On the Computational Efficiency of Geometric Multidimensional Scaling

Gintautas Dzemyda*
Martynas Sabaliauskas*
gintautas.dzemyda@mif.vu.lt
martynas.sabaliauskas@mif.vu.lt
Vilnius University Institute of Data Science and Digital technologies
Vilnius, Lithuania

ABSTRACT

Real-life applications often deal with multidimensional data. In the general case, multidimensional data means a table of numbers whose rows correspond to different objects and columns correspond to features characterizing the objects. Usually, the number of objects is large, and the dimensionality (number of features) is greater than it is possible to represent the objects as points in 2D. The goal is to reduce the dimensionality of data to such one that objects, characterized by a large number of features or by proximities between pairs of the objects, be represented as points in lower-dimensional space or even on a plane. Multidimensional scaling (MDS) is an often-used method to reduce the dimensionality of multidimensional data nonlinearly and to present the data visually. MDS minimizes some stress function. We have proposed in [8] and [9] to consider the stress function and multidimensional scaling, in general, from the geometric point of view, and the so-called Geometric MDS has been developed. Geometric MDS allows finding the proper direction and step size forwards the minimum of the stress function analytically. In this paper, we disclose several new properties of Geometric multidimensional scaling and compare the simplest realization (GMDS1) of Geometric MDS experimentally with the well-known SMACOF version of MDS.

CCS CONCEPTS

• **Human-centered computing** → **Visualization**; • **Mathematics of computing**; • **Theory of computation** → **Theory and algorithms for application domains**;

KEYWORDS

Visualization, Multidimensional Scaling, MDS, SMACOF, Geometric MDS

*Both authors contributed equally to this research.

ESSE 2021, November 6–8, 2021, Larissa, Greece

ACM ISBN 978-1-4503-8506-0/21/11...$15.00
https://doi.org/10.1145/3501774.3501794

ACM Reference Format:
Gintautas Dzemyda and Martynas Sabaliauskas. 2021. On the Computational Efficiency of Geometric Multidimensional Scaling. In *2021 2nd European Symposium on Software Engineering (ESSE 2021), November 6–8, 2021, Larissa, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3501774.3501794

1 INTRODUCTION

Visualization methods are recent techniques to discover knowledge hidden in multidimensional data sets. In the general case, multidimensional data means a table of numbers whose rows correspond to different objects and columns correspond to features characterizing the objects. Usually, the number of objects is large, and the dimensionality (number of features) is greater than it is possible to represent the objects as points in 2D. The goal is to reduce the dimensionality of data to such one that objects, characterized by a large number of features or by proximities between pairs of the objects, be represented as points in lower-dimensional space or even on a plane. The proper reduction of dimensionality must be such that after representation of the multidimensional data in a low-dimensional space the certain properties (e.g. clusters, outliers) of the structure of the data set are preserved as faithfully as possible. A large number of methods have been developed for multidimensional data visualization, including wide applications (see e.g. [1–3, 5, 7, 13–17, 19, 20, 24]). Multidimensional scaling (MDS) is one of the most popular methods for multidimensional data dimensionality reduction and visualization [3, 7].

Consider the multidimensional data set as a matrix $X = \{X_i = (x_{i1}, \dots, x_{in}),\ i = 1, \dots, m\}$ of n-dimensional data points $X_i \in \mathbb{R}^n$, $n \geqslant 3$. Data point $X_i = (x_{i1}, \dots, x_{in})$ is the result of observation of some object characterized by n features. Dimensionality reduction means finding the set of coordinates (layout) of points $Y_i = (y_{i1}, \dots, y_{id})$, $i = 1, \dots, m$, in a lower-dimensional space ($d < n$), where the particular point $X_i = (x_{i1}, \dots, x_{in}) \in \mathbb{R}^n$ is represented by $Y_i = (y_{i1}, \dots, y_{id}) \in \mathbb{R}^d$. If $d \leqslant 3$, dimensionality reduction results may be presented visually for human decision.

In [21], an example of dimensionality reduction and visualization using MDS is presented by visualizing 5-dimensional $n = 5$ data of customers shopping behaviour during an online advertising campaign. MDS was applied to reduce the dimensionality to 2 ($d = 2$). The results are presented in Fig. 1. After dimensionality reduction from $n = 5$ to $d = 2$, one point on a plane corresponds to one particular sale, i.e. the dimensionality was reduced nonlinearly from 5 features to 2 new features. The number of objects remains the same (2644), but the dimensionality is 2. The total number of points in Fig. 1 is 2644 – one point for one sale. We do not present legends and units for both axes in Fig. 1 with visualization results because we are interested in observing the interlocation of $m = 2644$ sales and finding some regular structures of data. Results in Fig. 1 allow deciding on the efficiency of the campaign. We see two clusters of sales and some regular structures inside the clusters. More interpretations of the results are given in [21].

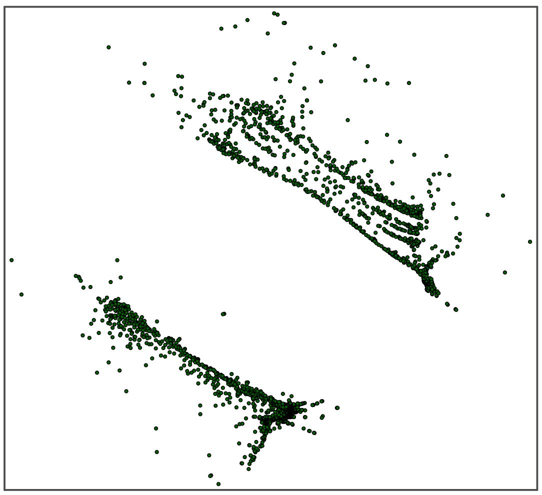

Figure 1: Example of dimensionality reduction, data of customers shopping behavior during an online advertising campaign

Data for MDS is the symmetric $m \times m$ matrix $\mathbf{D} = \{d_{ij}, i, j = 1, \ldots, m\}$ of proximities, where $d_{ij} = d_{ji}$. MDS tries to hold the proximities [7] d_{ij} between pairs of multidimensional points X_i and X_j, $i, j = 1, \ldots, m$, as much as possible. Proximity d_{ij} can be the distance between points X_i and X_j, i.e. proximities may be computed from a table of numbers, whose rows correspond to different objects and columns correspond to features characterizing the objects.

MDS looks for coordinates of points Y_i representing X_i in a lower-dimensional Euclidean space $\mathbb{R}^d$ by minimizing some stress function. Several realizations of MDS with different stress functions have been proposed (see review in [7]), seeking less dependence of the resulting stress value on the magnitude of the proximities (dissimilarities). However, their minimization is more complicated.

We have proposed a new approach, Geometric MDS, in [8] and [9]. We have disclosed its unique properties in [10] and [23]. The advantage of Geometric MDS is that it can use the simplest stress function, and there is no need for its normalization depending on the number of data points and the scale of proximities. The performance was compared with multidimensional scaling using majorization (SMACOF [4]). It is shown in [9] that Geometric MDS does not depend on the scales of dissimilarities and, therefore, may use a much simpler stress function like the raw stress function [18]:

$$S(Y_1, \ldots, Y_m) = \sum_{i=1}^{m} \sum_{j=i+1}^{m} (d_{ij} - d_{ij}^*)^2, \tag{1}$$

where d_{ij}^* is the Euclidean distance between points Y_i and Y_j in a lower-dimensional space:

$$d_{ij}^* = \sqrt{\sum_{l=1}^{d} \left(y_{il} - y_{jl}\right)^2}. \tag{2}$$

The optimization problem is to find a minimum of the function $S(\cdot)$, defined by (1), and optimal coordinates of points $Y_i = (y_{i1}, \ldots, y_{id})$, $i = 1, \ldots, m$:

$$\min_{Y_1, \ldots, Y_m \in \mathbb{R}^d} S(Y_1, \ldots, Y_m). \tag{3}$$

The number of variables in problem (3) is $m \times d$. This number is very large, often. Therefore, special methods to solve such optimization problem are developed ([2, 3, 7]). Moreover, the problem is multiextremal usually.

The motivation of this research is to compare the simplest realization of Geometric MDS (denote it by GMDS1) with the well-known SMACOF version of MDS from the point of view of minimal reached stress value and used computing time.

2 OVERVIEW OF GEOMETRIC MDS

Geometric MDS for minimization of the stress function (1) proposed in [8] and [9] is reviewed in this section briefly.

Let's have some initial configuration of points $Y_1, \ldots, Y_m$. The main idea of Geometric MDS focuses on optimizing the position of one chosen point (let it be Y_j) when the positions of the remaining points $Y_1, \ldots, Y_{j-1}, Y_{j+1}, \ldots, Y_m$ are fixed. In the case of optimization of one point position, we minimize a part of the global stress $S(\cdot)$ in (3). This part is named by the local stress function $S^*(\cdot)$ in [9]. $S^*(\cdot)$ depends on Y_j, only:

$$S^*(Y_j) = \sum_{\substack{i=1 \\ i \neq j}}^{m} \left(d_{ij} - \sqrt{\sum_{l=1}^{d} \left(y_{il} - y_{jl}\right)^2} \right)^2. \tag{4}$$

Denote a new position of Y_j by Y_j^*. Let Y_j^* be chosen so that

$$Y_j^* = \frac{1}{m-1} \sum_{\substack{i=1 \\ i \neq j}}^{m} A_{ij}, \tag{5}$$

where the point A_{ij} lies on the line between Y_i and Y_j, $i \neq j$, in a distance d_{ij} from Y_i.

Equation (5) is the main formulae of Geometric MDS, when defining transition from Y_j to its new position Y_j^*. It is reasoned in [9]. The transition from Y_j to its new position Y_j^*, when positions of the remaining points $Y_1, \ldots, Y_{j-1}, Y_{j+1}, \ldots, Y_m$ are fixed, is called by geometric step from the point Y_j. After the transition, we get new coordinates of the point Y_j, i.e. the set $Y_1, \ldots, Y_m$ is being updated.

Eight propositions of this section are proved in [9]. They disclose the advantages of the transition defined by (5).

Proposition 1. *The gradient of local stress function $S^*(\cdot)$ is as follows:*

$$\nabla S^*|_{Y_j} = \left(2 \sum_{\substack{i=1 \\ i \neq j}}^{m} \frac{d_{ij} - d^*_{ij}}{d^*_{ij}} \left(y_{ik} - y_{jk} \right),\ k = 1, \dots, d \right). \quad (6)$$

Proposition 2. *The coordinates of point Y^*_j are equal to:*

$$\left(\frac{1}{m-1} \sum_{\substack{i=1 \\ i \neq j}}^{m} \left(\frac{d_{ij} \left(y_{jk} - y_{ik} \right)}{d^*_{ij}} + y_{ik} \right),\ k = 1, \dots, d \right). \quad (7)$$

Proposition 3. *The step direction from Y_j to Y^*_j corresponds to the anti-gradient of the function $S^*(\cdot)$ at the point Y_j:*

$$Y^*_j = Y_j - \frac{1}{2(m-1)} \nabla S^*|_{Y_j}. \quad (8)$$

Proposition 4. *Size of a step from Y_j to Y^*_j is equal to*

$$\frac{1}{m-1} \sqrt{\sum_{k=1}^{d} \left(\sum_{\substack{i=1 \\ i \neq j}}^{m} \frac{d_{ij} - d^*_{ij}}{d^*_{ij}} \left(y_{ik} - y_{jk} \right) \right)^2}.$$

Proposition 5. *Let Y_j does not match to any local extreme point of the function $S^*(\cdot)$. If Y^*_j is chosen by (5), then a single step from Y_j to Y^*_j reduces a local stress $S^*(\cdot)$:*

$$S^*(Y^*_j) < S^*(Y_j).$$

Proposition 6. *The value of the local stress function $S^*(\cdot)$ (4) will converge to a local minimum when repeating steps (8) and $Y_j \leftarrow Y^*_j$.*

Proposition 7. *Let Y_j does not match to any local extreme point of the function $S^*(\cdot)$. Movement of any projected point by the geometric method reduces the stress (1) of MDS: if Y^*_j is chosen by (5), then the stress function $S(\cdot)$, defined by (1), decreases:*

$$S(Y_1, \dots, Y_{j-1}, Y^*_j, Y_{j+1}, \dots, Y_m) < S(Y_1, \dots, Y_{j-1}, Y_j, Y_{j+1}, \dots, Y_m).$$

Proposition 8. *The local stress function $S^*(\cdot)$ defined by (4) could be multimodal for dimensionality $1 \leqslant d < \infty$.*

3 NEW PROPERTIES OF GEOMETRIC MDS

Proposition 9. *The weight center C_Y of points $Y_1, \dots, Y_m$ changes its position after the geometric step from point Y_j if $Y_j \neq Y^*_j$, $1 \leqslant j \leqslant m$.*

Proof. The point Y_j moves to the new position Y^*_j after the geometric step. Before the geometric step, the weight center C_Y of points $Y_1, \dots, Y_m$ is equal to

$$\frac{1}{m} \sum_{i=1}^{m} Y_i. \quad (9)$$

After the geometric step, the weight center of points $Y_1, \dots, Y_{j-1}, Y^*_j, Y_{j+1}, \dots, Y_m$ is equal to

$$\frac{1}{m} \left(Y^*_j + \sum_{\substack{i=1 \\ i \neq j}}^{m} Y_i \right). \quad (10)$$

By subtracting (9) from (10), we get that after the single geometric step from point Y_j, the weight center C_Y moves by vector

$$\frac{1}{m} \left(Y^*_j + \sum_{\substack{i=1 \\ i \neq j}}^{m} Y_i \right) - \frac{1}{m} \sum_{i=1}^{m} Y_i$$
$$= \frac{1}{m} \left(Y^*_j + \sum_{\substack{i=1 \\ i \neq j}}^{m} Y_i \right) - \frac{1}{m} \left(Y_j + \sum_{\substack{i=1 \\ i \neq j}}^{m} Y_i \right)$$
$$= \frac{1}{m} \left(Y^*_j - Y_j \right).$$

□

Proposition 10. *The weight center C_Y of points $Y_1, \dots, Y_m$ changes to*

$$\frac{1}{m(m-1)} \sum_{\substack{i=1 \\ i \neq j}}^{m} \left(m Y_i + (Y_j - Y_i) \frac{d_{ij}}{d(Y_i, Y_j)} \right) \quad (11)$$

after the geometric step from point Y_j, $1 \leqslant j \leqslant m$.

Proof. According to proposition 9, the weight center becomes equal to

$$C_Y + \frac{1}{m} \left(Y^*_j - Y_j \right) \quad (12)$$

after the geometric step from point Y_j. Since the point A_{ij} lies on the line between Y_i and Y_j, $1 \leqslant i, j \leqslant m$, $i \neq j$, in a distance d_{ij} from Y_i, then

$$A_{ij} = Y_i + (Y_j - Y_i) \frac{d_{ij}}{d(Y_i, Y_j)}. \quad (13)$$

By substituting (13) into (5), we get

$$Y^*_j = \frac{1}{m-1} \sum_{\substack{i=1 \\ i \neq j}}^{m} \left(Y_i + (Y_j - Y_i) \frac{d_{ij}}{d(Y_i, Y_j)} \right). \quad (14)$$

By substituting (14) into (12), we get the new weight center:

$$\frac{1}{m} \left(Y^*_j - Y_j \right) + C_Y$$
$$= \frac{1}{m} \left(Y^*_j - Y_j \right) + \frac{1}{m} \sum_{i=1}^{m} Y_i$$
$$= \frac{1}{m} \left(Y^*_j + \sum_{\substack{i=1 \\ i \neq j}}^{m} Y_i \right)$$
$$= \frac{1}{m} \left(\frac{1}{m-1} \sum_{\substack{i=1 \\ i \neq j}}^{m} \left(Y_i + (Y_j - Y_i) \frac{d_{ij}}{d(Y_i, Y_j)} \right) + \frac{1}{m-1} \sum_{\substack{i=1 \\ i \neq j}}^{m} (m-1) Y_i \right)$$
$$= \frac{1}{m(m-1)} \sum_{\substack{i=1 \\ i \neq j}}^{m} \left(m Y_i + (Y_j - Y_i) \frac{d_{ij}}{d(Y_i, Y_j)} \right).$$

□

PROPOSITION 11. *The weight center C_Y of points $Y_1, \ldots, Y_m$ changes after m geometric steps consecutively performed from all m points $Y_1, \ldots, Y_m$.*

The proposition has been proved experimentally by making any of m consequential geometric steps of points Y_i, $1 \leqslant i \leqslant m$. The experiments (counterexamples) show that the weight center also depends on the order in which points Y_i, $1 \leqslant i \leqslant m$ are selected.

4 COMPARISON OF EFFICIENCY OF THE SIMPLEST REALIZATION OF GMDS WITH SMACOF

Two algorithms realizing the idea of Geometric MDS are presented in [9]. The simplest way to minimize the stress $S(\cdot)$ by Geometric MDS is a consecutive changing of positions of points $Y_1, \ldots, Y_m$ many times. Denote this realization of Geometric MDS by GMDS1.

One iteration of GMDS1 consists of a consecutive changing of positions of all points $Y_1, \ldots, Y_m$ once, starting from Y_1 and finishing by Y_m. The stress is minimized, namely by a consequent changing the positions of separate d-dimensional projected points. In more detail, the simplest realization [9] of Geometric MDS recalculates the coordinates of a single d-dimensional point Y_j at each step. The result is a new point Y_j^*. The properties of such transition from Y_j to Y_j^* are disclosed on Propositions proved in [9] and given above. The raw stress function (1) decreases after movement Y_j to Y_j^* (see Proposition 7). The position of each selected point Y_j is changed once consecutively. When we recalculate coordinates of one d-dimensional point (e.g. of Y_j) in Geometric MDS, we consider coordinates of all m points $Y_1, \ldots, Y_m$. After recalculation, we will have an updated set $Y = \{Y_1, \ldots, Y_m\}$, where Y_j will be different as compared with the previous content of this set, and coordinates of all other points $Y_i, i = 1, \ldots, m, i \neq j$, remain not changed. Afterwards, when we recalculate coordinates of another d-dimensional point (e.g. of $Y_i, i \neq j$), we will use all $Y_1, \ldots, Y_m$ including the updated point Y_j.

In this section, we compare the simplest realization (GMDS1) of Geometric MDS experimentally with the SMACOF version of MDS realized in R [22]. GMDS1 is realized in Python.

Experiments were carried out using real data set on 40 regions of 8 Eastern European countries. Description of data is given in [6]. Data are collected from the Eurostat database [11]. Therefore, the number of data points is equal to 40 ($m = 40$), and data is 11-dimensional ($n = 11$). The multidimensional data is normalized, and the visualization is applied in [11]. The matrix $X = \{X_i = (x_{i1}, \ldots, x_{in}),\ i = 1, \ldots, m\}$ before and after normalization by z-score is available online in MIDAS archive [12]. We use data before normalization here. In our case, projections of 11-dimensional data to the plain were chosen, i.e. $d = 2$. So, optimization problem (3) has $40 \times 2 = 80$ variables in this case. The result (minimal obtained value of the stress function) depends on the starting coordinates of points $Y_i = (y_{i1}, \ldots, y_{id}), i = 1, \ldots, m$. Therefore, problem (3) was solved many times using different starting coordinates of the points, and results were averaged for the objectivity and reliability of the results.

1000 experiments of minimization of the stress function $S(\cdot)$, defined by (1), were carried using different starting coordinates of points $Y_i = (y_{i1}, \ldots, y_{id}), i = 1, \ldots, m$, generated at random in the interval (0; 1). In our case projections of 11-dimensional data to the plain was chosen, i.e. $d = 2$.

Table 1: Comparison of time performance

Algorithm	Time, s	Programming language
SMACOF	0.001540	R
GMDS1	0.002333	Python

SMACOF optimizes the coordinates of points $Y_i = (y_{i1}, \ldots, y_{id})$, $i = 1, \ldots, m$, during each iteration. GMDS1 optimizes coordinates of a particular point Y_i. One iteration of GMDS1 means the run of i value from 1 to m. In this way, GMDS1 optimizes coordinates of all points Y_i during each iteration like SMACOF.

In each of 1000 experiments, the value of the stress function $S(\cdot)$ was measured, and the results were averaged. The average results are presented in Figures 2, 3, 4. Both linear and logarithmic scales were used to clarify results when presenting the average stress during the first ten iterations. A logarithmic scale was used because the values of stress function vary in a large range.

Moreover, the computer time used for one iteration of both program realizations (GMDS1 and CMACOF) was measured. The average results are presented in 1. Let us note that the SMACOF algorithm has been developed for a long time, and there are many implementations of it that are optimized for performance. Geometric MDS is a new method, and research is needed to increase the efficiency of its implementation, including the use of other programming tools, e.g. C++ or R.

5 CONCLUSIONS

Geometric MDS is a new method for multidimensional data visualization that extends the understanding of multidimensional scaling (MDS). As a result, new realizations of MDS have been proposed recently ([8–10, 23]). These realizations are based on the ideas of Geometric MDS.

The paper presents both theoretical and experimental findings on Geometric MDS. Several new properties of Geometric MDS have been discovered theoretically. Experiments show that SMACOF is more effective than the simplest realization of Geometric MDS (GMDS1) in the first iteration only. Then GMDS1 outperforms the efficiency of SMACOF. GMDS1 and SMACOF find similar small values of the stress function when the number of iterations is large because the optimization problem is not very difficult, i.e. 80 variables, only. The results on computer time consumption are quite relatively because realizations were made using different means - R and Python.

This paper expands our knowledge of the ideas and capabilities of Geometric MDS for the future development of a class of new effective and practical algorithms for multidimensional data visualization, including realizations for big data.

ACKNOWLEDGMENTS

This research has received funding from the Research Council of Lithuania (LMTLT), agreement No S-MIP-20-19.

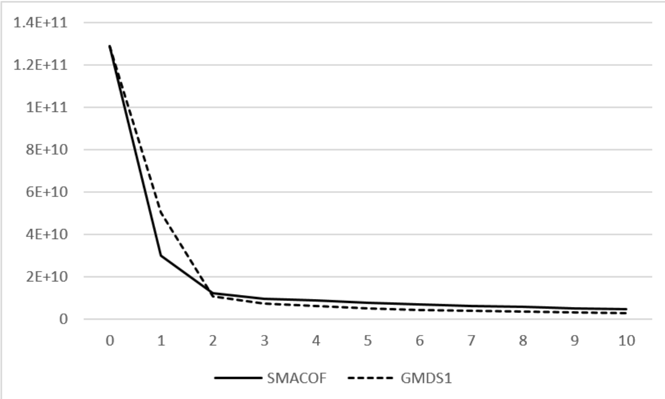

Figure 2: The average stress during the first 10 iterations, linear scale

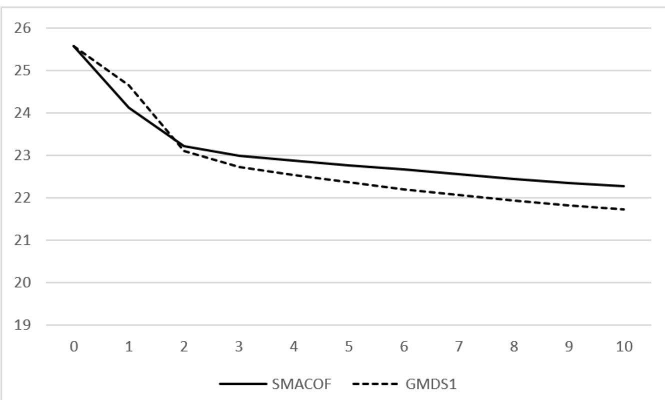

Figure 3: The average stress during the first 10 iterations, logarithmic scale

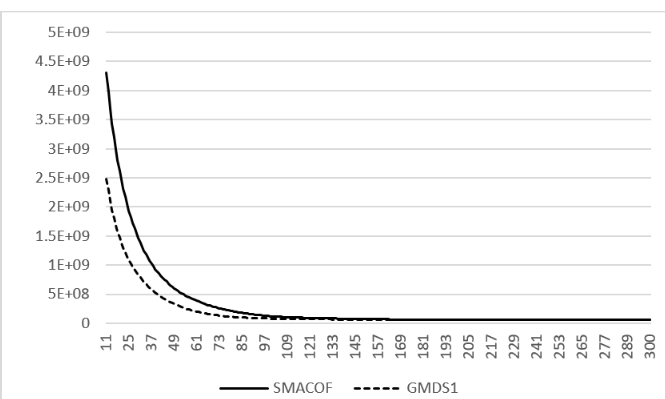

Figure 4: The average stress during 11-300 iterations, linear scale

REFERENCES

[1] Jolita Bernataviciene, Gintautas Dzemyda, Olga Kurasova, Virginijus Marcinkevicius, and Viktor Medvedev. 2007. The problem of visual analysis of multidimensional medical data. In *Models and Algorithms for Global Optimization*, A. Torn and J. Zilinskas (Eds.). Springer Series in Optimization and Its Applications, Vol. 4. Springer, Boston, 277–298. https://doi.org/10.1007/978-0-387-36721-7_17

[2] Ingwer Borg and Patrick J. F. Groenen. 2005. *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York. https://doi.org/10.1007/978-1-4757-2711-1

[3] Ingwer Borg, Patrick J. F. Groenen, and Patrick Mair. 2018. *Applied Multidimensional Scaling and Unfolding* (2nd ed.). Springer, Cham. https://doi.org/10.1007/978-3-319-73471-2

[4] Jan de Leeuw and Patrick Mair. 2009. Multidimensional scaling using majorization: SMACOF in R. *Journal of Statistical Software* 31, 3 (2009), 1–30. https://doi.org/10.18637/jss.v031.i03

[5] Gintautas Dzemyda and Olga Kurasova. 2006. Heuristic approach for minimizing the projection error in the integrated mapping. *European Journal of Operational Research* 171, 3 (2006), 859–878. https://doi.org/10.1016/j.ejor.2004.09.011

[6] Gintautas Dzemyda, Olga Kurasova, Viktor Medvedev, and Giedre Dzemydaite. 2019. Visualization of data: methods, software, and applications. In *Advances in Mathematical Methods and High Performance Computing*, V.K. Singh, D. Gao, and A. Fischer (Eds.). Advances in Mechanics and Mathematics, Vol. 41. Springer, Cham, 295–307. https://doi.org/10.1007/978-3-030-02487-1_18

[7] Gintautas Dzemyda, Olga Kurasova, and Julius Zilinskas. 2013. *Multidimensional Data Visualization: Methods and Applications*. Optimization and its Applications, Vol. 75. Springer, Cham. https://doi.org/10.1007/978-1-4419-0236-8

[8] Gintautas Dzemyda and Martynas Sabaliauskas. 2020. A novel geometric approach to the problem of multidimensional scaling. In *Numerical Computations: Theory and Algorithms, NUMTA 2019*, Y. D. Sergeyev and D. E. Kvasov (Eds.). Lecture Notes in Computer Science, Vol. 11974. Springer, Cham, 354–361. https://doi.org/10.1007/978-3-030-40616-5_30

[9] Gintautas Dzemyda and Martynas Sabaliauskas. 2021. Geometric multidimensional scaling: A new approach for data dimensionality reduction. *Appl. Math. Comput.* 409, 125561 (2021), 1–14. https://doi.org/10.1016/j.amc.2020.125561"

[10] Gintautas Dzemyda and Martynas Sabaliauskas. 2021. New capabilities of the geometric multidimensional scaling. In *Trends and Applications in Information Systems and Technologies. WorldCIST 2021*, A. Rocha et al. (Ed.). Advances in Intelligent Systems and Computing, Vol. 1366. Springer, Cham, 264–273. https://doi.org/10.1007/978-3-030-72651-5_26

[11] Eurostat-European Commission and others. 2011. *Regions in the European Union. Nomenclature of territorial units for statistics*. Technical Report. NUTS 2010/EU-27. Luxemburg: Publications Office of the European Union.

[12] Data for Visual Analysis of Regional Economic Development. 2017. MIDAS: The National Open Access Research Data Archive. http://dx.doi.org/10.18279/MIDAS.RegionalData.xlsx.30026.

[13] Chun houh Chen, Wolfgang K. Härdle, and Antony Unwin. 2008. *Handbook of Data Visualization*. Springer, Berlin. https://doi.org/10.1007/978-3-540-33037-0

[14] Ian T. Jolliffe. 1986. *Principal Component Analysis*. Springer, Berlin. https://doi.org/10.1007/978-1-4757-1904-8

[15] Rasa Karbauskaite and Gintautas Dzemyda. 2015. Optimization of the maximum likelihood estimator for determining the intrinsic dimensionality of high–dimensionaldata. *International Journal of Applied Mathematics and Computer Science* 25, 4 (2015), 895–913. https://doi.org/10.1515/amcs-2015-0064

[16] Rasa Karbauskaite and Gintautas Dzemyda. 2016. Fractal-based methods as a technique for estimating the intrinsic dimensionality of high-dimensional data: A survey. *Informatica* 27, 2 (2016), 257–281. https://doi.org/10.15388/Informatica.2016.84

[17] Teuvo Kohonen. 2002. Overture. In *Self-Organizing Neural Networks: Recent Advances and Applications*. Springer, Berlin, 1–12. https://doi.org/10.1007/978-3-7908-1810-9

[18] Joseph B. Kruskal. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29, 1 (1964), 1–27. https://doi.org/10.1007/BF02289565

[19] Viktor Medvedev and Gintautas Dzemyda. 2006. Optimization of the local search in the training for SAMANN neural network. *Journal of Global Optimization* 35, 4 (2006), 607–623. https://doi.org/10.1007/s10898-005-5368-1

[20] Viktor Medvedev, Gintautas Dzemyda, Olga Kurasova, and Virginijus Marcinkevicius. 2011. Efficient data projection for visual analysis of large data sets using neural networks. *Informatica* 22, 4 (2011), 507–520. https://doi.org/10.15388/Informatica.2011.339

[21] Julija Pragarauskaite and Gintautas Dzemyda. 2012. Visual decisions in the analysis of customers online shopping behavior. *Nonlinear Analysis: Modelling and Control* 17, 3 (2012), 355–368. https://doi.org/10.15388/NA.17.3.14061

[22] R. 2021. Symmetric Smacof. https://www.rdocumentation.org/packages/smacof/versions/2.0-0/topics/smacofSym.

[23] Martynas Sabaliauskas and Gintautas Dzemyda. 2020. Visual analysis of multidimensional scaling Using GeoGebra. In *Intelligent Methods in Computing, Communications and Control. ICCCC 2020*, I. Dzitac, S. Dzitac, F. Filip, J. Kacprzyk, MJ. Manolescu, and H. Oros (Eds.). Advances in Intelligent Systems and Computing, Vol. 1243. Springer, Cham, 179–187. https://doi.org/10.1007/978-3-030-53651-0_15

[24] Julius Venskus, Povilas Treigys, Jolita Bernataviciene, Viktor Medvedev, Miroslav Voznak, Mindaugas Kurmis, and Violeta Bulbenkiene. 2017. Integration of a self-organizing map and a virtual pheromonefor real-time abnormal movement detection in marine traffic. *Informatica* 28, 2 (2017), 359–374. https://doi.org/10.15388/Informatica.2017.133