NOTES ON NON-METRIC MULTI-DIMENSIONAL SCALING

Suppose we have a matrix of *dissimilarities* (distances) for 5 stimuli. That is

	z_1	z_2	z_3	z_4	z_5
z_1	0				
z_2	5	0			
z_3	6	2	0		
z_4	1	7	10	0	
z_5	4	9	3	8	0

Where z_1 and z_4 are the most similar, $\delta_{41} = \delta_{14} = 1$, and z_4 and z_3 are the most dissimilar, $\delta_{43} = \delta_{34} = 10$.

What we wish to do is find 5 **z** points in a space of a given dimensionality such that their interpoint distances reproduce the rank ordering of the dissimilarity matrix. That is,
let

$$d_{jm} = \left[\sum_{k=1}^{s} \left(z_{jk} - z_{mk} \right)^2 \right]^{1/2} \qquad (1)$$

be the distance between z_j and z_m. In the best of all possible worlds we would like to see the following:

$$\text{if } \delta_{jm} < \delta_{hg} \text{ then } \mathbf{d}_{jm} < \mathbf{d}_{hg} \qquad (2)$$

where, h, g, j, m all index the stimuli.

If we found 5 **z** points such that the rank ordering of the δ's was exactly reproduced, then we would have a "perfect" solution.

What we *want* to do is clear enough, but how do we find the **z**'s? We can't just pick a configuration of **z**'s randomly and check to see if it reproduces the rank orderings. This would be impractical. What we need is a *systematic* method of moving the **z**'s around so that we get closer and closer to reproducing the rank ordering of the **d**'s. Kruskal invented just such a systematic way of doing this.

Assume that the observed dissimilarities are unknown transformations of d_{jm}. That is:

$$\delta_{jm} = f(d_{jm}) \qquad (3)$$

where the function **f()** could be any function as long as it is *weakly monotone*. That is:

$$\text{if } \delta_{jm} < \delta_{hg} \text{ then } \mathbf{d}_{jm} \leq \mathbf{d}_{hg}\,. \qquad (4)$$

The task of non-metric multidimensional scaling (MDS) is to recover the **z**'s and the unknown transformation **f()**.

Kruskal’s ingenious solution consists of two parts. First, he created a loss function he dubbed *STRESS* that contains the rank ordering condition; and second, he came up with an algorithm that was part ordinary gradient minimization and part what he dubbed *monotone regression* that minimized STRESS (more on this below).

To start the process, suppose we pick an arbitrary configuration for the **z**'s and compute the $\mathbf{d}_{jm}$. We then construct the following table:

j,m	δ_{jm}	$\mathbf{d}_{jm}^{(start)}$	$\hat{\mathbf{d}}_{jm}^{(1)}$	$\mathbf{d}_{jm}^{(1)}$	$\hat{\mathbf{d}}_{jm}^{(2)}$
4,1	1	.1	.1	.101	.101
3,2	2	***.15***	.135	.14	.14
5,3	3	***.12***	.135	.14	.14
5,1	4	.8	.8	.8	.8
2,1	5	1.0	1.0	***1.005***	1.0
3,1	6	1.0	1.0	***.995***	1.0
4,2	7	***1.3***	1.2	1.16	1.16
5,4	8	***1.2***	1.2	***1.17***	1.165
5,2	9	***1.1***	1.2	***1.16***	1.165
4,3	10	2.0	2.0	2.0	2.0

The violations of rank order are shown by bold-italic (see the third column--our starting values). Clearly our arbitrary configuration is close to what we want because the violations are not serious.

How do we proceed from here? We have to move the **z**'s around to reduce the violations of rank order. Kruskal's solution for this problem was to produce distances, $\hat{\mathbf{d}}$'s in his terminology, that are in the proper rank order vis a vis the δ's, and then try to reproduce them.

To see how he did this, recall that the simple squared-error loss function for the metric dissimilarities problem is:

$$\mu = \sum_{j=1}^{q}\sum_{m=1}^{q}\left(\hat{\mathbf{d}}_{jm} - \mathbf{d}_{jm}\right)^2 \qquad (5)$$

Now, suppose we minimize $\boldsymbol{\mu}$ subject to the constraint:

$$\text{if } \delta_{jm} < \delta_{hg} \text{ then } \hat{\mathbf{d}}_{jm} \leq \hat{\mathbf{d}}_{hg}$$

$\boldsymbol{\mu}$ is continuous and differentiable everywhere except when $\mathbf{d}_{jm} = 0$, so standard methods of steepest descent (gradient methods) can be used to find **z**'s that minimize $\boldsymbol{\mu}$. For example, from the solution to the metric similarities problem the update formula for the **z**'s is:

$$\mathbf{z}_{jk}^{(h)} = \frac{1}{\mathbf{q}} \sum_{m=1}^{q} \left[\mathbf{z}_{mk}^{(h-1)} + \frac{\hat{\mathbf{d}}_{jm}^{(h-1)}}{\mathbf{d}_{jm}^{(h-1)}} \left(\mathbf{z}_{jk}^{(h-1)} - \mathbf{z}_{mk}^{(h-1)} \right) \right] \tag{6}$$

Where h is the iteration number. The closer the $\mathbf{d}_{jm}$ are to the $\hat{\mathbf{d}}_{jm}$, the closer they are to reproducing the rank ordering of the δ's.

The $\hat{\mathbf{d}}$'s are produced from the **d**'s by a method Kruskal dubbed monotone regression. The way this is done is quite simple. Referring back to the Table, Kruskal simply takes the **d**'s (in bold-italic in the Table) that are out of order, adds them up and divides by their number to produce the corresponding $\hat{\mathbf{d}}$'s. Those **d**'s that are in the proper order become $\hat{\mathbf{d}}$'s without any alteration (the $\hat{\mathbf{d}}^{(1)}$'s in the Table). The rationale for this is straightforward. The **d**'s are exact. We want to alter them as little as possible in order to get the $\hat{\mathbf{d}}$'s which are in the correct rank order. If we don't alter the **d**'s very much, we won't have to move the **z**'s around very much.

For example, the sum of squared error, μ, between the starting **d**'s and the $\hat{\mathbf{d}}$'s computed from them is

$$\mu^{(1)} = \sum_{j=1}^{q} \sum_{m=1}^{q} \left(\hat{\mathbf{d}}_{jm}^{(1)} - \mathbf{d}_{jm}^{(start)} \right)^2 = (.015)^2 + (-.015)^2 + (.1)^2 + (-.1)^2$$

We now find **z**'s that minimize $\mu^{(1)}$. This gives us $\mathbf{d}_{jm}^{(1)}$. Again, recall that the $\mathbf{d}^{(1)}$'s are *exact*--that is they are computed from a known configuration of **z**'s. We form $\hat{\mathbf{d}}^{(2)}$'s from the $\mathbf{d}^{(1)}$'s and our squared error is now:

$$\mu^{(2)} = \sum_{j=1}^{q}\sum_{m=1}^{q}\left(\hat{\mathbf{d}}_{jm}^{(2)} - \mathbf{d}_{jm}^{(1)}\right)^2 = (.005)^2 + (-.005)^2 + (.005)^2 + (-.005)^2$$

Notice that $\mu^{(1)} > \mu^{(2)}$. To see this, first note that it must be the case that

$$\mu^{(1)} = \sum_{j=1}^{q}\sum_{m=1}^{q}\left(\hat{\mathbf{d}}_{jm}^{(1)} - \mathbf{d}_{jm}^{(start)}\right)^2 \geq \sum_{j=1}^{q}\sum_{m=1}^{q}\left(\hat{\mathbf{d}}_{jm}^{(1)} - \mathbf{d}_{jm}^{(1)}\right)^2$$

because we could always use $\mathbf{z}^{(start)}$ for $\mathbf{z}^{(1)}$ so that $\mathbf{d}^{(1)} = \mathbf{d}^{(start)}$. By the principle of least squares it must be the case that:

$$\sum_{j=1}^{q}\sum_{m=1}^{q}\left(\hat{\mathbf{d}}_{jm}^{(1)} - \mathbf{d}_{jm}^{(1)}\right)^2 \geq \sum_{j=1}^{q}\sum_{m=1}^{q}\left(\hat{\mathbf{d}}_{jm}^{(2)} - \mathbf{d}_{jm}^{(1)}\right)^2 = \mu^{(2)}$$

This will be true because both $\hat{\mathbf{d}}^{(1)}$ and $\hat{\mathbf{d}}^{(2)}$ *by construction* are a weakly monotone ordering of the δ's. However, $\hat{\mathbf{d}}^{(2)}$ is constructed from $\mathbf{d}^{(1)}$ so that $\mathbf{d}^{(1)}$ *by definition* is "closer" to $\hat{\mathbf{d}}^{(2)}$ than to $\hat{\mathbf{d}}^{(1)}$ which was constructed from $d^{(start)}$.

This ingenious algorithm of Kruskal's can be run until no further improvement is possible. That is, the algorithm is guaranteed to converge to a configuration of **z**'s such that the **d**'s they produce in turn produce $\hat{\mathbf{d}}$'s which in turn reproduce the original **z**'s.

Note that the method has a serious weakness. Because we are only imposing the *weak monotone* constraint, that is:

$$\text{if } \delta_{jm} < \delta_{hg} \text{ then } \hat{\mathbf{d}}_{jm} \leq \hat{\mathbf{d}}_{hg}$$

then setting all the coordinates equal to zero, that is, **z = 0**, *is a solution* because all the $\mathbf{d}_{jm} = 0$! In other words, using the metric similarities loss function in equation (5) works great *for a few iterations*! That is, given $\hat{\mathbf{d}}_{jm} \neq \mathbf{0}$ then the update formula in (6) will work fine. But in the next iteration the $\hat{\mathbf{d}}_{jm}$'s are created from the $\mathbf{d}_{jm}$'s. Because there is no constraint on the points they can very quickly collapse in on the origin because that is a solution.

Kruskal's solution for this was to normalize the squared error loss function by dividing it by the sum of the squared distances. He called this loss function STRESS:

$$\mathbf{S} = \sqrt{\frac{\mathbf{S*}}{\mathbf{T*}}} = \sqrt{\frac{\sum_{j=1}^{q}\sum_{m=1}^{q}\left(\hat{\mathbf{d}}_{jm} - \mathbf{d}_{jm}\right)^2}{\sum_{j=1}^{q}\sum_{m=1}^{q}\mathbf{d}_{jm}^2}} \quad (7)$$

Where **S*** is the squared error loss function given in equation (5).

The problem with STRESS is that the first derivatives that form the gradient vector *are not simple*! In particular:

$$\frac{\partial \mathbf{S}}{\partial \mathbf{z}_{jk}} = \frac{1}{2}\sqrt{\frac{\mathbf{T*}}{\mathbf{S*}}}\frac{\left(\mathbf{T*}\frac{\partial \mathbf{S*}}{\partial \mathbf{z}_{jk}} - \mathbf{S*}\frac{\partial \mathbf{T*}}{\partial \mathbf{z}_{jk}}\right)}{\left(\mathbf{T*}^2\right)} = \frac{1}{2}\mathbf{S}\left(\frac{1}{\mathbf{S*}}\frac{\partial \mathbf{S*}}{\partial \mathbf{z}_{jk}} - \frac{1}{\mathbf{T*}}\frac{\partial \mathbf{T*}}{\partial \mathbf{z}_{jk}}\right) =$$

$$\frac{1}{2}\mathbf{S}\left\{\begin{array}{l}\frac{1}{\mathbf{S*}}\left[-2\sum_{\mathbf{m=1}}^{\mathbf{q}}\left\{\left(\frac{\widehat{\mathbf{d}}_{\mathbf{jm}}}{\mathbf{d}_{\mathbf{jm}}}-1\right)\left(\mathbf{z}_{\mathbf{jk}}-\mathbf{z}_{\mathbf{mk}}\right)\right\}\right]- \\ \qquad\qquad \frac{1}{\mathbf{T*}}2\sum_{\mathbf{m=1}}^{\mathbf{q}}\left\{\mathbf{d}_{\mathbf{jm}}\left(\frac{1}{2}\right)\left[\sum_{\mathbf{k=1}}^{\mathbf{s}}(\mathbf{z}_{\mathbf{jk}}-\mathbf{z}_{\mathbf{mk}})^2\right]^{-\frac{1}{2}}\left(2\left[\mathbf{z}_{\mathbf{jk}}-\mathbf{z}_{\mathbf{mk}}\right]\right)\right\}\end{array}\right\}=$$

$$\mathbf{S}\left\{-\frac{1}{\mathbf{S*}}\sum_{\mathbf{m=1}}^{\mathbf{q}}\left[\left(\frac{\widehat{\mathbf{d}}_{\mathbf{jm}}}{\mathbf{d}_{\mathbf{jm}}}-1\right)\left(\mathbf{z}_{\mathbf{jk}}-\mathbf{z}_{\mathbf{mk}}\right)\right]-\frac{1}{\mathbf{T*}}\sum_{\mathbf{m=1}}^{\mathbf{q}}\left(\mathbf{z}_{\mathbf{jk}}-\mathbf{z}_{\mathbf{mk}}\right)\right\}$$

Unfortunately, these derivatives do not have a nice simple interpretation like those for the metric similarities problem.