INFORMATICA, 2007, Vol. 18, No. 2, 187–202

Conditions for Optimal Efficiency of Relative MDS

Jolita BERNATAVIČIENĖ, Gintautas DZEMYDA,
Virginijus MARCINKEVIČIUS
Institute of Mathematics and Informatics
Akademijos 4, 08663 Vilnius, Lithuania
e-mail: {jolitab, dzemyda, virgism}@ktl.mii.lt

Received: November 2006

Abstract. In this paper, the relative multidimensional scaling method is investigated. This method is designated to visualize large multidimensional data. The method encompasses application of multidimensional scaling (MDS) to the so-called basic vector set and further mapping of the remaining vectors from the analyzed data set. In the original algorithm of relative MDS, the visualization process is divided into three steps: the set of basis vectors is constructed using the k-means clustering method; this set is projected onto the plane using the MDS algorithm; the set of remaining data is visualized using the relative mapping algorithm. We propose a modification, which differs from the original algorithm in the strategy of selecting the basis vectors. The experimental investigation has shown that the modification exceeds the original algorithm in the visualization quality and computational expenses. The conditions, where the relative MDS efficiency exceeds that of standard MDS, are estimated.

Key words: multidimensional scaling, visualization, clustering, relative MDS, large data sets, data mining.

1. Introduction

Nowadays, computer systems store large amounts of data. Due to the lack of abilities to explore adequately the large amounts of collected data, even potentially valuable data becomes useless and the data of databases dumps. Visual data exploration, which aims at providing an insight by visualizing the data and information visualization techniques, can help to solve this problem. Visual data exploration strives for integrating humans into the data exploration process, by applying their perceptual abilities of large data sets, which are available at present. The main idea is to present the data in some visual form, allowing data analysts to gain the insight into it and draw conclusions, as well as to interact with it (Keim, 2001). Therefore, a great attention has been paid to the analysis of large data sets of late, particularly to their visual analysis (Bernatavičienė *et al*., 2006a).

A set of some numerical parameters $x_1, x_2, ..., x_n$ characterizes a real object; here n is the number of parameters. These parameters compose a multidimensional vector, which corresponds to that object. The number of the analysed objects is m. Denote the multidimensional vectors describing all m objects by $X_1, X_2, ..., X_m$. $X_i = (x_{i1}, x_{i2}, ..., x_{in}),\ i = 1, ..., m$. A human being can comprehend visual information easier and more quickly than the numerical one. Data visualization allows people to detect

the presence of clusters, outliers or regularities in the analysed data. Various methods can be used for this purpose. It is possible to divide them into a few groups: (1) direct visualization methods (parallel coordinates, scatterplots, Chernoff faces, dimensional stacking, etc. (Hoffman and Grinstein, 2002)); (2) dimension reduction methods (principal component analysis (Taylor, 2003), projection pursuit (Brunsdon *et al*., 1998), multidimensional scaling (Borg and Groenen, 1997), etc.). There are methods based on neural networks (self-organizing maps (SOM) (Kohonen, 2001; Dzemyda and Kurasova, 2002), combination of the SOM and Sammon's mapping (Dzemyda and Kurasova, 2006), etc.) that may be allocated to the dimension reduction methods.

The multidimensional scaling method (MDS) (Borg and Groenen, 1997) is a popular method usable to visualize multidimensional data. It found a lot of applications in technology, economy, medicine, etc. For example, a two-dimensional view in the breast cancer data set (Bennett, 1992), where nine numerical parameters were measured on 683 patients, is presented in Fig. 1. We observe the interlocation of data on individual patients on a plane. Benign (non-malignant) cases are dark-coloured, while the malignant ones are light-coloured. Benign cases are concentrated in a small zone.

The problems with the standard MDS algorithm are faced when we have to project (visualize) a large data set or a new data point among the previously mapped points has to be projected. In the standard MDS, every iteration requires each point to be compared with all other points and the iteration complexity is $O(m^2)$. Thus, the MDS method is unsuitable for large data sets: it takes much computing time or there is not enough computing memory. Furthermore, it is necessary to recalculate the projection of all data points, when a point has to be mapped. Various methods have been proposed for mapping of new points without recalculating all the previously mapped points: Sammon's mapping (a particular application of MDS) based on an artificial neural network (SAMANN) (Mao and Jain, 1995), distance mapping (Pekalska *et al.*, 1999), incremental scaling (Basalaj, 1999), relative mapping (Naud and Duch, 2000), and neuroscale (Tiping, 1996).

The paper (Bernatavičienė *et al*., 2006a) focuses on the relative MDS method. This approach is to take a subset of initial data set (basis data set) and then map the basis data set, using the standard multidimensional scaling (MDS). As a second step, the remaining

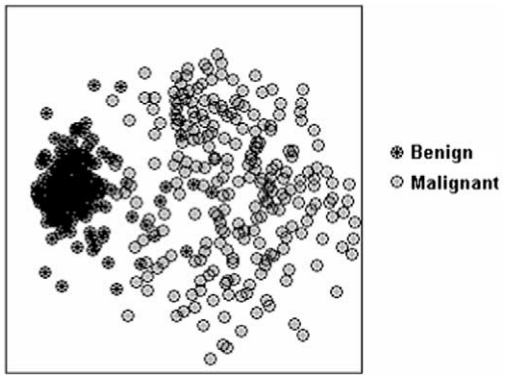

Fig. 1. Visualization of breast cancer data.

vectors of initial data are added to the basis layout using relative mapping (Naud and Duch, 2000). Strategies of selecting a set of basis vectors are analysed in (Bernatavičienė *et al*., 2006a). One strategy has been proposed and analysed in (Naud, 2004; Naud, 2006), that is based on the results of k-means algorithm. Two other superior strategies are proposed in (Bernatavičienė *et al*., 2006a). It has been defined that when selecting basis vectors at random from the initial set, one can get a similar result as that by using the strategy selecting basis vectors by the k-means clustering algorithm.

The aim of this paper is to determine the optimal number of basis vectors and the way of selecting them out of the whole set so that the calculation costs are low enough, while the projection is rather precise and informative. It is also of great importance to choose a proper way of two-dimensional vector initialization when using the relative MDS algorithm, because that influences the accuracy of projection. Next problem, which is examined in the paper, is to find out when it is reasonable to use the relative MDS algorithm instead of the standard MDS.

2. Background for the Relative MDS

The multidimensional scaling (MDS) is a group of methods that project multidimensional data onto a low- (usually two-) dimensional space and retain the interpoint distances among data as much as possible. Let us have vectors $X_i = (x_{i1}, x_{i2}, ..., x_{in}),\ i = 1, ..., m (X_i \in R^n)$. The pending problem is to get the projection of these n-dimensional vectors $X_i,\ i = 1, ..., m$ onto the plane R^2. The two-dimensional vectors $Y_1, Y_2, ..., Y_m \in R^2$ correspond to them. Here $Y_i = (y_{i1}, y_{i2}),\ i = 1, ..., m$. Denote the distance between the vectors X_i and X_j by d^*_{ij}, and the distance between the corresponding vectors on the projected space (Y_i and Y_j) by d_{ij}. In our case, the initial dimensionality is n, and the resulting one is 2. There exists a multitude of variants of MDS with slightly different so-called stress functions. In our experiments, the raw stress (1) is minimized.

$$E_{MDS} = \sum_{i,j=1,i<j}^{m} (d^*_{ij} - d_{ij})^2. \tag{1}$$

Various types of minimization of the stress function are possible (Borg and Groenen, 1997; Mathar and Zilinskas, 1993). In the original relative MDS algorithm (Naud, 2004), the minimization of the error E_{MDS} is realized through the steepest descent procedure that incorporates second order derivatives. We use here the SMACOF algorithm based on iterative majorization. It is one of the best optimisation algorithms for this type of minimization problem (Groenen and van de Vaelden, 2004). This method is simple and powerful, because it guarantees a monotone convergence of the stress function (Borg and Groenen, 1997; Groenen and van de Vaelden, 2004).

Relative mapping (Naud and Duch, 2000) is a part of the Relative MDS algorithm. In classification tasks, it may be of interest to see where a new data point "falls" among

the known cases and discover the class topology of its neighbouring known cases to get an insight on how a classifier would classify this new point. The realization of this purpose requires a method that allows the mapping of one new point on a set of data points previously mapped, using the topology-preserving mapping. The MDS is a topology preserving mapping, but it does not offer a possibility to project new points on the existing set of mapped points. To get a mapping that presents the previously mapped points together with the new ones requires a complete re-run of the MDS algorithm on the new and the old data points. Let us denote the number of known data points by N_{fixed}, the number of new data points by N_{new}, the total number of points considered during the mapping by N_{total} ($N_{total} = N_{fixed} + N_{new}$), the set of known data points by F (it will be called a set of basis vectors), the set of new data points by M. The algorithm scheme is as follows:

1. Map set F using the MDS mapping (the number of fixed points is equal to N_{fixed}).
2. Map set M with respect to the mapped set F, using the relative mapping (the number of new points is equal to N_{new}).

The difference between the relative mapping and the standard MDS is that during the minimization of the stress function, only the points from set M are allowed to move, while the points from set F are kept fixed. This is achieved by modifying the stress function so that it sums only over the distances that change during iterations, i.e., the distances between the fixed and the moving points, and interpoint distances between moving points. The stress function (1) is rewritten as:

$$E_{Relative\_MDS} = \sum_{i,j=1,i<j}^{N_{new}} (d^*_{ij} - d_{ij})^2 + \sum_{i=1}^{N_{new}} \sum_{j=N_{new}+1}^{N_{total}} (d^*_{ij} - d_{ij})^2. \tag{2}$$

In the original relative MDS algorithm (Naud, 2004), the minimization of the projection error $E_{\text{Relative\_}MDS}$ is also realized through the steepest descent procedure. However it is not that effective as compared with the Quasi–Newton algorithm. Therefore, in our experiments, we use the Quasi–Newton algorithm to minimize $E_{Relative\_MDS}$.

The k-means method is an iterative clustering algorithm in which the analysed vectors are moved among the sets of clusters until the desired set is reached (Dunham, 2003). Let us map the set of vectors to the ith cluster be $\{X_{i1}, X_{i2}, ..., X_{i\mu_i}\}$. Here μ_i is the number of the objects in the ith cluster $\{X_i^j = (x_{i1}^j, x_{i2}^j, ..., x_{in}^j),\ j = 1, ..., \mu_i\}$. The squared error is defined as:

$$E_k = \sum_{i=1}^{k} \sum_{j=1}^{\mu_i} ||X_i^j - C_i||^2. \tag{3}$$

Here $C_i = (c_{i1}, c_{i2}, ..., c_{in})$ is the centre of the cluster, ($c_{ik} = \frac{1}{\mu_i} \sum_{j=1}^{\mu_i} x_{ik}^j,\ k = 1, ..., n$). The above method will be employed in selecting basis vectors.

3. Data Sets for the Analysis

Five data sets were used in the experiments. Four of them are artificial, and the last one is real, namely:

1. *Ellipsoidal*[*m, n*] set, where $m = 1115$ is the number of vectors, $n = 50$ is the dimensionality; the set contains 20 overlapping ellipsoidal-type clusters.
2. *Ellipsoidal*[*m, n*] set , $m = 3140,\ n = 50$; the set contains 10 non overlapping ellipsoidal-type clusters.
3. *Gaussian*[*m, n*] set, where $m = 2729,\ n = 10$; it contains 10 overlapping clusters.
4. *Paraboloid*[*m, n*] set, where $m = 2583,\ n = 3$; the data set contains 2 non overlapping clusters.
5. *Abalone*[*m, n*] set, where $m = 4177,\ n = 8$ with contains 29 clusters.

In the experiments two data sets, obtained using the ellipsoidal cluster generator (Handl and Knowles, 2005) are used. This generator creates ellipsoidal clusters with the major axis of an arbitrary orientation. The boundary of a cluster is defined by four parameters:

- the origin (which is also the first focus);
- the interfocal distance, uniformly distributed in the range [1.0, 3.0];
- the orientation of the major axis, uniformly located amongst all orientations;
- the maximum sum of Euclidean distances to two foci, belonging to the range [1.05, 1.15] – equivalent to the eccentricity ranging from [0.870, 0.952].

For each cluster, data points are generated at a Gaussian-distributed distance from a uniformly random point on the major axis, in a uniformly random direction, and are rejected if they lie outside the boundary. Using this ellipsoidal generator two data sets are generated: *Ellipsoidal*[1115, 50] and *Ellipsoidal*[3140, 50].

The *Gaussian*[2729, 10] data set has been generated using the Gaussian cluster generator. This generator is based on a standard cluster model using multivariate normal distributions. See (Handl and Knowles, 2005) for more details.

The *Paraboloid*[2583, 3] data set is also an artificially created data set. There are two classes in this data set. The vectors are generated as follows: the first two coordinates of the vector are randomly generated in a predefined area (for the first class it is a circle with radius 0.4, for the second class this area is a ring, limited by two circles with radii 0.7 and 1.2). Then the third coordinate is added using such a rule $x_3 = 1.8 \cdot \sqrt{x_1^2 + x_2^2}$. The created paraboloid is rotated to make the classification more difficult.

The *Abalone*[4177, 8] data set is taken from the UCI repository (Blake and Mertz, 1998). Each vector describes 8 parameters of abalone: x_1 – length (the longest shell measurement), x_2 – diameter (perpendicular to the length), x_3 – height (with meat in the shell), x_4 – the whole weight of abalone, x_5 – shucked weight (the weight of meat), x_6 – viscera weight (gut weight after bleeding), x_7 – shell weight after being dried, and x_8 – rings. The data set samples are highly overlapping. Since the scales of parameters are different, it is necessary to normalize them: to calculate the average $\bar{x}_j$ and variance σ_j^2 of each parameter; the values of each parameter x_{ij} are normalized by the formula: $(x_{ij} - \bar{x}_j)/\sigma_j$.

4. Strategies for Selecting the Set of Basis Vectors

In the relative MDS, there a problem of selecting the set F of basis vectors arises. Some strategies can be used:

I. Set F consists of the cluster centres, obtained by the k-means clustering algorithm (Naud, 2004; Naud, 2006);

II. Set F consists of data set points that are the closest points to the cluster centres, obtained by the k-means algorithm. Additional points of each cluster are added to set F: these points are selected as most distant to the respective cluster centre (Bernatavičienė *et al*., 2006a).

III. Set F consists of data set points, chosen randomly from the whole (X_i, $i = 1, ..., m$) data set (Bernatavičienė *et al*., 2006a).

The general scheme of the visualization process is presented in Fig. 2.

Using Strategy I for selecting basis vectors in (Bernatavičienė *et al*., 2006a), we have obtained the poorest results; therefore we will not use it for the further research. It has been demonstrated that the visualization results by means of Strategies II and III are quite similar. However, clustering of the initial data and selection of basis vectors by Strategy II takes much more calculating time. By increasing the number of basis vectors we can get a more accurate projection error. According to (Naud, 2006) when using Strategy I, it is reasonable to increase the number of basis vectors up to 500. If the number of basis vectors is increasing even more, the clustering time is also increasing, while the calculations become slower. We can avoid that using Strategy III and increase the number of basis vectors up to 1000 or more, seeking for lower projection error. Due to these reasons, we will apply Strategy III for selecting basis vectors in further experiments.

When using relative MDS algorithm for visualization of large sets, it is of utmost importance to determine the optimal number of basis vectors, i.e., to establish how much their amount can be increased so as to avoid great calculation costs, the error remaining low enough and projection being informative.

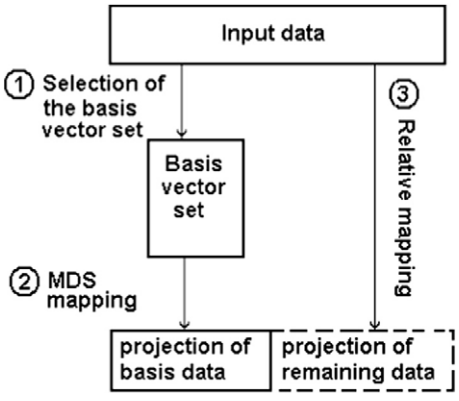

Fig. 2. Scheme of the visualization process: (1) selection the set of basis vectors; (2) the set of basis vectors is projected by standard MDS mapping; (3) the remaining points are projected by relative mapping.

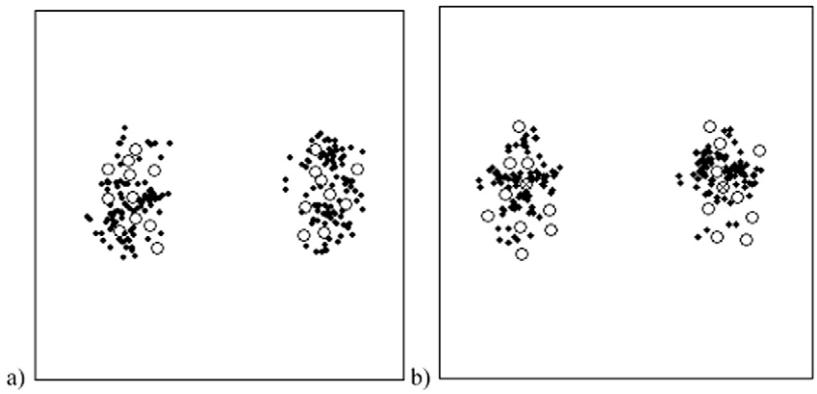

Fig. 3. Projections of two spheres: (a) using Strategy III, $E = 0.118037$, (b) using Strategy II, $E = 0.12265$.

In Fig. 3, the projections of two spheres (the data set comprised of the points of two hyper-spheres with 100 10-dimensional points in each sphere) are presented to illustrate these strategies. We do not present scales of variables in the figures with visualization results, because we are interested in observing the interlocation of points on a plane. This set is used to illustrate of strategies of selection of the basis vector set. First of all, set F of basis vector is comprised. Afterwards, the points of the formed basis set F are mapped on the plane, using the MDS algorithm. Using Strategy III, set F is made up out of 20 points from the data set chosen randomly (marked by unfilled circles in Fig. 3a). Using Strategy II, set F consists of two subsets: (a) two points (marked by circled crosses in Fig. 3b), which are closest to the centres of two clusters and (b) 9 points of each cluster (marked by unfilled circles in Fig. 3b). The visualization results, using Strategy I, are very similar to that of Fig 3a. The number of the basis vectors N_{fixed} is equal to 20 in both cases (Strategies II, and III). Then the remaining vectors (set M), marked by filled circles, are mapped by the relative MDS algorithm.

To compare the obtained visualization results, the projection error is calculated:

$$E = \sqrt{\sum_{i<j}^{m}(d_{ij}^{*} - d_{ij})^2 \Big/ \sum_{i<j}^{m}(d_{ij}^{*})^2}. \tag{4}$$

The projection error E in (4) is used here instead of E_{MDS} in (1), because the inclusion of the normalized parameter $\sum_{i<j}^{m}(d_{ij}^{*})^2$ gives a clear interpretation of the image quality that does not depend on the scale and the number of distances in an n-dimensional space. The reason for using E rather than the squared error E^2 is that E^2 is almost always very small in practice, so E values are easier to discriminate (Borg and Groenen, 1997). Of course, the error E in (4) may be used in the MDS. However, it is impossible to decompose and apply this error for the relative MDS. Therefore, E_{MDS} in (1) is minimized.

Using Strategy III, the projection error (4) is obtained smaller ($E = 0.11803$) than using Strategy II ($E = 0.12265$) (Fig. 3).

5. Results of Comparative Analysis

5.1. *Investigation of Initialization*

The visualization results strongly depend on the way of initializing the two-dimensional vectors. Initialization is assigning the initial values to two-dimensional vectors $Y_1, Y_2, ..., Y_m \in R^2$. With an aim to get as precise projection on the plane as possible, we have used the principal component analysis (PCA) algorithm for the initialization in the standard MDS algorithm (Bernatavičienė *et al*., 2006b). In the relative MDS, the basis vectors are projected using the standard MDS algorithm with PCA-based initialization. However, we have to find out which way of initialization to choose in order to project the remaining vectors on the fixed two-dimensional map of basis vectors using the relative mapping.

We have chosen 6 different initialization ways:

(a) the matrix $A[1 \times n]$ of average and rotation matrix $T[n \times 2]$, obtained by using PCA in basis vector initialization, are saved; two-dimensional coordinates of the remaining vectors are initialized by the formula: $Y_i = (X_i - A)T,\ i = 1, ..., m$;
(b) the initial coordinates of the vector from the remaining vector set is chosen as a two-dimensional projection of the closest basis vector;
(c) a random two-dimensional vector, generated in the area of projection of the nearest basis vector, is attributed to the initial coordinates of a vector from the remaining vector set (radius of the area $r = 0.01$);
(d) and (e) are analogous to c), only with different radii: (d) $r = 0.1$ and (e) $r = 1$;
(f) a random two-dimensional vector, generated in the area covered by all the two-dimensional projections of basis vectors, is attributed to the initial coordinates of a vector from the remaining vector set.

The experiments have been done on two data sets: *Ellipsoidal*[1115, 50] and *Gaussian*[2729, 10].

One experiment has been done with each (a) and (b) ways of initializing, 10 experiments have been done with each (c), (d), (e), (f) ways of initializing by selecting a different data set of basis vectors each time. The averages of projection errors have been calculated only for (c), (d), (e), (f) ways of initializing, because a certain randomness factor influenced the result of initialization.

The number of basis vectors varied from 100 to 1000 (step 100). 10 different sets of basis vectors of fixed size were formed (using Strategy III for selecting basis vectors). The data were projected on the plane employing all the 6 ways of initializing the two-dimensional vectors. The errors and their means have also been estimated. In fact, in the case of (c), (d), (e), (f), the means of errors obtained from 10 experiments for fixed size of basis vectors have been averaged. Tables 1 and 2 illustrate the results.

The experiments have shown that the worst way of initialization (f) is the random selection of basis vectors in the projection area (way (f)). Other ways (c), (d), (e) demonstrate similar results. Though the average of error is a little lower using the way (a) by PCA than that obtained by other ways (b), (c), (d), (e), the difference between these averages of projection errors is insignificant.

Table 1
Experimental results for the *ellipsoidal* [1115, 50] data set

	100		300		500		700		900	
	mean	variance	mean	variance	mean	variance	mean	variance	mean	variance
(a)	0.24609	0.00184	0.24261	0.00171	0.24103	0.00048	0.24061	0.00049	0.24023	0.00018
(b)	0.24620	0.00193	0.2426	0.00163	0.24103	0.00038	0.24059	0.00049	0.24023	0.00017
(c)	0.24617	0.00185	0.24261	0.00156	0.24103	0.00036	0.24059	0.00047	0.24023	0.00016
(d)	0.24616	0.00185	0.24261	0.00156	0.24103	0.00036	0.24059	0.00048	0.24023	0.00016
(e)	0.24636	0.00179	0.24266	0.00157	0.24105	0.00036	0.2406	0.00047	0.24023	0.00016
(f)	0.26150	0.00553	0.25252	0.00481	0.24683	0.00218	0.24384	0.00151	0.24231	0.00150

Table 2
Experimental results for the *Gausian*[2729, 10] data set

	100		300		500		700		900	
	mean	variance	mean	variance	mean	variance	mean	variance	mean	variance
(a)	0.28253	0.00640	0.27783	0.00493	0.27652	0.00511	0.27368	0.00061	0.27350	0.00052
(b)	0.28283	0.00685	0.27843	0.00507	0.27693	0.00516	0.27394	0.00065	0.27371	0.00041
(c)	0.28281	0.00647	0.27842	0.00482	0.27693	0.00492	0.27394	0.00062	0.27371	0.00039
(d)	0.28281	0.00647	0.27841	0.00483	0.27693	0.00492	0.27394	0.00063	0.27371	0.00039
(e)	0.28282	0.00647	0.27842	0.00483	0.27693	0.00492	0.27394	0.00063	0.27371	0.00039
(f)	0.28707	0.00733	0.28079	0.00516	0.27957	0.00505	0.27562	0.00090	0.27533	0.00103

With an increase of the number of basis vectors the variance decreases, while the variation of error is insignificant. This kind of regularity remains when using all the ways of initialization.

5.2. *Comparison of the Relative MDS and the Standard MDS*

The target of another research was to determine when it was expedient to use the relative MDS and when the standard one. We visualized 5 data sets of the different dimension and number of vectors. In the standard MDS algorithm the calculation time and projection error (4) were evaluated after each iteration. In the relative MDS algorithm, time and error (4) were evaluated after visualizing the whole data set, with the number of basis vectors varying from 100 to 1000 for small data sets, and from 100 to 1500 for large data sets (step 100) (Fig. 4). The number of basis vectors exceeds the above mentioned, the time for visualising basis vectors essentially increases, while the projection error diminishes insignificantly.

While using Strategy III, the experiments are done with the following number of the data set vectors chosen randomly: $N_{fixed} = 100, 200, ..., 1500$. Each experiment has been repeated 10 times with a different set of basis vectors, the projection error and calculating time being estimated each time. These errors and time, obtained in 10 exper-

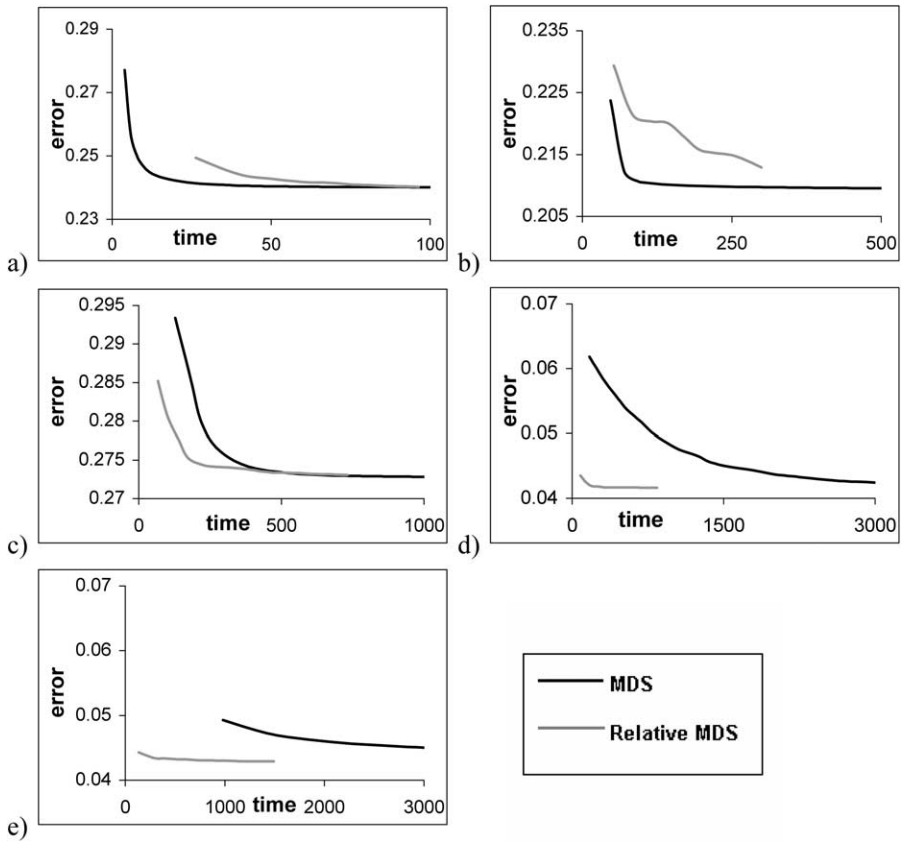

Fig. 4. Dependence of the projection error on the computing time: (a) *ellipsoidal*[1115, 50] data set; (b) *paraboloid*[2583, 3] data set; (c) *Gaussian*[2729, 10] data set; (d) *ellipsoidal*[3140, 50] data set; (e) *abalone*[4177, 8] data set.

iments, are averaged and presented in Fig. 4 (Relative MDS, grey line). The projection error and calculation time in each iteration are presented in Fig. 4 (standard MDS, black line) for each data set.

The results obtained confirm that it is reasonable to apply the relative MDS algorithm (Fig. 4 c, d, e) when visualizing the data set with more than 3000 vectors whose dimensionality exceeds 5. In these cases, given limited computing time the relative MDS algorithm yields a more precise mapping than the standard one. However, the number of visualizing vectors and dimension being small, the standard MDS is always better.

Fig. 5 illustrates the visualization results of the *ellipsoidal*[3140, 50] data set using both algorithms. When all the vectors ($m = 3140$) of the data set are mapped by MDS, the computing time is 7530 seconds, and $E = 0.04174$ (after 50 iterations). Using the relative MDS, the projection error is lower, and the computing time is saved significantly (842 seconds, $E = 0.04161$), The standard MDS takes 9 times more computing time. As seen from the projection by the standard MDS algorithm (Fig. 5a), one cluster is not

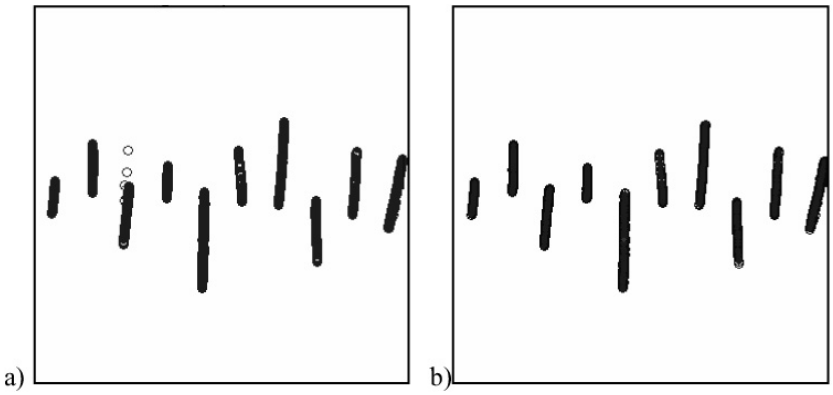

Fig. 5. Projection of the *ellipsoidal*[3140, 50] non overlapping data: (a) obtained using MDS (b) obtained using Relative MDS (number of basic vectors is equal to 1500).

Table 3

Error and computing time for both algorithms

Data set	MDS_50 iteration		Relative MDS	
	error	time, s	error	time, s
ellipsoidal [1115, 50]	0.240134	102	0.240232	96
ellipsoidal [3140, 50]	0.041745	7530	0.041615	842
Gaussian [2729, 10]	0.272748	1389	0.273075	732
paraboloid[2583, 3]	0.209169	1079	0.212924	300
abalone [4177, 8]	0.043681	24071	0.042907	1445

completely formed.

Table 3 presents the computing results after visualizing 5 data sets: error and time obtained by both algorithms. The standard MDS algorithm was terminated after 50 iterations. The same number of iterations was applied in the standard MDS algorithm when projecting basis vectors. The time in the relative algorithm was evaluated after all the projections of vectors of the whole data set have been obtained. The number of basis vectors is fixed, namely: for the *ellipsoidal* [1115, 50], and *paraboloid* [2583, 3] data sets it was equal to the 1000, while for others – 1500.

The results obtained confirm that with large datasets and limited computing time, the relative MDS algorithm yields a sufficiently precise projection and more time is saved.

5.3. *Selection of Basis Vectors*

The dependence of the projection error on the number of the basis vectors N_{fixed} is presented in Fig. 6. It shows that the averaged projection error E constantly decreases, when N_{fixed} increases. The averaged projection error E stabilizes itself at $N_{fixed} \approx 700$ for small data sets (from 1000 to 3000 vectors) (Fig. 6a) and at $N_{fixed} \approx 900$ for large

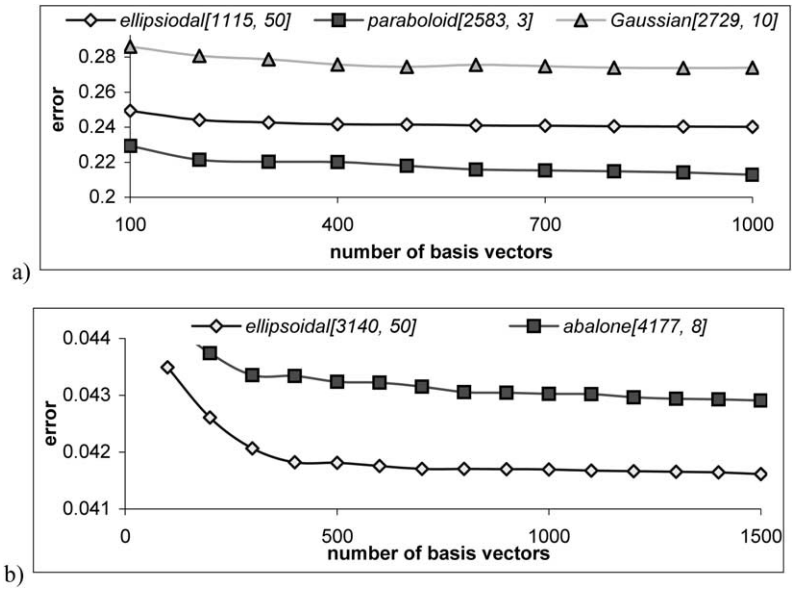

Fig. 6. Dependence of the projection error on the number of the basis vectors.

Table 4

Projection errors, obtained using Strategy II for the *ellipsoidal*[1115, 50] data set

p	k				
	10	**20**	**30**	**40**	**50**
5	0.3039	0.2640	0.2532	0.2499	0.2468
10	0.3045	0.2572	0.2469	0.2482	0.2448
15	0.2991	0.2576	0.2459	0.2448	0.2430
20	0.3038	0.2555	0.2446	0.2434	0.2413
25	0.2982	0.2503	0.2436	0.2427	0.2408

data sets (more than 3000 vectors) (Fig. 6b). By increasing the number N_{fixed} even more the projection error (4) changes insignificantly: its difference is observed only in the 4–5 th digit after a point.

The aim of the next experiment is to show how important it is to select the basis vectors uniformly distributed all over the data set. We use Strategy II for selecting basis vectors, in which the centres of clusters, obtained by the k-means algorithm, are rather uniformly distributed throughout the data set. Meanwhile, when taking additional vectors of each cluster, the uniformity of distribution of the basis vectors retains.

The projection errors, obtained by using a different number of N_{fixed} (the number of clusters $k = 10, 20, ..., 50$, and the number of the data set vectors from each cluster $p = 5, 10, ..., 25$), are presented in Table 4. When composing a data set of basis vectors, it is better to take a large number of clusters and less additional vectors from each cluster.

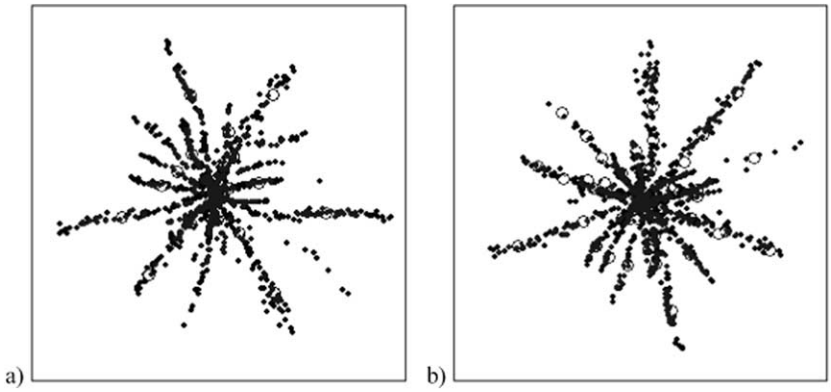

Fig. 7. Projection of the *ellipsoidal*[1115, 50] data set: (a) $k = 20,\ p = 25,\ E = 0.2503$, (b) $k = 50,\ p = 10,\ N_{fixed} = 500,\ E = 0.2448$.

In this way, the computing time grows but we distribute the basis vectors all over the whole data set more uniformly. For instance, if we take the fixed number of basis vectors $N_{fixed} = 200$ (Table 4), as $k = 10$, and $p = 20$, then the projection error is equal $E = 0.3038$, while for $k = 20$, and $p = 10,\ E = 0.2572$. Thus, the experiment demonstrates that the more uniform the distribution of the basis vectors is, the more precise the projection is. Fig. 7 illustrates these results.

In Fig. 7, the visualization results of the *ellipsoidal*[1115, 50] data set are presented (number of basis vectors is equal $N_{fixed} = 500$): (a) $k = 20,\ p = 25$, (b) $k = 50,\ p = 10$. The lower projection error is obtained and the quality of visualization is better in case (b) using the relative algorithm.

6. Conclusions

The visual analysis of large data sets is a topical problem. However, when a large data set of multidimensional vectors is visualized by the standard MDS method, it takes much computing time. In this paper, we have investigated a modification of the standard MDS method for large data sets (the relative MDS): first of all, some basis vectors are projected onto the plane, then the remaining points are projected among the previously mapped vectors.

The investigation allows us to draw the following conclusions:

- The visualization results are very dependant on the selected set of the basis vectors.
- With an aim to obtain a more precise projection of whole data set, we propose to apply the PCA algorithm for the initialization of two-dimensional vectors, corresponding to the remaining n-dimensional points. However, the differences of visualization results obtained by all five investigated ways of initialization are not so significant. The worst way of initialization is a generation of random two-dimensional vectors in the area covered by all the two-dimensional projections of basis vectors.

- In visualizing data sets that dimensionality is larger than 5 and that contain more than 3000 vectors, it is more reasonable to use the relative MDS algorithm. Under the above mentioned conditions, the relative MDS algorithm gives precise mapping and saves much computing time as compared with the standard MDS algorithm. Therefore, in the case of limited computing time, the projection by the relative MDS algorithm will be better than that by the standard MDS algorithm.
- The larger dimensionality of visualized vectors needs the larger number of the basis vectors.
- When the number of the basis vectors increases, a more precise projection is obtained. However, too large number of the visualized basis vectors extends the computing time. The optimal number of basis vectors ranges from 700 to 1000 for small data sets (up to 3000), while for larger than 3000 data sets it ranges from 900 to 1500.
- With an increase of number of basis vectors, the mean value of the projection error decreases; the variance of the projection error decreases significantly.
- The basis vectors should be selected so that the basis vectors were distributed as uniformly as possible all over the data set, which shows better results of obtained visualization.

References

Basalaj, W. (1999). Incremental multidimensional scaling method for database visualization. *Proceedings of Visual Data Exploration and Analysis VI*, SPIE, **3647**, 149–158.

Bennett, K.P., and O.L. Mangasarian (1992). Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, **1**, 23–34 (Gordon & Breach Science Publishers). `http://www.ics.uci.edu/~mlearn/databases/breast-cancer-wisconsin/breast-cancer-wisconsin.names`

Bernatavičienė, J., G. Dzemyda, O. Kurasova and V. Marcinkevičius (2006a). Strategies of selecting the basis vector set in the relative MDS. *Technological and Economic Development of Economy*, **12**(4), 283–288. `http://www.tede.vgtu.lt`

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

Blake, C.L., and C.J. Mertz (1998). *UCI Repository of Machine Learning Databases*. Irvine, CA: University of California, Department of Information and Computer Science.

Brunsdon, C., A.S. Fotheringham and M.E. Charlton (1998). An investigation of methods for visualising highly multivariate datasets. In D. Unwin, P. Fisher (Eds.) *Case Studies of Visualization in the Social Sciences*. pp. 55–80.

Borg, I., and P. Groenen (1997). *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York.

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

Dzemyda, G., and O. Kurasova (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, Inc. (Prentice Hall).

Groenen, P.J.F., and M. van de Vaelden (2004). Multidimensional scaling. *Econometric Institute Report EI2004-15*. `https://ep.eur.nl/handle/1765/1274/1/ei200415.pdf`

Handl, J., and J. Knowles (2005). *Cluster Generators for Large High-Dimensional Data Sets with Large Numbers of Clusters*. `http://dbkgroup.org/handl/generators/`

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

Keim, D.A. (2001). Visual exploration of large data sets. *Communications of the ACM*, **44**(8), 38–44.

Kohonen, T. (2001). *Self-Organizing Maps*. 3rd edn. Springer Series in Information Sciences, **30**.

Mao, J., and A.K. Jain (1995). Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, **6**(2), 296–317.

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

Naud, A., and W. Duch (2000). Interactive data exploration using MDS mapping. *Proceedings of the Fifth Conference: Neural Networks and Soft Computing*. pp. 255–260.

Naud, A. (2004). Visualization of high-dimensional data using a association of multidimensional scaling to clustering. *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, **1**, 252–255.

Naud, A. (2006). An accurate MDS-based algorithm for the visualization of large multidimensional datasets. *Lecture Notes in Computer Science*, **4029**, 643–652.

Pekalska, E., D. de Ridder, R.P. Duin and M.A. Kraaijveld (1999). A new method of generalizing Sammon mapping with application to algorithm speed-up. In M. Boasson, J. Karndorp, J. Torino, M. Vosselman (Eds.), *Proceedings of 5th Annual Conference of the Advanced School for Computing and Image (ASCI'99)*.

Taylor, P. (2003). Statistical methods. In M. Berthold, D.J. Hand, (Eds.), *Intelligent Data Analysis: An Introduction*. Springer-Verlag. pp. 69–129.

Tiping, M.E. (1996). Topographic mappings and feed-forward neural networks. *PhD Thesis*, Aston University, Birmingham, UK.

J. Bernatavičienė graduated from the Vilnius Pedagogical University in 2004 and received the degree of master of informatics. She is a PhD student at the Institute of Mathematics and Informatics from 2004 to 2008. She is the software engineer in the System Analysis Department of IMI and the lecturer in Vilnius Pedagogical University. Her research interests include data bases, data mining, visualization, and internet technologies.

G. Dzemyda graduated from Kaunas University of Technology, Lithuania, in 1980, and in 1984 received there the doctoral degree in technical sciences (PhD) after post-graduate studies at the Institute of Mathematics and Informatics, Vilnius, Lithuania. In 1997 he received the degree of Doctor Habilius from Kaunas University of Technology. He was conferred the title of professor (1998) at Kaunas University of Technology. He is a director of the Institute of Mathematics and Informatics and heads the System Analysis Department of the institute. The areas of research are the theory, development and application of optimization, and the interaction of optimization and data analysis. The interests include optimization theory and applications, data mining in databases, multiple criteria decision support, neural networks, parallel optimization, Internet databases, the models of epidemic spread.

V. Marcinkevičius graduated from the Vilnius Pedagogical University in 2003 and received the degree of bachelor of mathematics and informatics (2001) and master of mathematics. He is the software engineer and PhD student in the System Analysis Department of IMI. His research interests include data bases, internet technologies and parallel algorithms.

Reliatyvaus daugiamačių skalių algoritmo efektyvaus veikimo sąlygos

Jolita BERNATAVIČIENĖ, Gintautas DZEMYDA, Virginijus MARCINKEVIČIUS

Šiame straipsnyje tiriamas reliatyvus daugiamačių skalių algoritmas, skirtas didelių daugiamačių duomenų aibių vizualizavimui. Šis algoritmas sudarytas iš kelių dalių: daugiamačiai duomenys dalijami į dvi aibes (bazinių vektorių aibę ir likusių taškų aibę); baziniai vektoriai, naudojant standartinį daugiamačių skalių algoritmą, projektuojami į plokštumą, o likusių vektorių aibė projektuojama į plokštumą, atsižvelgiant tik į bazinių vektorių projekcijas. Vizualizavimo kokybė, naudojant šį algoritmą, labai priklauso nuo dvimačių vektorių inicijavimo būdo, nuo bazinių vektorių skaičiaus bei jų parinkimo strategijų. Straipsnyje tiriamos trys bazinių vektorių parinkimo strategijos, ieškoma optimalaus bazinių vektorių skaičiaus, lyginami skirtingi dvimačių vektorių inicijavimo būdai. Eksperimentai parodė, kad straipsnyje siūlomi sprendimai pagerina vizualizavimo kokybę realiatyvaus daugiamačių skalių algoritmu ir taupo skaičiavimo laiką.