Majorizing Kruskals Stress Formula One

Jan de Leeuw

January 18, 2025

The majorization algorithm to minimize raw stress is extended to minimization of normalized stress (stress formula one).

Table of contents

Note: This is a working manuscript which will be expanded/updated frequently. All suggestions for improvement are welcome. All Rmd, tex, html, pdf, R, and C files are in the public domain. Attribution will be appreciated, but is not required. The files can be found at https://github.com/deleeuw in the repositories smacofCode, smacofManual, and smacofExamples.

1 Introduction

The majorization algorithm to minimize Kruskal's raw stress (Kruskal (1964a), Kruskal (1964b)) defined as

$$\sigma(X) := \sum_{1\leq i<j\leq n}\sum w_{ij}(\delta_{ij}-d_{ij}(X))^2 \tag{1}$$

is by now well-known. It has been the default method to minimize Equation 1 for almost 50 years, and its convergence properties have been studied in considerable detail. We refer to De Leeuw and Heiser (1980) and De Leeuw (1988) for theory, and to De Leeuw and Mair (2009) and Mair, Groenen, and De Leeuw (2022) for implementations.

In non-metric MDS we have to normnalize stress in order to prevent trivial zero solutions. The two common normalizations are stress formula one

$$\sigma_1(X) := \frac{\sum\sum_{1\leq i<j\leq n} w_{ij}(\delta_{ij}-d_{ij}(X))^2}{\sum\sum_{1\leq i<j\leq n} w_{ij}d_{ij}^2(X)}, \tag{2}$$

and stress formula two

$$\sigma_2(X) := \frac{\sum\sum_{1\leq i<j\leq n} w_{ij}(\delta_{ij}-d_{ij}(X))^2}{\sum\sum_{1\leq i<j\leq n} w_{ij}(d_{ij}^2-\overline{d}(X))^2}, \tag{3}$$

with $\overline{d}$ the weighted mean of the d_{ij}.

Majorizing normalized stress is more complicated than minimizing raw stress. For stress formula two a majorization algorithm was proposed by De Leeuw (n.d.), using a combination of the usual stress majorization based on Cauchy-Schwartz and the fractional programming method of Dinkelbach (1967). In this short paper we apply the same ideas to stress formula one.

2 Majorization

Define

$$\omega(X,Y) := \sigma(X) - \sigma_1(Y)\eta^2(X),$$

{#eq:omegadef} where σ is defined in Equation 1, σ_1 is defined in Equation 2, and

$$\eta^2(X) := \sum_{1\leq i<j\leq n}\sum w_{ij}d_{ij}^2(X). \tag{4}$$

Note that $\sigma_1(X) = \sigma(X)/\eta^2(X)$. It is clear that

- for all non-zero X, Y we have $\omega(X,Y) < 0$ if and only if $\sigma_1(X) < \sigma_1(Y)$,
- for all non-zero X, Y we have $\omega(X,Y) = 0$ if and only if $\sigma_1(X) = \sigma_1(Y)$.

This is sometimes called *Dinkelbach majorization*, because the basic step of converting a ratio of two functions to a weighted difference of the two is due to Dinkelbach (1967).

We now use standard MDS notation and define

$$\phi(X,Y) := 1 - 2\text{ tr } X'B(Y)Y + (1-\sigma_1(Y))\text{tr } X'VX, \tag{5}$$

with

$$B(Y) := \sum_{1\leq i<j\leq n}\sum w_{ij}\frac{\delta_{ij}}{d_{ij}(Y)}A_{ij} \tag{6}$$

and

$$V := \sum_{1\leq i<j\leq n}\sum w_{ij}A_{ij} \tag{7}$$

and

$$A_{ij} := (e_i - e_j)(e_i - e_j)', \tag{8}$$

i.e. A_{ij} is symmetric, doubly centered, positive semi-definite, of rank one, with elements (i,i) and (j,j) equal to $+1$, elements (i,j) and (j,i) equal to -1, and all other elements zero. Note that $\eta^2(X) = \text{tr } X'VX$. In Equation 5 we have assumed, without loss of generality, that the weighted sum of squares of the the δ_{ij} is one.

Now, by standard MDS majorization,

- $\omega(X,Y) \leq \phi(X,Y)$, and
- $\omega(X,X) = \phi(X,X) = 0$.

Suppose $\sigma_1(Y) < 1$. Define Y^+ as the minimizer of η over X for fixed Y and suppose $Y^+ \neq Y$. Now

$$\omega(Y^+, Y) \leq \phi(Y^+, Y) < \phi(Y, Y) = 0, \tag{9}$$

and thus $\sigma_1(Y^+) < \sigma_1(Y)$. It follows that the iterative algorithm

$$X^{(k+1)} = \frac{1}{1 - \sigma_1(X^{(k)})} V^+ B(X^{(k)}) X^{(k)} \tag{10}$$

is convergent.

We assumed $\sigma_1(X^{(k)}) < 1$ for all k. As in De Leeuw (n.d.), this causes no loss of generality. Start the non-metric MDS iterations with any $X^{(0)}$ and compute the optimal δ_{ij} by projecting $d(X^{(0)})$ on the code of admissible (usually monotone) transformations. This guarantees $\sigma_1(X^{(0)}) < 1$, and because the algorithm generates a decreasing sequence of stress values $\sigma_1(X^{(k)}) < 1$ for all k.

We also assumed that $X^{(k+1)} \neq X^{(k)}$. This again causes no loss of generality because we can simply stop the iterations if $X^{(k+1)} = X^{(k)}$, i.e. if we have reached a fixed point. At a fixed point the gradient of σ_1 is zero.

3 Unnormalized iterations

The Guttman transform $\Gamma(X) := V^+ B(X) X$ is homogeneous of degree zero. Thus if $\tilde{X}^{(k)} = \Gamma^k(X^0)$ is the sequence of unnormalized Guttman transforms then

$$X^{(k+1)} = \frac{1}{1 - \sigma_1(\tilde{X}^{(k)})} \tilde{X}^{(k)}, \tag{11}$$

and the algorithm in Equation 10 generates the same sequence, up to proportionality, as the unnormalized Guttman transforms. This remains true if Guttman transforms are alternated with optimal scaling steps that minimize stress over the δ_{ij} for given X.

References

De Leeuw, J. 1988. "Convergence of the Majorization Method for Multidimensional Scaling." *Journal of Classification* 5: 163–80.

———. n.d. "Majorizing Kruskals Stress Formula Two." https://jansweb.netlify.app/publication/deleeuw-e-24-a/deleeuw-e-24-a.pdf.

De Leeuw, J., and W. J. Heiser. 1980. "Multidimensional Scaling with Restrictions on the Configuration." In *Multivariate Analysis, Volume V*, edited by P. R. Krishnaiah, 501–22. Amsterdam, The Netherlands: North Holland Publishing Company.

De Leeuw, J., and P. Mair. 2009. "Multidimensional Scaling Using Majorization: SMACOF in R." *Journal of Statistical Software* 31 (3): 1–30. https://www.jstatsoft.org/article/view/v031i03.

Dinkelbach, W. 1967. "On Nonlinear Fractional Programming." *Management Science* 13: 492–98.

Kruskal, J. B. 1964a. "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis." *Psychometrika* 29: 1–27.

———. 1964b. "Nonmetric Multidimensional Scaling: a Numerical Method." *Psychometrika* 29: 115–29.

Mair, P., P. J. F. Groenen, and J. De Leeuw. 2022. "More on Multidimensional Scaling in R: smacof Version 2." *Journal of Statistical Software* 102 (10): 1–47. https://www.jstatsoft.org/article/view/v102i10.