E_S$ . After training, the network is able to project previously unseen data generalising the mapping rule.

### 3. Large Data Sets Visualization Using SAMANN

Visualization of the large data set requires huge computational resources and time. That is why the most classical visualization methods are unsuitable for large data sets. In spite of some limitations, we'll show in this paper that SAMANN neural network can be successfully used for the large multidimensional data set visualization.

One of the ways to minimize the computational expenditure for the neural network training is working with a subset of the primary data set. The results of the experiments showed Ivanikovas *et al.* (2007) that it is possible to find such a subset of the analyzed data set that the lower projection errors are obtained faster while training the SAMANN network with this subset than by training with all the points of the set.

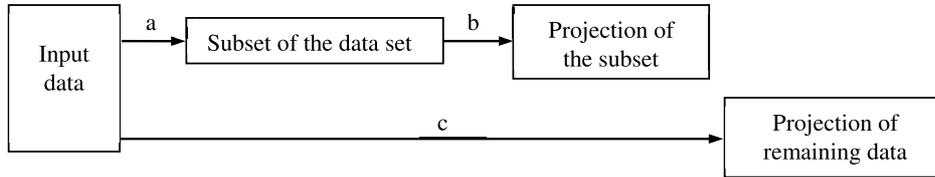


Fig. 2. Scheme of the visualization process: (a) selection of the subset of the primary data set; (b) projection of the subset using SAMANN; (c) projection of the remaining points.

We suggest the following algorithm for the visualization of large multidimensional data set:

- (a) at first, it is necessary to form the subset  $B$  of the primary data set  $A = \{X_i = (x_{1i}, \dots, x_{ni}), i = \overline{1, m}\}$ ,  $B \subset A$ . The subset  $B$  consists of the points chosen randomly (or using some deterministic way) from the primary data set;
- (b) the points of the subset are mapped on  $R^d$  ( $d < n$ ) using SAMANN neural network where the weights of the network are calculated using iteration weights updating process (2);
- (c) all the remaining points from  $A \setminus B$  are projected using the calculated weights of the network.

In Fig. 2, the scheme of visualization process is presented.

The training process of the SAMANN network depends on some control parameters, such as learning rate, activation function, momentum term, the range of the initial network weights and etc. Looking for the optimal control parameters, we can speed up the training process of the network and apply it for the large data set.

One training iteration of the neural network is complete if all possible pairs of points from the subset  $B$  are shown to the neural network at once. If the optimal values of the control parameters are chosen, one can expect that good projection is obtained even after a small number of training iterations. This hypothesis is confirmed below.

#### 4. The Background for the Experiments

Control parameters that are associated with the neural network in general can be divided into two main groups. The first group involves control parameters of the neural network training: learning rate, momentum term, initial network weights. The second group involves parameters of neural network's architecture: number of hidden layers, number of units in each hidden layer, weights, type and parameters of activation function, etc. The dependence of the projection error on the learning rate and momentum term was investigated in Dzemyda *et al.* (2007), Medvedev and Dzemyda (2006).

In this paper, we perform the experiments to evaluate how the SAMANN network training process (visualizing the large data set) depends on:

- the initial range of the weights,
- the slope parameter of the sigmoid activation function.

The SAMANN network in our experiments (Section 5) was trained by a subset (10% of the primary data set) of the analyzed data set to speed up the training process and to make the network suitable to visualize large data set because of possible computational economy. Such reduction (10%) of the primary data set was selected quite empirically; see Ivanikovas *et al.* (2007). This parameter in some cases may be smaller as well.

For experiments, the SAMANN neural network was constructed to map the primary data set to a plane ( $d = 2$ ). The network with one hidden layer of units (neurons) has been used.

The experiments in Dzemyda *et al.* (2007) and Medvedev and Dzemyda (2006) demonstrate that with an increase of the learning rate value  $\eta$  to some rather high extent, a better projection error is obtained. That is why the learning rate  $\eta$  was set to 10.

Before applying the dimensionality reduction, it is useful to preprocess the data. The preprocessing consists of the data normalization. We normalize all input points as follows:

$$X'_i = \frac{X_i}{S}, \quad i = \overline{1, m}, \quad (4)$$

where  $S = \max_{i,j} \|X_i - X_j\|$  is the maximal distance between all input points. Such normalization changes all distances between the points proportionally, but does not change interlocation of points in  $R^n$ . The maximal distance between the normalized points  $X'_i = (x'_{1i}, \dots, x'_{ni})$ ,  $i = \overline{1, m}$  is equal to 1. The points  $X'_i$ ,  $i = \overline{1, m}$  are assumed to form the primary data set.

It may be shown that such preprocessing does not change optimal values of  $Y_i = (y_{1i}, \dots, y_{di})$ ,  $i = \overline{1, m}$  and the error function  $E_S$ .

## 5. Experiments on Real Data Sets

### 5.1. Real Data Sets

Four real data sets from UCI Machine Learning Repository (Asuncion and Newman, 2007) were used to investigate the ability to visualize large data set using SAMANN:

- *Dataset1.* Shuttle data set. The shuttle data set contains 9 attributes all of which are numerical. 58 000 data points in 9-dimensional input space, 7 classes ( $m = 58\,000$ ,  $n = 9$ ). Approximately 80% of the data belongs to class 1.
- *Dataset2.* MAGIC Gamma Telescope. The MAGIC Gamma Telescope reads data from the cosmic rays striking Earth's atmosphere. These results include both gamma rays and hadronic showers. It is important to determine between these for experimental purposes, therefore this project will mine the given data set of 11 attributes and 19 020 items to find association rules which can be used to classify future readings from this telescope ( $m = 19\,020$ ,  $n = 10$ ).
- *Dataset3.* Page blocks classification. The data set consists of 5473 10-dimensional points from five classes ( $m = 5473$ ,  $n = 10$ ).

Table 1  
Real data sets

Data set	No. of items	No. of attributes	No. of classes
<i>Dataset1</i>	58 000	9	7
<i>Dataset2</i>	19 020	10	2
<i>Dataset3</i>	5 473	10	5
<i>Dataset4</i>	10 992	16	10

- *Dataset4*. Pen-based recognition of handwritten digits. The data set consists of 10 992 16-dimensional items from ten classes ( $m = 10\,992$ ,  $n = 16$ ). The classes are pen-based handwritten digits with ten digits 0 to 9.

All data sets and their main properties are listed in Table 1.

### 5.2. Dependence of Projection Error on the Control Parameters

For the each analyzed data set, series of experiments have been performed. The SAMANN neural network was trained by choosing different values of the range of initial weights  $W$  of the neural network and the slope parameter  $k$  of the sigmoid function:

1. The network was initialized with random values of the uniformly distributed weights ( $W$ ) from the ranges:  $W \in [-1, 1]$ ,  $[-0.1, 0.1]$ ,  $[-0.01, 0.01]$ ,  $[-0.001, 0.001]$ .
2. Different values of the slope parameter of the sigmoid function  $g(x)$  (3) have been used:  $k = 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500, 1000$ .

Usually the sigmoid function with the slope parameter  $k = 1$  is used, and the range of initial weights is not presented in references on the SAMANN realization. We performed the experiments to evaluate how the SAMANN network training process depends on these parameters.

All possible combinations of the ranges of the initial weights and values of the slope parameter have been examined seeking to find the best their combination for the data set visualization (minimizing the projection error  $E_S$ ). Thus, for each data set, 44 experiments have been performed.

We assume that good projection can be obtained even after a small number of training iterations. Therefore, in our experiments, only one iteration of the network training was performed: all possible pairs of points from the subset  $B$  of the primary data set  $A$  have been shown to the network at once. In order to evaluate the mapping efficiency of whole data set, the projection error  $E_S$  is calculated using all points from the primary data set.

All the experiments with fixed combinations of slope parameter  $k$  and the range of initial weights  $W$  have been performed several times and the results were averaged. In this way, the experiments with several different random subsets of each data set and fixed combination of slope parameter and the range of initial weights have been performed.

The least and the average values of the projection error  $E_S$  after one iteration of the network training have been measured and presented below for different cases.

In Figs. 3–6, the projection error dependence  $E_S = \varphi(W, k)$  on the range of the initial weights  $W$  and the slope parameter  $k$  of the activation function is presented graphically for different data sets. Figures 3a, 4a, 5a, 6a show the best (least) projection errors for each analyzed data set over several different random subsets. The average projection errors are presented in Figs. 3b, 4b, 5b, 6b. The filled area in the figures and tables shows the projection errors that are smaller than 0.1.

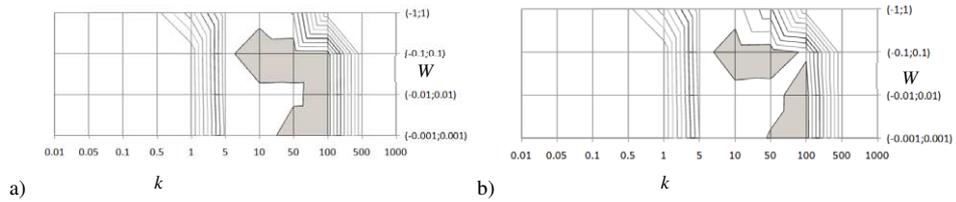


Fig. 3. The least (a) and the average (b) projection errors for *Dataset1*.

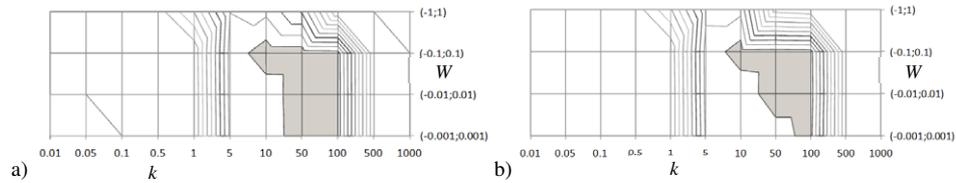


Fig. 4. The least (a) and the average (b) projection errors for *Dataset2*.

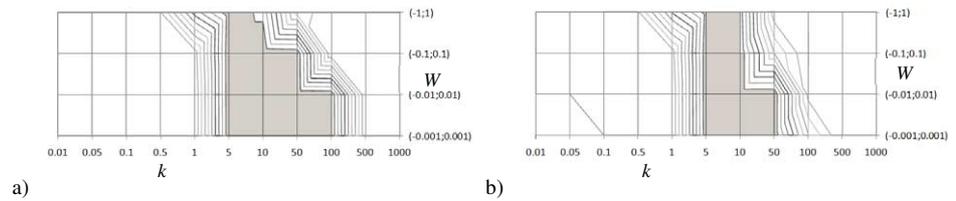


Fig. 5. The least (a) and the average (b) projection errors for *Dataset3*.

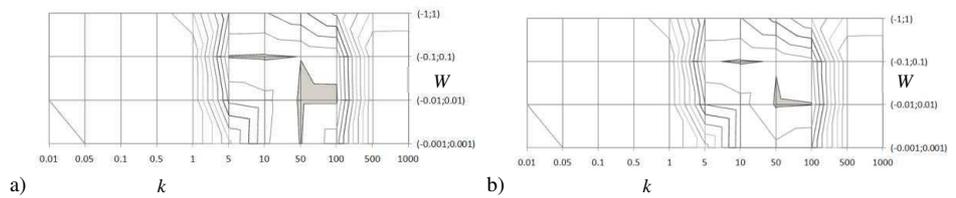


Fig. 6. The least (a) and the average (b) projection errors for *Dataset4*.

Figures 3–6 illustrate the fact that the different values of control parameters of the network can give so different projection results after the one iteration and there are zones of a small projection error. Moreover, the results in Figs. 3–6 lead to the idea that the projection error is the multiextremal function and some special methods (Mockus, 1989; Horst *et al.*, 2000) may be used for its minimization, if needful.

The comparison of the results obtained while using only the subset of the primary data set for SAMANN network training shows that it is possible to find such values of control parameters (the intervals of initial weights and the slope parameter of the sigmoid function) that can be used for the visualization of whole primary data set. The experiments show that even after one iteration rather small projection error can be obtained when the optimal values of parameters mentioned above are used.

When comparing Figs. 3–6 for different data sets, we observed the common tendency that the smallest values of projection error are obtained for the following combination of the initial weights range and the slope parameter of the sigmoid function:  $k = 10$ ,  $W \in [-0.1, 0.1]$ . Rather good results are obtained when  $k = 50$  and  $W \in [-0.01, 0.01]$ .

### 5.3. Visualization Results

The visualization results of two data sets (*Dataset1* and *Dataset4*) using different slope parameter’s values  $k$  and the initial weights range of  $W$  for the neural network training are presented in Figs. 7, 10. Figures 7a, 10a illustrate the visualization results of *Dataset1* and *Dataset4* using  $k = 1$  because the sigmoid function with the slope parameter  $k = 1$  is usually used in references on the SAMANN realization.

Using  $k = 1$  and  $W \in [-0.1, 0.1]$  (Figs. 7a and 10a illustrate results for *Dataset1* and *Dataset4*, respectively), the obtained projection error was quite high (bad):

- *Dataset1*:  $E_S = 0.99027$ ;
- *Dataset4*:  $E_S = 0.98962$ .

Using  $k = 10$ ,  $W \in [-0.1, 0.1]$  (Figs. 7b and 10b illustrate results for *Dataset1* and *Dataset4*), the obtained projection error was significantly smaller (better):

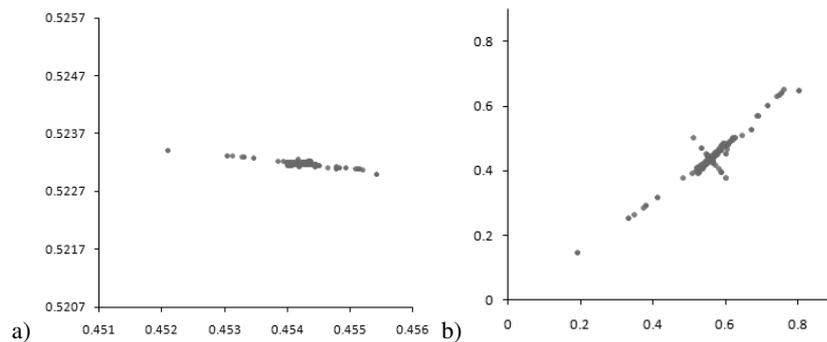


Fig. 7. Visualization results of *Dataset1*: (a)  $k = 1$ ,  $W \in [-0.1, 0.1]$ ,  $E_S = 0.99027$ ; (b)  $k = 10$ ,  $W \in [-0.1, 0.1]$ ,  $E_S = 0.06979$ .

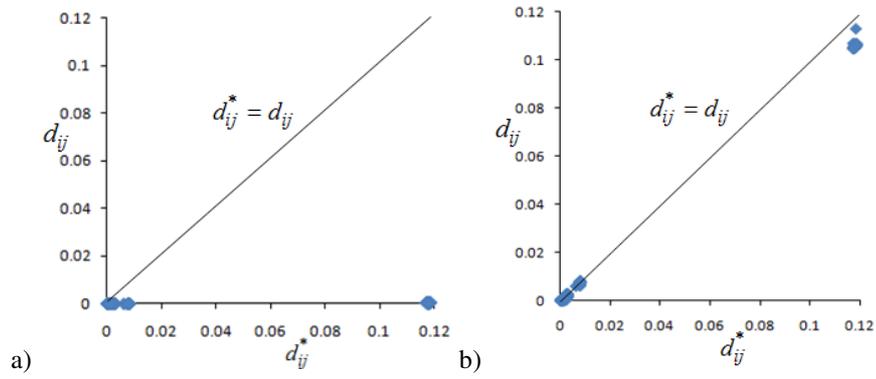


Fig. 8. Distribution  $(d_{ij}^*, d_{ij})$  for *Dataset1*: (a)  $k = 1$ ,  $W \in [-0.1, 0.1]$ ,  $E_S = 0.99027$ ; (b)  $k = 10$ ,  $W \in [-0.1, 0.1]$ ,  $E_S = 0.06979$ .

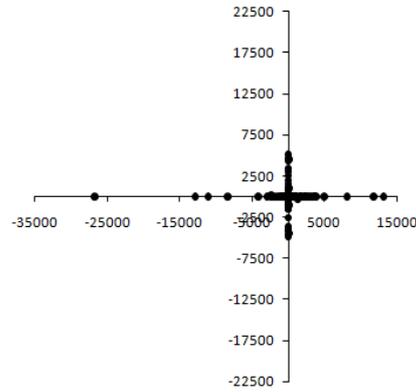


Fig. 9. Visualization results of *Dataset1* using PCA.

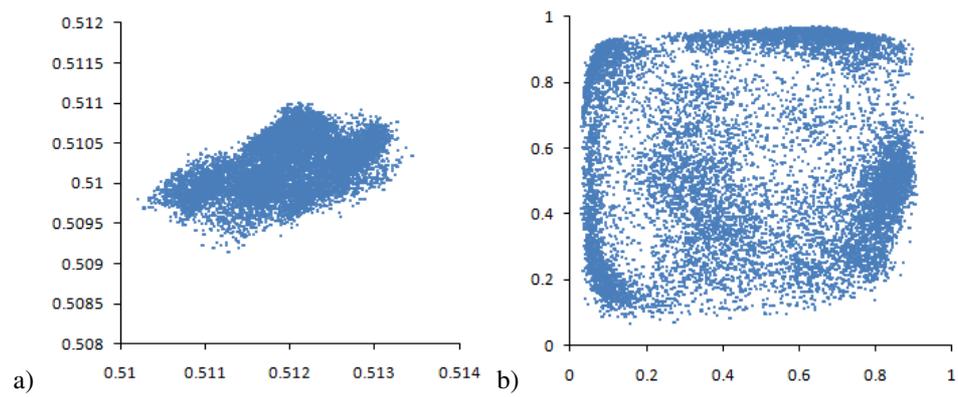


Fig. 10. Visualization results of *Dataset4*: (a)  $k = 1$ ,  $W \in [-0.1, 0.1]$ ,  $E_S = 0.98962$ ; (b)  $k = 10$ ,  $W \in [-0.1, 0.1]$ ,  $E_S = 0.08890$ .

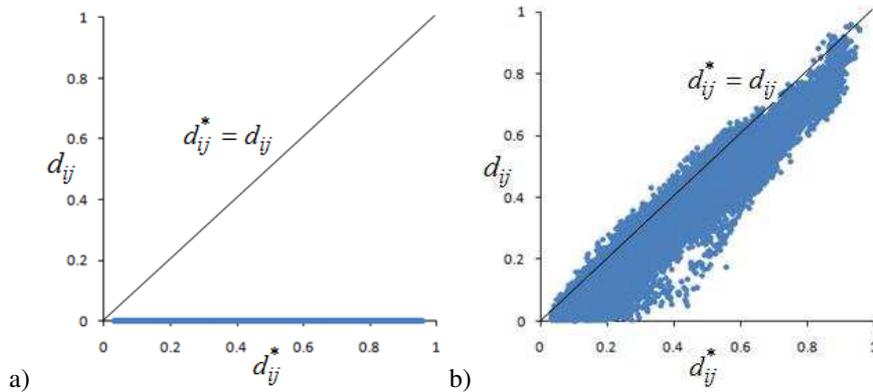


Fig. 11. Distribution  $(d_{ij}^*, d_{ij})$  for *Dataset4*: (a)  $k = 1$ ,  $W \in [-0.1, 0.1]$ ,  $E_S = 0.98962$ ; (b)  $k = 10$ ,  $W \in [-0.1, 0.1]$ ,  $E_S = 0.08890$ .

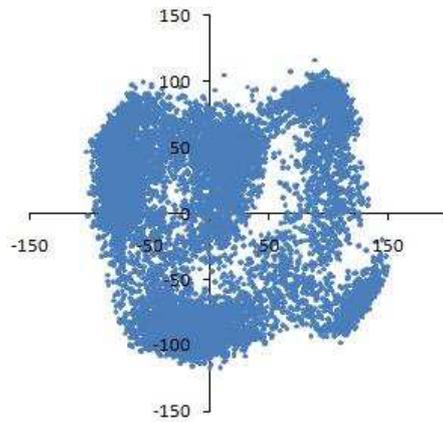


Fig. 12. Visualization results of *Dataset4* using PCA.

- *Dataset1*:  $E_S = 0.06979$ ;
- *Dataset4*:  $E_S = 0.08890$ .

Also we compared the best obtained visualization results using SAMANN (one iteration only) with PCA (Figs. 9, 12). Figures 7b, 10b (using SAMANN) and 9, 12 (using PCA) for all four data sets show that visualization results of large data sets have similarities. Using only 10% of the primary data set and optimal values of control parameters, the computational time of the SAMANN algorithm is approximate to the computational expenditure of PCA.

The experiments with remaining two data sets (*Dataset2* and *Dataset3*) lead to similar conclusions. In addition, it is noticed from the experiments with *Dataset2* that some data set analysis may require for more iterations to get rather good visualization results. Most probably, the main reason of this is that the maximal distance between the points in the input space is much larger than most remaining distances. However, if the optimal values

of the control parameters are chosen, a good projection is obtained even after a small number of training iterations.

Comparing the results in general, natural question arises – which representation of the data set is better: Figs. 7a or 7b of *Dataset1*, and Figs. 10a or 10b of *Dataset4*. Visually, results in Figs. 7a and 7b are similar in some extent. However, the projection error  $E_S$  values are quite different. In order to assess the quality of the visualization results, we can study the differences between the distances  $d_{ij}^*$  and  $d_{ij}$ . If projection error  $E_S$  is zero (in ideal case) then  $d_{ij}^* = d_{ij}$  for all  $i$  and  $j$ . Figures 8a and 8b show experimental dependences of  $d_{ij}^*$  and  $d_{ij}$  for visualization results of *Dataset1* that are presented in Figs. 7a and 7b, respectively. The idea to analyze such dependences of  $d_{ij}^*$  and  $d_{ij}$  (distributions of  $(d_{ij}^*, d_{ij})$ ) comes from Shepard diagram (Borg and Groenen, 2005).

Thus, we can conclude, the visualization results of *Dataset1* in Fig. 7b (using  $k = 10$ ,  $W \in [-0.1, 0.1]$ ) are better than in Fig. 7a (using  $k = 1$  and  $W \in [-0.1, 0.1]$ ) because the points in Fig. 8b scatter around the bisector from the lower left-hand corner to the upper right-hand corner.

Similar conclusions may be drawn comparing Figs. 10 and 11. The visualization results of *Dataset2* in Fig. 10b (using  $k = 10$ ,  $W \in [-0.1, 0.1]$ ) are better than in Fig. 10a (using  $k = 1$  and  $W \in [-0.1, 0.1]$ ).

## 6. Conclusions

This paper focuses on the visual analysis of large multidimensional data sets using dimensionality reduction methods. The most classical visualization (data dimensionality reduction) methods, including multidimensional scaling and its particular case – Sammon’s mapping, encounter difficulties when analyzing large data sets. One of possible ways to solve the problem is the application of artificial neural networks. This paper presents the visualization of large data sets with the feed-forward neural network SAMANN. In its initial form, SAMANN training is computation expensive. In this paper we discover conditions optimizing the computational expenditure in visualization even of large data sets.

In the research, the neural network was trained by choosing the different values of control parameters of the network: the interval of the initial weights of the neural network and the slope parameter of the sigmoid function. It is shown that the different values of control parameters of the network and its training can give so different projection results after the one iteration. The experiments proved the hypothesis that good projection is obtained even after a small number of training iterations if the optimal values of the control parameters are chosen.

The obtained results enable us to visualize large data sets using the neural network and to get good enough results. The recommendations for optimal values of control parameters are given.

## References

- Asuncion, A., Newman, D.J. (2007). *UCI Machine Learning Repository*. University of California, School of Information and Computer Science, Irvine.  
<http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- Borg, I., Groenen, P. (2005). *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York.
- Dzemyda, G., Kurasova, O. (2006). Heuristic approach for minimizing the projection error in the integrated mapping. *European Journal of Operation Research*, 171, 859–878.
- Dzemyda, G., Kurasova, O., Medvedev, V. (2007). Dimension reduction and data visualization using neural networks. In: *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, pp. 25–49. IOS Press, Amsterdam.
- Horst, R., Pardalos, P.M., Thoai, N.V. (2000). *Introduction to Global Optimization. Nonconvex Optimization and Its Applications*, Vol. 48, 2nd edn. Kluwer Academic, Dordrecht.
- Ivanikovas, S., Medvedev, V., Dzemyda, G. (2007). Parallel realizations of the SAMANN algorithm. In: *Adaptive and Natural Computing Algorithms, Lecture Notes in Computer Science*, Vol. 4432, pp. 179–188. Springer.
- Ivanikovas, S., Dzemyda, G., Medvedev, V. (2008). Large datasets visualization with neural network using clustered training data. In: *Advances in Databases and Information Systems, Lecture Notes in Computer Science*, Vol. 5207, pp. 143–152.
- Jackson, J.E. (2003). *A User's Guide to Principal Components*. Wiley, New York.
- Jain, A.K., Mao, J. (1992). Artificial neural network for nonlinear projection of multivariate data. In: *Proceedings International Joint Conference on Neural Networks*, pp. 335–340.
- Jolliffe, I.T. (2004). *Principal Component Analysis*, Springer Series in Statistics, 2nd edn. Springer, New York.
- Karbauskaitė, R., Dzemyda, G. (2009). Topology preservation measures in the visualization of manifold-type multidimensional data. *Informatica*, 20(3), 235–254.
- Kurasova, O., Molytė, A. (2011). Integration of the self-organizing map and neural gas with multidimensional scaling. *Information Technology and Control*, 40(1), 12–20.
- Lee, R.C.T., Slagle, J.R., Blum, H. (1977). A triangulation method for sequential mapping of points from n-space to two-space. *IEEE Transactions on Computers*, 27, 288–299.
- Lowe, D., Tipping, M.E. (1996). Feed-forward neural networks and topographic mappings for exploratory data analysis. *Neural Computing and Applications*, 4, 83–95.
- Mao, J., Jain, A.K. (1995). Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6, 296–317.
- Medvedev, V., Dzemyda, G. (2006). Optimization of the local search in the training for SAMANN neural network. *Journal of Global Optimization*, 35, 607–623.
- Mockus, J. (1989). *Bayesian Approach to Global Optimization*. Kluwer Academic, Dordrecht.
- de Ridder, D., Duin, R.P.W. (1997). Sammon's mapping using neural networks: a comparison. *Pattern Recognition Letters*, 18, 1307–1316.
- Sammon, J.W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18, 401–409.
- van Wezel, M.C., Kusters, W.A. (2004). Nonmetric multidimensional scaling: neural networks versus traditional techniques. *Intelligent Data Analysis*, 8(6), 601–613.

**V. Medvedev** was born in 1979 in Vilnius. In 2008 he received the doctoral degree in computer science (PhD) from Institute of Mathematics and Informatics jointly with Vilnius Gediminas Technical University. Recent employment is at the System Analysis Department of the Vilnius University, Institute of Mathematics and Informatics as researcher. His research interests include artificial intelligence, visualization of multidimensional data, dimensionality reduction, neural networks, data mining and parallel computing.

**G. Dzemyda** was born in Vilnius, Lithuania, in 1957. In 1984 he received the doctoral degree in technical sciences (PhD), and in 1997 he received the degree of doctor habilis from the Kaunas University of Technology. He was conferred the title of professor in 1998. Recent employment is at Vilnius University, Institute of Mathematics and Informatics as a director and head of System Analysis Department and principal researcher. The research interests include optimization theory and applications, data mining, neural networks, visualization of multidimensional data, parallel optimization, web-services, multiple criteria decision support. The author of more than 200 scientific publications and one monograph.

**O. Kurasova** was born in Zarasai district, Lithuania, in 1977. In 2005 she received the doctoral degree in computer science (PhD) from Institute of Mathematics and Informatics jointly with Vytautas Magnus University. Recent employment is at the System Analysis Department of the Vilnius University, Institute of Mathematics and Informatics as senior researcher, and at the Information Technology Department of Lithuanian University of Educational Sciences as associate professor. Research interests include data mining methods, optimization theory and applications, artificial intelligence, neural networks, visualization of multidimensional data, multiple criteria decision support, parallel computing. She is the author of more than 35 scientific publications.

**V. Marcinkevičius** was born in Alytus on the 21th of June in 1976. In 2010 he received the doctoral degree in computer science (PhD) from Vytautas Magnus University. Since 2001 he is employee of the Institute of Mathematics and Informatics. Recent employment is at the System Analysis Department of the Vilnius University, Institute of Mathematics and Informatics as researcher. The research interests include data mining in databases, multiple criteria decision support, neural networks, visualization of multidimensional data, parallel optimization. The author of more than 15 scientific publications. He is a member of the Computer Society and Lithuanian Mathematical Society.

## **Efektyvi duomenų projekcija didelės apimties duomenims vizualizuoti naudojant neuroninius tinklus**

Viktor MEDVEDEV, Gintautas DZEMYDA, Olga KURASOVA,  
Virginijus MARCINKEVIČIUS

Dauguma klasikinių vizualizavimo metodų, įskaitant daugiamates skales bei jų atskirą atvejį – Sammono algoritimą, nepajėgūs susidoroti su didelės apimties duomenų aibėmis. Vienas iš galimų šios problemos sprendimų būdų – neuroninių tinklų panaudojimas daugiamatiams duomenims vizualizuoti. Šiame straipsnyje siūloma didelės apimties duomenų aibes vizualizuoti naudojant tiesioginio sklidimo neuroninį tinklą SAMANN. Specifinė „klaidos sklidimo atgal“ mokymo taisyklė, pavadinta SAMANN, leidžia įprastam tiesioginio sklidimo neuroniniam tinklui realizuoti populiarią Sammono projekciją mokymo be mokytojo būdu. Šiame straipsnyje analizuojami didelės apimties duomenų vizualizavimo būdai, minimizuojant skaičiuojamąsias sąnaudas. Parodyta galimybė mažinti analizuojamų duomenų dimensiją, naudojant mažesnę iteracijų skaičių.