On the Parallelization of the Geometric Multidimensional Scaling

Gintautas Dzemyda, Viktor Medvedev, Martynas Sabaliauskas

Vilnius University, Institute of Data Science and Digital Technologies

Introduction

The procedure for mapping data from a high-dimensional space to a lower-dimensional space is **multidimensional scaling (MDS)**. Although MDS demonstrates great versatility, it is computationally expensive, especially when the data set is not fixed and its size is constantly growing. Traditional MDS approaches are limited when analysing very large datasets, as they require long computation times and large amounts of memory.

A way **to minimize MDS stress** has been developed using the ideas of **Geometric MDS** [1–4], where all points in a low-dimensional space change their coordinates simultaneously and independently during a single iteration of stress minimization.

We examine how the computational time expenses for data dimensionality reduction and multidimensional data visualization varies depending on the number of used CPU threads.

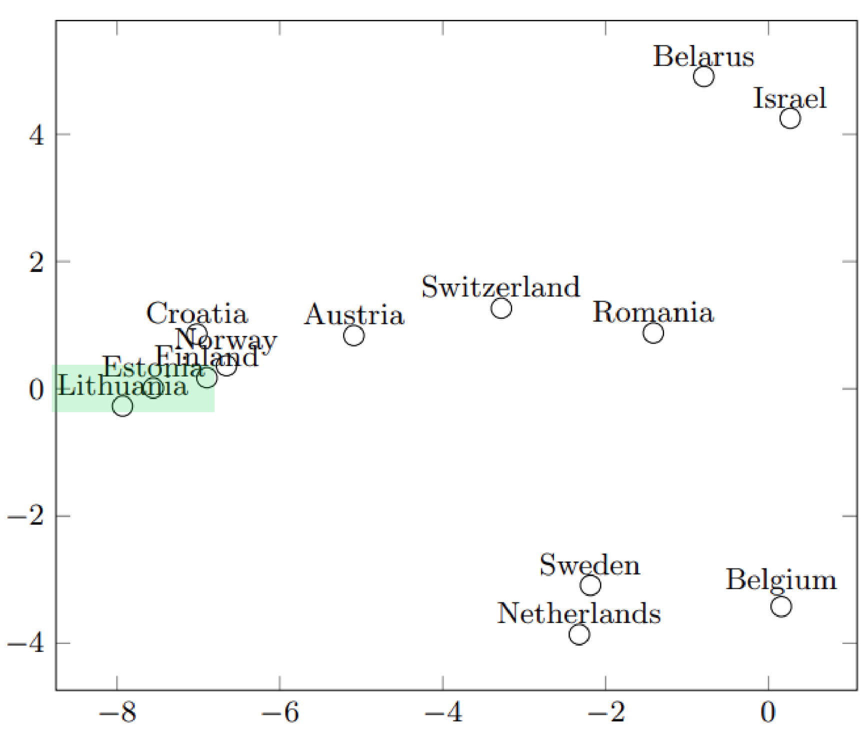

Figure 1. Example of dimensionality reduction results obtained using MDS: daily cumulative relative **COVID-19 data** [5] for selected countries, $n = 106$

$X=\{X_i=(x_{i1},\dots,x_{in}),\ i=1,\dots,m\}$, $X_i \in R^n$, $n \geq 3$ — multidimensional data set. **Dimensionality reduction means** finding the set of coordinates 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 is represented by $Y_i \in R^d$, $d \leq 3$.

Parallel Implementation of Geometric MDS

Algorithm 1: Algorithm for simultaneous calculation (parallel processing) of new points Y_j^* in the lower dimensional space

```
Data: n is the dimensionality of data, m is the number of data points;
D = (d_ij) is a symmetric m x m matrix of proximities between n-dimensional
data points;
Y = Y_1, ..., Y_m is m x d matrix of the projected points;
j^0 is an index of the point among Y_1, ..., Y_m, which coordinates will be
optimized;
acc is the accuracy of the global stress.
Result: Y* is m × d matrix of the projected points; S_1 - global stress
Y* ← starting values of m × d matrix;
S_1 ← S(Y*_1, ..., Y*_m);
while TRUE do
    foreach i ∈ {1, ..., m} in parallel do                          ⎫
        foreach i ∈ {1, ..., m} \ {j^0} do                          ⎪
            w ← d_ij0 / sqrt( Σ_{k=1}^{d} (y_ik − y_j0k)^2 );       ⎪
            for k ← 1 to d do                                       ⎪
                A_ik ← y_jk + (y_j0k − y_ik) w;                     ⎬ Working Parallel
            end                                                     ⎪
        end                                                         ⎪
        for k ← 1 to d do                                           ⎪
            y*_j0k ← 1/(m−1) Σ_{j=1, j≠j0}^{m} A_jk;                ⎪
        end                                                         ⎪
        Y_j^0 ← Y*_j0                                               ⎭
        /* Y*_j0 are final coordinates of the optimized point */
    end
    for i ← 1 to m do
        d_ij ← 0;
        for j ← i + 1 to m do
            d*_ii ← sqrt( Σ_{k=1}^{d} (y*_ik − y*_jk) );
            d*_ji ← d*_ij;
        end
    end
    S_0 ← S_1; S_1 ← S(Y*_1, ..., Y*_m);
    if S_0 − S_1 < acc then
        break
    else
end
```

The value of the global stress function $S(\cdot)$ will decrease when all the points $Y_1,\dots,Y_m$ change their coordinates to $Y_1^*,\dots,Y_m^*$ once at the same time (all $Y_1^*,\dots,Y_m^*$ are computed using $Y_1,\dots,Y_m$, exclusively):

$$S(Y_1,\dots,Y_m) > S(Y_1^*,\dots,Y_m^*).$$

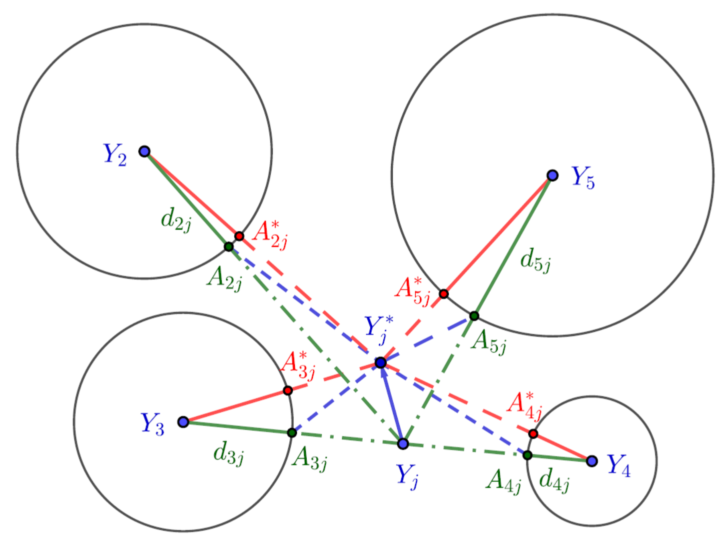

Figure 2. An example of a single iteration of the Geometric MDS method.

The California Housing dataset: 20,640 observations on 9 variables (8 numeric, predictive attributes, the target). For preliminarily experiments, only part of the data was used, i.e. data sizes of 500, 1000, 1500, 2000, 2500 and 3000 observations were analysed during the experiments.

Results

Table 1: The average dependency of the computation time of new coordinates (not including the complete visualisation process) on the size of the data, when using different numbers of CPU threads for parallel processing, n= 8.

	number of CPU threads (parallel processing)					
data size, m	1	2	4	6	8	10
100	0.0326	0.7382	0.8328	1.0078	1.0878	1.2058
500	0.9560	1.1408	1.1723	1.1864	1.4236	1.4195
1000	3.6718	2.6891	2.0750	2.0400	2.0721	2.2574
1500	8.6668	5.4726	4.1810	3.6891	3.5598	3.8951
2000	16.0643	9.4951	6.5537	5.9791	5.6784	6.0171
2500	25.0601	15.7463	10.4054	9.7634	9.1108	9.9568
3000	37.3540	24.3042	15.9319	14.5363	13.7851	14.3094

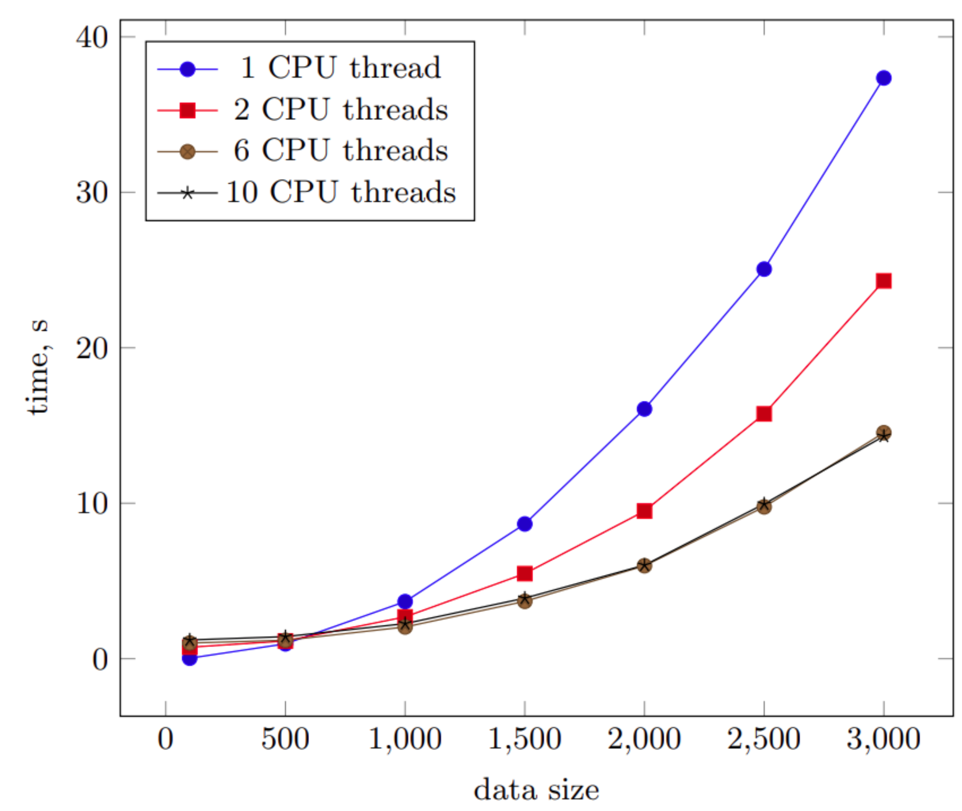

Figure 3. Average dependence of the computation time of new coordinates (not taking into account the complete visualisation process) on the size of the data, using different numbers of CPU threads for parallel processing, n = 8

Conclusions

A way to minimize MDS stress has been developed using the ideas of Geometric MDS, where all points in a low-dimensional space change their coordinates simultaneously and independently during a single iteration of stress minimization.

Geometric MDS allows to implement parallel computing using multithreaded multicore processors, as a result of which the coordinate calculation time in low-dimensional space can be reduced on average by 2.5 times. This allows to conclude that the proposed algorithm can be used to reduce the dimensionality of large-scale data more efficient and faster than the classical MDS method.

References

1. G. Dzemyda and Sabaliauskas M. A novel geometric approach to the problem of multidimensional scaling. *Numerical Computations: Theory and Algorithms*, NUMTA 2019, vol. 11974 LNCS, 354–361. Springer, 2020.
2. M. Sabaliauskas and G. Dzemyda. Visual analysis of multidimensional scaling using GeoGebra. *In Intelligent Methods in Computing, Communications and Control*. ICCCC 2020, vol. 1243 of Advances in Intelligent Systems and Computing, pages 179–187. Springer, 2020.
3. G. Dzemyda and M. Sabaliauskas. New capabilities of the geometric multidimensional scaling. In *World Conference on Information Systems and Technologies*, pages 264–273. Springer, 2021.
4. G. Dzemyda and M. Sabaliauskas. Geometric multidimensional scaling: A new approach for data dimensionality reduction. *Applied Mathematics and Computation*, 409:125561, 2021.
5. J. Markeviciute, J. Bernataviciene, R. Levuliene, V. Medvedev, P. Treigys, J. Venskus. Attention-Based and Time Series Models for Short-Term Forecasting of COVID-19 Spread. *CMC-COMPUTERS MATERIALS & CONTINUA*,70(1), 2022.

Acknowledgement

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