# On Trivial Solutions in Nonmetric Scaling

Jan de Leeuw

December 29, 2024

TBD

## Table of contents

# 1 Introduction

Sixty years ago Joseph B. Kruskal introduced the basic framework for constructing nonmetric scaling techniques (Kruskal (1964a), Kruskal (1964b)). The basic ingredients are

- the **parameters** $\mathbb{X}$, a subset of $\mathbb{R}^m$,
- the **model** , a function $f : \mathbb{X} \Rightarrow \mathbb{R}^\nu$, and
- the **data**, a partial order $\preceq$ on $I_n := \{1, 2, \cdots, \nu\}$.

From these basic ingredients we construct some additional entities.

- The **quantifications**, an isotone cone $K$ in $\mathbb{R}^\nu$ with $x \in K$ if $x_i \leq x_j$ whenever $i \preceq j$, and
- the **raw least squares loss function** $\sigma_K : \mathbb{X} \Rightarrow \mathbb{R}_+$, defined as

$$\eta_K(x) := \frac{1}{2}\|f(x) - P_K(f(x))\|^2, \tag{1}$$

- the **monotone regression** $P_K : \mathbb{R}^n \Rightarrow K$, defined for all $z \in \mathbb{R}^\nu$ as

$$P_K(z) = \underset{y \in K}{\text{argmin}}\, \|z - y\|^2. \tag{2}$$

The raw nonmetric scaling (NMS) problem for model $f$ and data $\preceq$ is the minimization of $\eta_K$ over $x$.

Kruskal initially applied this framework to multidimensional scaling (MDS) with symmetric one-mode data.

- The data are partial order over the **dissimilarities** between $n$ objects,
- the parameters are **configurations**, sets of $n$ points in $p$ **dimensions**, i.e. in $\mathbb{R}^p$, and
- the model gives the **distances** between the $n$ points, or a subset of them, in the configuration.

Kruskal realized, of course, that if the parameters $\mathbb{X}$ are a cone, i.e. if $\lambda x \in \mathbb{X}$ whenever $x \in \mathbb{X}$ and $\lambda \geq 0$, then the solution of the raw NMS problem is the trivial solution $x = 0$, i.e. all $n$ points of the configuration in the origin. The reason is that distances are positively homogeneous and, as it turns out, most of the models of interest in NMS are positively homogeneous of some order, i.e. for all $x$ and $\lambda \geq 0$ we have $f(\lambda x) = \lambda^r f(x)$ for some $r$. For all those models the raw NMS has the solution $x = 0$, which is trivial in the sense that it does not depend on the data at all.

As a consequence Kruskal introduced normalized stress, which is

$$\sigma_{KL}(x) := \frac{\eta_K(x)}{\eta_L(x)}.$$

where $L$ is another cone of quantifications and $\eta_L$ is defined in the same way as $\eta_K$, as the squared distance between $f(x)$ and $L$. Now $x = 0$ will lead to an undefined $\sigma_{KL}$, in which both numerator and denominator are zero. If $L$ is the cone consisting only of the origin then $\sigma_{KL}$ is stress-1, or **stress formula one**. If $L$ is the cone of all vectors proportional to $e$, the vector with all elements equal to one, then $\sigma_{KL}$ is stress-2, or **stress formula two**. Thus, for homogeneous models, the NMS problem becomes minimization of one of these $\sigma_{KL}$ over $\mathbb{X}$.

Soon after Kruskal's original papers Roskam (1968) and Kruskal and Carroll (1969) applied the same NMS framework to multidimensional unfolding (MDU), i.e. to conditional two-mode data. Instead of a single partial order over all $\binom{n}{2}$ dissimilarities there are $n$ separate partial orders over $m$ objects. Now $n\binom{m}{2}$ is very much smaller than $\binom{n+m}{2}$, especially if $n$ is large and $m$ is small, and consequently MDU data have only a small fraction of the ordinal information of complete MDS data. As a consequence the straightforward application of the Kruskal framework to MDU failed because the algorithm was able to find non-zero trivial solutions with perfect stress that were independent of the data. It was necessary to adapt Kruskal's framework to the low-information MDU situation. For a historical reviews of the various adaptations since 1964 we refer to De Leeuw (1983), Van Deun et al. (2005) and to the excellent dissertations of Van Deun (2005) and Frank M. T. A. Busing (2010).

In this paper we are interested in the initial modification of the NMS framework proposed by Roskam (1968) and Kruskal and Carroll (1969). The loss function is partitioned into separate normalized loss functions for each row. In this partitioning both $\eta_K$ and $\eta_L$ are zero for each row at a trivial solution, and thus stress-2 is undefined for each row. Define stress-3, or **stress formula three**, as

$$\sigma_{PQ}(x) = \sum_{i=1}^{n} \frac{\|f_i(x) - P_i(f_i(x))\|^2}{\|f_i(x) - Q_i(f_i(x))\|^2},$$

where $P_i$ and $Q_i$ are the monotone regressions that project on cones $K_i$ and $L_i$. There is now a separate model $f_i : \mathbb{X} \Rightarrow \mathbb{R}^{m_i}$ for each $i$.

It was shown by De Leeuw (1983) that this modification of the basic framework is still problematic, because the partitioned loss function is behaving quite normally near trivial points and can have solutions with arbitrarily small loss that are indistinguishable from trivial solutions. De Leeuw (1983) is somewhat tentative and not very explicit about both the results and the proofs. We want to make our treatment of the trivial solution problem both more explicit and more general. Although MDU motivated study of the problem, we build up the theory for more general models and their loss functions.

# 2 General Results

The monotone regression has two properties which are crucial for the results in this paper.

1. If $e$ has all elements equal to one and $\lambda \geq 0$ then $P_K(x + \lambda e) = P_K(x) + \lambda e$ (shift invariance).
2. If $\lambda \geq 0$ then $P_K(\lambda x) = \lambda P_K(x)$ (homogeneity). This is actually true for all cone projections.

In this paper we deal with a specific class of models which are differentiable and nallow for trivial solutions.

1. There is an $x_0 \in \mathbb{R}^m$ such that $f(x_0) = \lambda e$, with $\lambda \geq 0$ and with $e \in \mathbb{R}^n$ the vector with all elements equal to one.
2. The model $f$ is differentiable at $x_0$, and the Jacobian at $x_0$ is $J$.

We give five examples of such models.

1. A linear model.
2. Principal Component Analysis (PCA) of a symmetric matrix.
3. A small MDS four-point example in two dimensions.
4. MDS of $n$ points in $n - 1$ dimensions, with $x_0$ a regular simplex.
5. MDU with $x_0$ one of the trivial solutions.

It is clear that $\eta_K$ and $\eta_L$ are both equal to zero at $x_0$, and thus $\sigma_{KL}$ is undefined at $x_0$. We are interested in the behavior of $\sigma_{KL}$ near $x_0$.

## 2.1 Raw Loss

**Theorem 2.1.** *For all $\delta$*

$$\eta_K(x_0+\epsilon\delta)=\frac{1}{2}\epsilon^2\|J\delta-P_K(J\delta)\|^2+o(\epsilon^2). \tag{3}$$

*Proof.*

$$\begin{aligned}P_K(f(x_0+\epsilon\delta))&=P_K(f(x_0)+\epsilon J\delta+o(\epsilon))=\\&=P_K(\lambda e+\epsilon J\delta+o(\epsilon))=\\&=\lambda e+P_K(\epsilon J\delta+o(\epsilon))=\\&=\lambda e+\epsilon P_K(J\delta+o(1))=\\&=\lambda e+\epsilon P_K(J\delta)+o(1).\end{aligned}$$

which uses the shift-invariance and homogeneity of the monotone regression mentioned in the introduction, as well as its continuity. In addition

$$f(x_0+\epsilon\delta)=f(x_0)+\epsilon J\delta+o(\epsilon)=\lambda e+\epsilon(J\delta+o(1)).$$

Thus

$$f(x_0+\epsilon\delta))-P_K(f(x_0+\epsilon\delta))=\epsilon(J\delta-P_K(J\delta))+o(1))=\epsilon(J\delta-P_K(J\delta))+o(\epsilon),$$

and Equation 3 follows. □

Note that Theorem 2.1 applies to any partial order, or more generally to any cone with $P_K(x+\lambda e)=P_K(x)+\lambda e$ and $P_K(\lambda x)=\lambda P_K(x)$ for $\lambda\geq 0$. Theorem 2.1 also remains unchanged if we use a positive definite $W$ to define a weighted least squares norm.

The result in Theorem 2.1 can also be expressed in terms of (one-sided) directional derivatives. We have

$$dg_K(x_0;\delta):=\lim_{\epsilon\downarrow 0}\frac{g_K(x_0+\epsilon\delta)-g_k(x_0)}{\epsilon}=0,$$

and

$$d^2g_K(x_0;\delta):=\lim_{\epsilon\downarrow 0}\frac{g_K(x_0+\epsilon\delta)-g_k(x_0)-dg_K(x_0;\delta)}{\frac{1}{2}\epsilon^2}=\|J\delta-P_K(J\delta)\|^2.$$

## 2.2 Normalized Loss

**Theorem 2.2.** *For all $\delta$ with $J\delta \notin L$*

$$\lim_{\epsilon \downarrow 0} \sigma_{KL}(x_0 + \epsilon\delta) = \omega_{KL}(\delta), \tag{4}$$

*with*

$$\omega_{KL}(\delta) := \frac{\|J\delta - P_K(J\delta)\|^2}{\|J\delta - P_L(J\delta)\|^2}. \tag{5}$$

*Proof.* This is immediate from Theorem 2.1. □

Since $h(x_0)$ is undefined the result of Theorem 2.2 cannot be interpreted in terms of directional derivatives. Equation 4 can be understood as a variant of l'Hôpital's rule, in which both the function values and the first (directional) derivatives in the numerator and denominator vanish.

## 2.3 Derivatives

After approximating the function values of the non-metric loss function near a trivial solution, we can also approximate the derivatives.

**Theorem 2.3.**

$$\lim_{\epsilon\downarrow 0}\epsilon\mathcal{D}h(x_0+\epsilon\delta)=\mathcal{D}\omega(\delta).$$

*Proof.* Start with the standard formula for the derivative of a quotient.

$$\mathcal{D}h(x)=\frac{\mathcal{D}g_K(x)-h(x)\mathcal{D}g_L(x)}{g_L(x)},$$

and thus

$$\mathcal{D}h(x+\epsilon\delta)=\frac{\mathcal{D}g_K(x+\epsilon\delta)-h(x+\epsilon\delta)\mathcal{D}g_L(x+\epsilon\delta)}{g_L(x+\epsilon\delta)}. \tag{6}$$

Now both $g_k$ and $g_L$ are differentiable, with

$$\mathcal{D}g_K(x)=(f(x)-P_K(f(x))'\mathcal{D}f(x) \tag{7}$$

and with a corresponding result for $\mathcal{D}g_L(x)$. Thus, using Equation 7,

$$\mathcal{D}g_K(x_0+\epsilon\delta)=\epsilon(J\delta-P_K(J\delta))'J+o(\epsilon). \tag{8}$$

In the same way

$$\mathcal{D}g_L(x_0+\epsilon\delta)=\epsilon(J\delta-P_L(J\delta))'J+o(\epsilon). \tag{9}$$

Now substitute Equation 8, Equation 9, and Equation 4 in Equation 6 and gather terms. This gives eventually

$$\lim_{\epsilon\downarrow 0}\epsilon\mathcal{D}h(x_0+\epsilon\delta)=\frac{\{(J\delta-P_K(J\delta))-\omega(\delta)(J\delta-P_L(J\delta))\}'J}{\frac{1}{2}\|J\delta-P_L(J\delta)\|^2}, \tag{10}$$

and the right hand side of Equation 10 is the derivative of $\omega$ at $\delta$. □

Note this result is somewhat non-standard, because of the $\epsilon$ on the left hand side. We do not have convergence of the derivative of $h$ at $x_0+\epsilon\delta$, in fact the derivative diverges if $\epsilon\downarrow 0$. But we have convergence of $\epsilon$ times the derivative.

# 3 Examples

## 3.1 Linear Model

Suppose $A$ is an $n \times m$ matrix and there is a $x-$ _0 such that $Ax_0 = e$. For instance the first column of $A$ could be an intercept, and $x_0 = e_1$, or the rows of $A$ could add up to one and $x_0 = e$, as in compositional or confusion data.

We have

$$P_K(A(x_0 + \epsilon\delta)) = e + \epsilon P_K(A\delta)$$

without any approximations or residuals. Thus

$$\sigma_S(x_0 + \epsilon\delta) = \omega_S(\delta) = \frac{\|A\delta - P_K(A\delta)\|^2}{\|A\delta - P_L(A\delta)\|^2}.$$

Again this is an exact, although not very interesting, relationship. At a trivial solution the linear loss function for $Ax$ is replaced by the linear loss function for $A\delta$.

## 3.2 PCA

Suppose $C$ is a square symmetric matrix that we want to approximate by the outer product $XX'$. Take $C_0 = ee'$. Now $(e+\epsilon\delta)(e+\epsilon\delta)' = ee' + \epsilon(e\delta' + \delta e') + o(\epsilon)$ and thus $J\delta = e\delta' + \delta e'$, or

$$\{J\delta\}_{ij} = \delta_i + \delta_j + o(e).$$

More generally $c_{ij} = \text{tr } X'E_{ij}X$ with $E_{ij} = \frac{1}{2}(e_ie_j' + e_je_i')$. Thus $J_{ij,ks} = \mathcal{D}c_{ij}(X) = \{E_{ij}X\}_{ks}$ and $J\delta$ is

$$\frac{1}{2}\text{tr } X'E_{ij}\Delta = \frac{1}{2}\{x_i'\delta_j + x_j'\delta_i\}$$

Take $C_0 = ee'$. Now $(e+\epsilon\delta)(e+\epsilon\delta)' = ee' + \epsilon(e\delta' + \delta e') + o(\epsilon)$ and thus $J\delta = e\delta' + \delta e'$, or

$$\{J\delta\}_{ij} = \delta_i + \delta_j + o(e).$$

## 3.3 MDS

$$J(X_0)\Delta = \frac{1}{d_{ij}(X_0)}\text{tr } X_0'A_{ij}\Delta$$

Our next example is the local behavior of Kruskal's stress-formula-two near a regular simplex in $n$ dimensions. In a regular simplex all interpoint distances are equal. This example does not have much practical relevance, because full-dimensional regular simplexes only happen in artificial MDS examples. But it does serve as the basis for the MDU results in the next section.

Define $d(X)$ as the vector of the $\frac{1}{2}n(n-1)$ distances $d_{ij}(X)$ with $i < j$. Matrix $X_0$ has the coordinates of the unit simplex, i.e. it is scalar square matrix which is $\frac{1}{2}\sqrt{2}$ times the identity, for which $d(X_0) = e$.

Stress-formula-two is

$$\sigma_S(X) := \frac{\sigma_K(X)}{\sigma_L(X)}, \tag{11}$$

where

$$\sigma_K(X) := \frac{1}{2}\|d(X) - P_K(d(X))\|^2, \tag{12}$$

and

$$\sigma_L(X) = \frac{1}{2}\|d(X) - P_L(d(X))\|^2. \tag{13}$$

In Equation 12, which defines *raw stress*, we project on the monotone regression cone. In Equation 13 $P_L$ projects on the cone of vectors proportional to $e$, i.e. $P_L$ takes the average of the elements of $d(X)$. Cone $L$ is smaller than cone $K$, and thus $0 \leq \sigma_S(X) \leq 1$. Note that both $\sigma_R(X_0)$ and $\sigma_L(X_0)$ are zero, so $\sigma_S(X_0)$ is undefined.

## 3.4 Loss Function

We use Theorem 2.2 to approximate stress near the regular simplex. I apologize for using superscripted delta for Kronecker's symbol and subscripted delta for the perturbation.

**Theorem 3.1.** *For all* $\Delta$

$$\sigma_S(X_0 + \epsilon\Delta) = \omega_S(\Delta) + o(\epsilon^2), \tag{14}$$

*where*

$$\omega_s(\Delta) := \frac{\|\xi(\Delta) - P_K(\xi(\Delta))\|^2}{\|\xi(\Delta) - P_L(\xi(\Delta))\|^2}, \tag{15}$$

*and*

$$\xi_{ij}(\Delta) := \delta_{ii} + \delta_{jj} - \delta_{ij} - \delta_{ji}. \tag{16}$$

*Proof.* In view of Theorem 2.2 we need an expression for $J\delta$. We first find $J$, using double indexing.

$$\{J\}_{ij,kl} = \mathcal{D}_{kl}d_{ij}(X_0) = (x^0_{il} - x^0_{jl})(\delta^{ik} - \delta^{jk}) = \frac{1}{2}\sqrt{2}(\delta^{il} - \delta^{jl})(\delta^{ik} - \delta^{jk}).$$

Thus $J\delta$ is

$$\{J\delta\}_{ij} = \frac{1}{2}\sqrt{2}\sum_{k=1}^{n}\sum_{l=1}^{p}(\delta^{il} - \delta^{jl})(\delta^{ik} - \delta^{jk})\delta_{kl} = \frac{1}{2}\sqrt{2}(\delta_{ii} + \delta_{jj} - \delta_{ij} - \delta_{ji}).$$

□

Equation 15 defines a Kruskal-type non-metric loss function for the linear model defined by Equation 16. And, of course, $0 \leq \omega(\Delta) \leq 1$. Thus for small epsilon we have a configuration which cannot be distinguished from a regular simplex, but for which stress-formula-two is very close to $\omega(\Delta)$. And if we have a $\Delta$ for which $\omega(\Delta)$ is close to zero, then stress-formula-two will be close to zero too, although the configuration $X_0 + \epsilon\Delta$ is very much like a regular simplex.

There is an interesting case of Equation 14 which deserves to be mentioned separately.

**Corollary 3.1.** *If $\Delta = ZZ'$ for some $n \times p$ matrix $Z$, then*

$$\lim_{\epsilon \downarrow 0} \sigma_S(X_0 + \epsilon\Delta) = \frac{\|d^2(Z) - P_K(d^2(Z))\|^2}{\|d^2(Z) - P_L(d^2(Z))\|^2}, \tag{17}$$

*where $d^2(Z)$ has the squared distances $d^2_{ij}(Z)$.*

*Proof.* If $\Delta = ZZ'$ then $\xi_{ij}(\Delta) = d^2_{ij}(Z)$. □

Thus if $\Delta = ZZ'$ the approximation $\omega(\Delta)$ is actually an example of the sstress non-metric scaling formula used in ALSCAL (Takane, Young, and De Leeuw (1977)). Finding a $Z$ with a low value of sstress means finding a configuration that looks like a regular simplex and has a low value of stress-formula-two.

Here is a numerical example of a regular simplex in ten dimensions, actually of course in a nine-dimensional subspace. We choose $\Delta$ using the R statement

```
outer(1:10, 1:10) / 10)
```

where $z$ is the vector with elements 1 to 10. We vary epsilon from $10^{-1}$ to $10^{-8}$. The cone $K$ is the set of non-decreasing vectors. The sigma column gives stress-formula-two and the omega column, which does not vary with epsilon, gives the approximation from Equation 17.

Table 1: Stress-formula-two and Approximation

| epsilon | sigma | omega |
|---:|---:|---:|
| 1e-01 | 0.957591846280264 | 0.962931881332623 |
| 1e-02 | 0.962509060289917 | 0.962931881332623 |
| 1e-03 | 0.962895792316799 | 0.962931881332623 |
| 1e-04 | 0.962928342184946 | 0.962931881332623 |
| 1e-05 | 0.962931528123117 | 0.962931881332623 |
| 1e-06 | 0.962931846016105 | 0.962931881332623 |
| 1e-07 | 0.962931877754530 | 0.962931881332623 |
| 1e-08 | 0.962931880468995 | 0.962931881332623 |

We see that the sigma get closer to omega if $\epsilon$ gets smaller. Numerically we cannot go much smaller than $\epsilon = 10^{-8}$, because rounding errors will take over and the approximation will get worse. In this example the omega is larger than the actual values of stress, which increase with decreasing epsilon. In the next example omega is smaller then the values of stress, which now decrease with decreasing epsilon. In this second example $\Delta$ is given by the R statement

```r
matrix(1:100 %% 9, 10, 10)
```

Table 2: Stress-formula-two and Approximation

| epsilon | sigma | omega |
|---:|---:|---:|
| 1e-01 | 0.877002555781771 | 0.845814132104455 |
| 1e-02 | 0.848177795762601 | 0.845814132104455 |
| 1e-03 | 0.846058360474781 | 0.845814132104455 |
| 1e-04 | 0.845838672180776 | 0.845814132104455 |
| 1e-05 | 0.845816587323383 | 0.845814132104455 |
| 1e-06 | 0.845814377637802 | 0.845814132104455 |
| 1e-07 | 0.845814156611029 | 0.845814132104455 |
| 1e-08 | 0.845814134507446 | 0.845814132104455 |

## 3.5 Derivatives

Intro

**Theorem 3.2.**

$$\epsilon\mathcal{D}\sigma_S(X_0+\epsilon\Delta)=\mathcal{D}\omega_S(X_0+\epsilon\Delta)+o(\epsilon)$$

*Proof.*

$$\sum_i\sum_j r_{ij}\{J\}_{ij,ks}=\frac{1}{2}\sqrt{2}\sum_i\sum_j r_{ij}(\delta^{is}-\delta^{js})(\delta^{ik}-\delta^{jk})=\delta^{ks}\sum_j r_{kj}-r_{ks}=\sum_{i<j}r_{ij}A_{ij}$$

$$\sum_i\sum_j\{J\delta\}_{ij}\{J\}_{ij,ks}=\sum_i\sum_j\{J\}_{ij,ks}\delta_{ks}=\frac{1}{2}\sum_i\sum_j(\delta_{ii}+\delta_{jj}-\delta_{ij}-\delta_{ji})(\delta^{is}-\delta^{js})(\delta^{ik}-\delta^{jk})$$

□

To show how well the derivates of $\omega$ approximate those of $\sigma$ at $X_0+\epsilon\Delta$ we use $n=4$ and $\Delta$ equal to

```
     [,1] [,2] [,3] [,4]
[1,]    1    2    0    1
[2,]    2    0    1    2
[3,]    0    1    2    0
[4,]    1    2    0    1
```

The derivative of $\omega$ at $\Delta$ is

```
     [,1]          [,2]          [,3]          [,4]
[1,]  0.1666666667 -0.0833333333 -0.0833333333  0.0000000000
[2,] -0.0833333333  0.0000000000  0.0000000000  0.0833333333
[3,] -0.0833333333  0.0000000000  0.0000000000  0.0833333333
[4,]  0.0000000000  0.0833333333  0.0833333333 -0.1666666667
```

The next three matrices give $\epsilon\mathcal{D}\sigma(X_0+\epsilon\Delta)$ for $\epsilon$ equal to 1e-2, 1e-4 and 1e-6.

```
     [,1]          [,2]          [,3]          [,4]
[1,]  0.1656849540 -0.0799710700 -0.0865322836  0.0008112829
[2,] -0.0834765853  0.0009825449 -0.0011638146  0.0836881185
[3,] -0.0830335304 -0.0011267899  0.0009806252  0.0831497861
[4,]  0.0008251618  0.0801153150  0.0867154729 -0.1676491876
```

```
             [,1]          [,2]          [,3]          [,4]
[1,]  0.1666568458 -0.0832996158 -0.0833654141  0.0000081834
[2,] -0.0833349681  0.0000098210 -0.0000114596  0.0833366097
[3,] -0.0833300624 -0.0000114559  0.0000098208  0.0833316945
[4,]  0.0000081848  0.0833012507  0.0833670528 -0.1666764876
```

```
             [,1]          [,2]          [,3]          [,4]
[1,]  0.1666665684 -0.0833329962 -0.0833336541  0.0000000819
[2,] -0.0833333497  0.0000000982 -0.0000001146  0.0833333661
[3,] -0.0833333006 -0.0000001146  0.0000000982  0.0833333170
[4,]  0.0000000819  0.0833330125  0.0833336705 -0.1666667649
```

One interesting aspect of this simplex example is that we can easily find a $\Delta$ where the derivative of $\omega$ is zero. The matrix $J$ is $6 \times 16$ and there are plenty of $\delta$ such that $J\delta = y$ for any $y$. We choose $y$ is $(1, 2, \cdots, 6)$, which is in the correct order.

We choose a symmetric solution $\Delta$.

```
     [,1] [,2] [,3] [,4]
[1,] +3.0 +3.5 +3.5 +0.0
[2,] +3.5 +5.0 +3.5 +0.0
[3,] +3.5 +3.5 +6.0 +0.0
[4,] +0.0 +0.0 +0.0 +0.0
```

The derivatives of $\omega$ at this $\Delta$ are indeed zero. If we choose $\epsilon$ equal to $1e-6$ then the distances for $X_0 + \epsilon\Delta$ are

```
         1        2        3
2 1.000001                  
3 1.000001 1.000003         
4 1.000002 1.000004 1.000004
```

Note that these are in the correct order, and thus $\sigma(X_0 + \epsilon\Delta)$ is zero, and so are the its derivatives. We have found a solution $X$ with zero stress, which is practically indistinguishable from a regular tetrahedron and which has a derivative equal to zero. Any iterative algorithm would stop happily at this point, instead of keeping away from it because it is indistinguishable from a solution with stress equal to $0/0$.

For the simplex example there is a more direct way to accomplish the same result. Take a distance matrix $D$ with all distances equal to one. The Torgerson transform (minus one half times the doubly-centered squared distances) is of rank $n-1$ and has three eigenvalues equal to $\frac{1}{2}$. Take a second hollow symmetric non-negative matrix $E$ with elements in the "correct" order and perturb to $D(\epsilon)=D+\epsilon E$. Then the elements of $D(\epsilon)$ are in the correct order and for small enough positive epsilon the Torgerson transform will still be of rank $n-1$ with three positive eigenvalues. Stress-formula-two and its derivatives will be zero at the classic scaling solution, which will be as indistinguishable as you like from a regular simplex.

## 3.6 Second

Suppose we have $n=4$ and the four model functions $d_{12}(X), d_{23}(X), d_{34}(X)$, and $d_{14}(X)$. Thus $d_{13}(X)$ and $d_{24}(X)$ are missing. For $X_0$ we use the matrix

$$\begin{bmatrix}0&0\\1&0\\1&1\\0&1\end{bmatrix}$$

These are the four corners of a square, for which the four distances in the model are all equal to one. As in the previous example

$$\{J\}_{ij,ks}=\mathcal{D}_{ks}d_{ij}(X_0)=(x_{is}^0-x_{js}^0)(\delta^{ik}-\delta^{jk}),$$

which is a $4\times 8$ matrix. Also

$$\{J\delta\}_{ij}=\sum_{k=1}^{4}(x_{is}^0-x_{js}^0)(\delta^{ik}-\delta^{jk})\delta_{ks}=\sum_{s=1}^{2}(x_{is}^0-x_{js}^0)(\delta_{is}-\delta_{js})$$

Or

$$\begin{aligned}\{J\delta\}_{12}&=-(\delta_{11}-\delta_{21}),\\ \{J\delta\}_{23}&=-(\delta_{22}-\delta_{32}),\\ \{J\delta\}_{34}&=+(\delta_{31}-\delta_{41}),\\ \{J\delta\}_{14}&=-(\delta_{12}-\delta_{42}).\end{aligned}$$

Again, it is easy to choose $\delta$ such that the $\{J\delta\}_{ij}$ are in the correct order, it is also easy to choose them such that $\{J\delta\}$ is zero and both $\omega_S$ and $\sigma_S$ are undefined.

## 3.7 MDU

Our theorems also apply to a generalization of what we could call Roskam's stress, giving credit to Roskam (1968), although this stress was proposed at the same time by Kruskal and Carroll (1969).

In MDU there are two configurations $X$ and $Y$ and

$$\sigma(X,Y)=\sum_{i=1}^n\frac{\|d(x_i,Y)-P_i^K(d(x_i,Y))\|^2}{\|d(x_i,Y)-P_i^L(d(x_i,Y))\|^2}$$

o

$$\|d-P_K(d)\|^2=\frac{\epsilon^2}{r_i^2}\|z_0'\tilde{\eta}_j-P_i(z_0'\tilde{\eta}_j)\|^2+o(\epsilon^2)$$

# 4 Discussion

Interpretation

For squared distance models: same formulas

# 5 Code

Code is written in R (R Core Team (2024)). We use the monotone() function from F. M. T. A. Busing (2022) for the monotone regressions and the numerical differentiation routines from Gilbert and Varadhan (2019) to check our formulas.

```r
library(monotone)
library(numDeriv)

# functions y1() and y2() generate perturbations for any n

y1 <- function(n) {
  return(outer(1:n, 1:n) / n)
}

y2 <- function(n) {
  return(matrix(1:(n^2) %% (n - 1), n, n))
}

# perturbLoss() computes stress and omega for any perturbation

perturbLoss <- function(n, eps, y) {
  x <- sqrt(1 / 2) * diag(n)
  z <- x + eps * y
  dz <- as.vector(dist(z))
  mz <- monotone(dz)
  nz <- mean(dz)
  sr <- sum((dz - mz)^2) / 2
  sl <- sum((dz - nz)^2) / 2
  ss <- sr / sl
  dy <- NULL
  for (j in 1:(n - 1)) {
    for (i in (j + 1):n) {
      ey <- y[i, i] + y[j, j] - y[i, j] - y[j, i]
      dy <- c(dy, sqrt(1 / 2) * ey)
    }
  }
  my <- monotone(dy)
  ny <- mean(dy)
  ey <- sum((dy - my)^2)
  fy <- sum((dy - ny)^2)
  ty <- 0.5 * ey * eps^2
  qy <- 0.5 * fy * eps^2
  sy <- ey / fy
  return(c(ss, sy))
}

# perturbDerivatives() computes dsigma near a
# trivial solution and the approximation
# domega

perturbDerivatives <- function(n, eps, y) {
  s <- sqrt(1 / 2)
  x <- s * diag(n)
  z <- x + eps * y
  dz <- as.vector(dist(z))
  mz <- monotone(dz)
  nz <- mean(dz)
  sr <- sum((dz - mz)^2) / 2
  sl <- sum((dz - nz)^2) / 2
  ss <- sr / sl
  b1 <- b2 <- matrix(0, n, n)
  k <- 1
  for (j in 1:(n - 1)) {
    for (i in (j + 1):n) {
      b1[i, j] <- b1[j, i] <- -(dz[k] - mz[k])
      b2[i, j] <- b2[j, i] <- -(dz[k] - nz)
      k <- k + 1
    }
  }
  diag(b1) <- -rowSums(b1)
  diag(b2) <- -rowSums(b2)
  # gz is the analytical expression for Dsigma(X0+eps * Delta)
  gz <- ((b1 %*% z) - ss * (b2 %*% z)) / sl
  dy <- NULL
  for (j in 1:(n - 1)) {
    for (i in (j + 1):n) {
      ey <- y[i, i] + y[j, j] - y[i, j] - y[j, i]
      dy <- c(dy, sqrt(1 / 2) * ey)
    }
```

```r
  }
  my <- monotone(dy)
  ny <- mean(dy)
  ey <- sum((dy - my)^2) / 2
  fy <- sum((dy - ny)^2) / 2
  oy <- ey / fy
  b1 <- b2 <- matrix(0, n, n)
  k <- 1
  for (j in 1:(n - 1)) {
    for (i in (j + 1):n) {
      b1[i, j] <- b1[j, i] <- -(dy[k] - my[k])
      b2[i, j] <- b2[j, i] <- -(dy[k] - ny)
      k <- k + 1
    }
  }
  diag(b1) <- -rowSums(b1)
  diag(b2) <- -rowSums(b2)
  # gy us the analytic expression for Domega(y)
  gy <- s * (b1 - oy * b2) / fy
  return(list(
    ss = ss,
    oy = oy,
    gz = gz,
    gy = gy
  ))
}

# nStressDerivative computes the derivative of stress formula two
# using numerical differentiation

nStressDerivative <- function(n, eps, y) {
  numFunc <- function(z, n) {
    z <- matrix(z, n, n)
    dz <- as.vector(dist(z))
    mz <- monotone(dz)
    nz <- mean(dz)
    sr <- sum((dz - mz)^2) / 2
    sl <- sum((dz - nz)^2) / 2
    ss <- sr / sl
  }
  s <- sqrt(1 / 2)
  x <- s * diag(n)
  z <- x + eps * y
  return(matrix(jacobian(numFunc, as.vector(z), n = n), n, n))
}

# nOmegaDerivative computes the derivative of omega
# using numerical differentiation

nOmegaDerivative <- function(n, eps, y) {
  numFunk <- function(z, n) {
    z <- matrix(z, n, n)
    dy <- NULL
    for (j in 1:(n - 1)) {
      for (i in (j + 1):n) {
        dy <- c(dy, z[i, i] + z[j, j] - z[i, j] - z[j, i])
      }
    }
    my <- monotone(dy)
    ny <- mean(dy)
    sr <- sum((dy - my)^2) / 2
    sl <- sum((dy - ny)^2) / 2
    ss <- sr / sl
  }
  s <- sqrt(1 / 2)
  x <- s * diag(n)
  z <- x + eps * y
  return(matrix(jacobian(numFunk, as.vector(y), n = n), n, n))
}

# matrixPrint() is a small utility to print a rectangular matrix

matrixPrint <- function(x,
                        digits = 10,
                        width = 15,
                        format = "f",
                        flag = "") {
  print(noquote(
    formatC(
      x,
      digits = digits,
      width = width,
      format = format,
      flag = flag
    )
  ))
}
```

# References

Busing, F. M. T. A. 2022. “Monotone Regression: A Simple and Fast O(n) PAVA Implementation.” *Journal of Statistical Software* 102 (Code Snippet 1).

Busing, Frank M. T. A. 2010. “Advances in Multidimensional Unfolding.” PhD thesis, Leiden University. https://scholarlypublications.universiteitleiden.nl/access/item%3A2837800/view.

De Leeuw, J. 1983. “On Degenerate Multidimensional Unfolding Solutions.” Research Report. Leiden, The Netherlands: Department of Data Theory FSW/RUL.

Gilbert, P., and R. Varadhan. 2019. *numDeriv: Accurate Numerical Derivatives*. https://CRAN.R-project.org/package=numDeriv.

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.

Kruskal, J. B., and J. D. Carroll. 1969. “Geometrical Models and Badness of Fit Functions.” In *Multivariate Analysis, Volume II*, edited by P. R. Krishnaiah, 639–71. North Holland Publishing Company.

R Core Team. 2024. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Roskam, E. E. 1968. “Metric Analysis of Ordinal Data in Psychology.” PhD thesis, University of Leiden.

Takane, Y., F. W. Young, and J. De Leeuw. 1977. “Nonmetric Individual Differences in Multidimensional Scaling: An Alternating Least Squares Method with Optimal Scaling Features.” *Psychometrika* 42: 7–67.

Van Deun, K. 2005. “Degeneracies in Unfolding.” PhD thesis, KU Leuven. https://www.academia.edu/455100/Degeneracies_in_unfolding.

Van Deun, K., P. L. F. Groenen, W. J. Heiser, F. M. T. A. Busing, and L. Delbeke. 2005. “Interpreting Degenerate Solutions in Unfolding by Use of the Vector Model and the Compensatory Distance Model.” *Psychometrika* 70: 45–69.