Nonmetric Multidimensional Scaling, Fit Assessment, and Interpretation

Measurement, Scaling, and Dimensional Analysis
2019 ICPSR Summer Program
Prof. Adam M. Enders

Nonmetric Multidimensional Scaling

- General strategy:
  - Obtain scaling solution in a specified dimensionality
  - If fit is adequate, stop and report results
  - If fit is poor, try solution in higher dimensionality
- General procedure:
  - Begin with initial configuration of points in space
  - Move points around, as necessary, to make distances between points *monotonic with dissimilarities*

Example: Feelings Toward Political Candidates

- Dissimilarities among four American presidential candidates: Cruz, Paul, Bush, and Trump
- This is a very small dataset!
  - Useful for illustrating the steps of the scaling process
  - Note that we would never perform a "real" nonmetric MDS with only four objects (which produce six dissimilarities)
  - Too few metric constraints – the scaling solution (i.e., the point locations) could be changed without affecting the fit to the data
  - This problem is alleviated when more objects are included in the scaling analysis

Hypothetical Data Matrix, Δ

Matrix of rank-ordered dissimilarities among four presidential candidates

	Cruz	Paul	Bush	Trump
Cruz	—	1	5	6
Paul	1	—	2	3
Bush	5	2	—	4
Trump	6	3	4	—

It is easy to demonstrate that these dissimilarities cannot be represented accurately with a unidimensional array of points

So, try a two-dimensional solution...

Strategy for Nonmetric MDS

- Start with random configuration of points in two-dimensional space
  - We do not take this configuration seriously as a scaling solution; it just provides a neutral starting position
  - Use Pythagorean formula to calculate distances between points in the random configuration
  - Distances should be monotonic to dissimilarities data, but they probably are not
- Move points around to create a new configuration that is closer to the objective of a monotonic relationship between dissimilarities and distances
  - Want to be as efficient with movements as possible – no unnecessary movements

Distances and Disparities

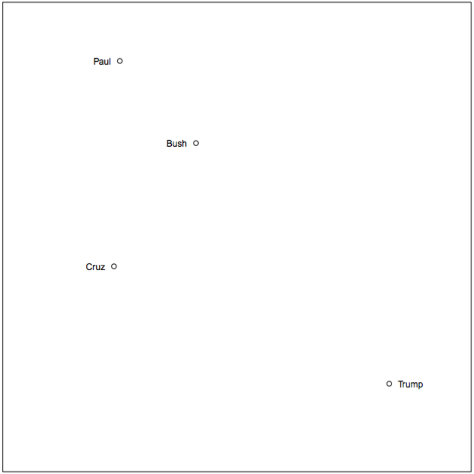
- Create a set of "target distances" that can be compared to current actual distances in order to guide point movements
- These target distances are called "disparities" in the MDS literature
  - There is a disparity (target distance) associated with each distance between a pair of points
  - The disparity associated with distance d_{ij} is designated $\hat{d}_{ij}$
- The disparities guide the point movements
  - If two points are too close together, then the disparity will be larger than the current distance (i.e., $\hat{d}_{ij} > d_{ij}$); if they are too far apart, then the disparity will be smaller than the current distance (i.e., $\hat{d}_{ij} < d_{ij}$)

Properties of Disparities

- Disparities are characterized by two important properties:
  1. Disparities are as similar to the actual distances as possible
     - That is, the correlation between the distances and the disparities ($r_{d_{ij}\hat{d}_{i}j}$) is maximized
  2. Disparities are *always* monotonic to dissimilarities, even if the associated distances are not
     - That is, if $\delta_{ij} < \delta_{il}$ then $\hat{d}_{ij} \leq \hat{d}_{il}$, even if $d_{ij} > d_{il}$

Random Starting Configuration

Point configuration obtained by generating random coordinates for the four presidential candidates:



First Set of Distances and Disparities

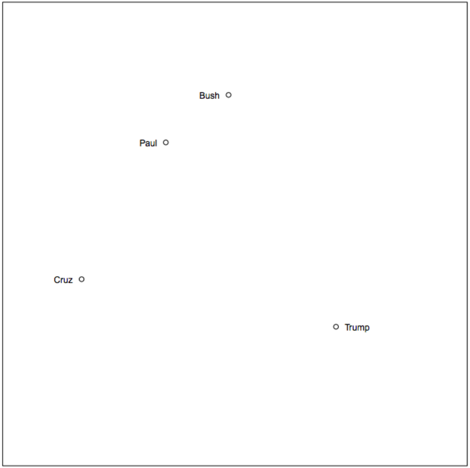
Stimulus Pair ij	Distance: d_{ij}	Disparity: $\hat{d}_{ij}$	Difference $d_{ij} - \hat{d}_{ij}$
Cruz-Paul	3.50	2.70	0.80
Paul-Bush	1.91	2.70	-0.79
Paul-Trump	7.17	4.98	2.19
Bush-Trump	5.26	4.98	0.28
Cruz-Bush	2.52	4.98	-2.46
Cruz-Trump	5.11	5.11	0.00

Properties of Disparities

- Use difference between distances and disparities to guide point movements:
  - For each pair, move the points along the line connecting them
  - If $d_{ij} - \hat{d}_{ij}$ is a positive value, current distance is larger than the target – so, move points representing i and j closer together
  - If $d_{ij} - \hat{d}_{ij}$ is a negative value, current distance is smaller than the target – so, move points representing i and j farther apart
  - Size of movement is determined by absolute value of $d_{ij} - \hat{d}_{ij}$
  - For present purposes, does not really matter which pair of points is moved first

Second Point Configuration

Point configuration obtained after making first set of moves, based upon previously-calculated disparities:

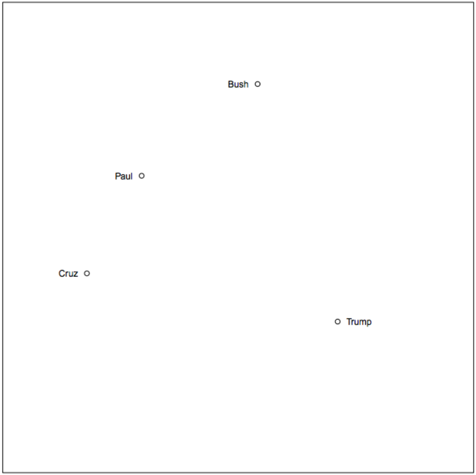


Second Set of Distances and Disparities

Stimulus Pair ij	Distance: d_{ij}	Disparity: $\hat{d}_{ij}$	Difference $d_{ij} - \hat{d}_{ij}$
Cruz-Paul	2.77	2.065	0.705
Paul-Bush	1.36	2.065	-0.705
Paul-Trump	4.32	4.263	0.057
Bush-Trump	4.41	4.263	0.147
Cruz-Bush	4.06	4.263	-0.203
Cruz-Trump	4.46	4.460	0.000

Third Point Configuration

Point configuration obtained after making second set of moves, based upon second set of calculated disparities:



Third Set of Distances and Disparities

Stimulus Pair ij	Distance: d_{ij}	Disparity: $\hat{d}_{ij}$	Difference $d_{ij} - \hat{d}_{ij}$
Cruz-Paul	1.90	1.90	0
Paul-Bush	2.51	2.51	0
Paul-Trump	4.15	4.15	0
Bush-Trump	4.26	4.26	0
Cruz-Bush	4.33	4.33	0
Cruz-Trump	4.34	4.34	0

Final Nonmetric MDS Solution

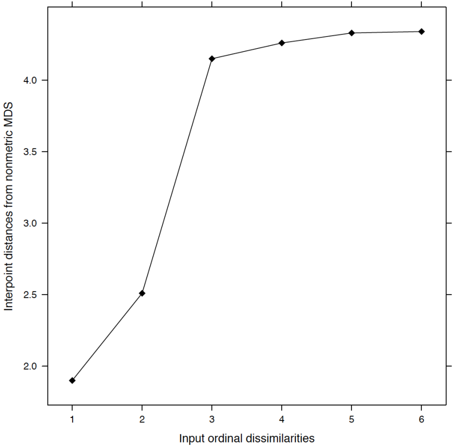
- With the third set of points, the disparities are always equal to the actual interpoint distances
  - The distances are monotonic to the ordinal dissimilarities
  - There is no reason to carry out further point movements
  - The objective of the nonmetric MDS has been achieved
- This solution fits the data *perfectly*

Steps in a Nonmetric MDS Routine

1. Starting configuration (most software uses Torgerson's metric MDS procedure)
2. Calculate fit for starting configuration
    - If perfect, then terminate
3. Calculate partial derivatives, move points
4. Calculate disparities and fit for new configuration
    - If perfect, then terminate
    - If no change, or fit worsens, then terminate
    - If fit is improving, go to Step 3 and repeat
5. Terminate MDS routine and print results

Graphical Depiction of Fit

Shepard diagram for simulated nonmetric MDS, showing scaled distances versus input ordinal dissimilarities:



Need for a Measure of Fit

- Perfect fit in scaling solution really only due to artificial nature of problem
- Problematic, because too few metric constraints in data to create sufficiently stable scaling solution
- Using nonmetric MDS on “real” data
  - More objects to be scaled
  - More difficult to obtain a *perfect* solution
  - But, often can reach a *very good* solution
  - Need to develop a fit measure that provides a summary of degree to which a nonmetric MDS solution achieves its analytic objective

Fit Statistic for Nonmetric MDS

- Logic:
  - Want distances to be monotonic to dissimilarities
  - Disparities are as close to distances as possible, but also monotonic to dissimilarities
  - So, develop a measure that summarizes how different the distances are from the disparities
- Kruskal's $Stress_1$ coefficient:

$$Stress_1 = \left[\frac{\sum_{i \neq j}^{\#pairs} (d_{ij} - \hat{d}_{ij})^2}{\sum_{i \neq j}^{\#pairs} d_{ij}^2} \right]$$

Alternative Fit Measures for Nonmetric MDS

- Kruskal's $Stress_2$ coefficient:

$$Stress_2 = \left[\frac{\sum_{i \neq j}^{\#pairs} (d_{ij} - \hat{d}_{ij})^2}{\sum_{i \neq j}^{\#pairs} (d_{ij} - \bar{d})^2} \right]^{0.5}$$

- Correlation between scaled distances and disparities, $r_{d\hat{d}}$
  - Logic is that disparities are optimally-scaled versions of the input data in the nonmetric MDS
- Several other fit measures are available (e.g., S-Stress; Guttman's uncorrected correlation coefficient)

Interpreting Stress

- The software we'll be using reports Stress_1, though the algorithm (which we'll talk about momentarily) actually reduces "raw Stress" (no scaling factor/denominator)
- With either Stress_1 or Stress_2, the smaller the better
- Rules of thumb:
  - Stress ≤ 0.05: excellent fit
  - Stress ≤ 0.10: good fit
  - Stress ≤ 0.15: OK/acceptable fit
  - Stress ≥ 20: bad fit, not acceptable
- Some caveats:
  - Should always use theory to guide decisions
  - Be sure to look at the output if you aren't sure...the whole point of MDS is graphical representation of structure
  - In a set dimensionality, Stress is usually going to increase with the number of objects to be scaled (unless the structure in the data is very clear)

Procedure for a "Real" Nonmetric MDS Routine

- The value of the Stress coefficient is a function of the point coordinates in the current configuration (which are used to calculate the distances and, indirectly, the disparities)
- Therefore, can calculate the partial derivatives of *Stress*, relative to the coordinates:

$$\frac{\partial Stress_1}{\partial x_{ip}}$$

- For $i = 1, 2, \ldots, k$ and $p = 1, 2, \ldots, m$
- Partial derivatives show how Stress changes when point coordinates are changed by a minute amount
- Therefore, change point coordinates in ways that make the partial derivatives the smallest possible negative values
- This is the "steepest descent" approach recommended by Kruskal

SMACOF

- SMACOF = Scaling by MAjorizing a COmplicated Function
- An optimization strategy for finding a set of point coordinates that minimize a (raw) stress function

$$\text{Stress}_r = \sum_{i<j\leq n} w_{ij}(d_{ij} - \delta_{ij})^2$$

- $w_{ij} \geq 0$ is a weight for the measurement between a pair of points (i, j), d_{ij}
- (Sidenote: inclusion of the weights was a huge step because it allowed for missing data and other types of models, like WMDS or confirmatory models)
- This iterative majorization algorithm underlies all of the functions in the smacof R package, as well as PROXSCAL in SPSS

Dealing with "Ties"

- Consider imperfect ranked data, ratings of objects along by some criterion, or proportions of times two objects behaved in the same way
- There is a potential for "ties" in the input dissimilarities data
  - That is, two pairs of objects could take on the same value
  - $\delta_{ij} = \delta_{kl}$
- (At least) two ways to handle this case when estimating MDS model (could always try to sort things out theoretically before modeling):
  1. "Primary" method: ties can be "broken" in corresponding distances
  2. "Secondary" method: ties must be preserved
- The default in most (all, that I know of) MDS software is "primary" since it is less restrictive and doesn't tend to make a difference with respect to substantive interpretation

Stress-Per-Point

- Can also decompose the model Stress coefficient per object (stimulus point)
- This provides us some understanding of which points are contributing most to Stress (which, again, we want to be low)
- Usually expressed as a percentage – what proportion of Stress is due to a particular object
- High SPP objects are sort of like outliers in a regression
  - Like outliers, some can be *influential*, other not so much
  - Like outliers, we want to be careful about carelessly removing objects, as well
- Not really good rules of thumb for how much is "too" much
- More a question of some objects contributing more to Stress than others

Other Ways of Assessing Fit

- Don't want to limit ourselves to Stress alone
  - One important problem with a mechanical reliance on Stress rules of thumb is that Stress is dependent on the number of objects being scaled
  - The larger the number of objects, the larger the Stress
- Can use simulations and resampling to better understand model fit
- Two additional options for deciphering fit
  1. Permutation tests
  2. Bootstrapping the configuration and looking at confidence ellipses

Permutation Tests

- Question: is Stress/configuration randomly obtained (accidentally "good")?
- Basic idea: resample original data and see if Stress is equally low in some subsample
- Steps:
  1. Resample original, non-dissimilarities data X times
  2. Compute dissimilarities matrices and conduct MDS analyses
  3. Compare new Stress values to full sample Stress
  4. Compute p-values of null hypothesis that configuration is derived from random permutation of dissimilarities (i.e., want a small p-value)
- Caveat: built for data that was not collected as square, symmetric matrix of dissimilarities (which is usually the case)
- Can plot the empirical cumulative distribution Stress values

Bootstrapping MDS Models

- Question: how *stable* is the output point configuration?
  - Basically, can we find some measure of uncertainty, like a standard error in a typical statistical model?
- Just like we can bootstrap a regression model, we can bootstrap an MDS analysis
- We can use fluctuations in the locations each point to construct 95% confidence ellipses
- Steps:
  1. Resample original, non-dissimilarities data X times
  2. Compute dissimilarities matrices and conduct MDS analyses
  3. Use each MDS configuration to construct confidence ellipses (see Jacoby and Armstrong 2014)
- Caveat: can only be used data do not consist of direct dissimilarities judgements

Example: Country Perceptions

- Data from Wish 1971
- Asked 18 students to rate perceived similarity between each pair of 12 nations
- Used a nine-point scale ranging from 1 ("very different") to 9 ("very similar")
- Used this information to create a matrix of similarity ratings
- Nations: Brazil, Congo, Cuba, Egypt, France, India, Israel, Japan, China, USSR, USA, Yugoslavia

Interpreting Results

- MDS is inherently graphical, so the hope is that visualization will help with interpretation
- The locations of the coordinate axes for the point configuration are completely arbitrary
  - Final MDS point configuration usually rotated to a varimax orientation
  - Point coordinates usually standardized to a mean of zero on each axis and a variance of 1.0 (or some other specified value)
  - Axes have no intrinsic substantive importance or interpretation!
- Thoughts on purely visual interpretation:
  - Look for clusters of points
  - Look for "directions" or systematic variation across the span of the plotting region

Interpreting Results, cont'd

- Visualization and "eyeballing" the configuration should be considered a virtue of the technique
- BUT, this will make some people uncomfortable because it's too "subjective"
- Two other more systematic strategies for interpreting results:
  1. Embedding external variables
  2. Cluster analysis

Embedding External Variables

- Even though the coordinate axes are arbitrary and really shouldn't been substantively interpreted, we will probably want to apply some interpretation to the dimensions, or interesting sources of variation in the plot
- One way to do this is obtain some external measures of these substantive dimensions and embed an axis that measures this dimension into the MDS configuration
- Can do this very simply with OLS regression
- Assume an external variable, Y, is available:
  - Each of the k objects in the MDS have scores on the external variable, $y_1, y_2, \ldots, y_k$
- Regress Y on the MDS coordinate axes (Dim_1, Dim_2,. . . , Dim_m):

$$y_i = \alpha + \beta_1 \text{Dim}_{1i} + \beta_2 \text{Dim}_{2i} + \ldots + \beta_k \text{Dim}_{mi} + e_i$$

Embedding External Variables

- If the model fits well (i.e., R^2 is large, at least one of the dimension coordinates is statistically significant), then Y is consistent with the spatial configuration of object
- Possible interpretation:
  - The external variable, Y, is a substantive source of variability in the point locations within the MDS configuration
- Can take ratio of regression coefficients to obtain slope of line representing external variable, relative to a pair of the MDS axes. For example:

$$\text{Slope}_Y = \hat{\beta}_2 / \hat{\beta}_1$$

- Can locate a line with the preceding slope anywhere within the two-dimensional subspace of the MDS configuration
- It is usually convenient to run the line through the origin of the space

MDS Biplot

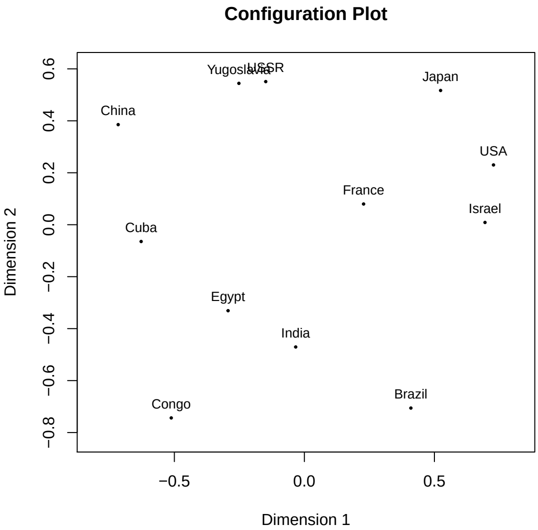
- Sometimes, this sort of plot is called an "MDS biplot"
- The information is exactly the same and the presentation is even very similar
- Difference:
  - Rather than represent the slope as a line that cuts across the plotting region, it represents the embedded variable as a vector
- Vector has the same exact orientation since it, too, is based on $\hat{\beta}_2$ and $\hat{\beta}_1$, which serve as coordinates for the terminal point
  - Orientation is the exact same as the slope approach
- Length of the vector is proportional to the R^2
  - Higher the R^2, longer the external variable vector
- Sometimes this approach is preferred if using multiple external variables because more information (direction, fit) is depicted for sake of comparison

Example: Country Perceptions Data

- Suppose I suspected that economic development had something to do with the MDS configuration of student country perceptions
- I could regress a new variable capturing the relative economic development of each country on the first and second dimension coordinates
  - "econdev": ranking of economic development of each country

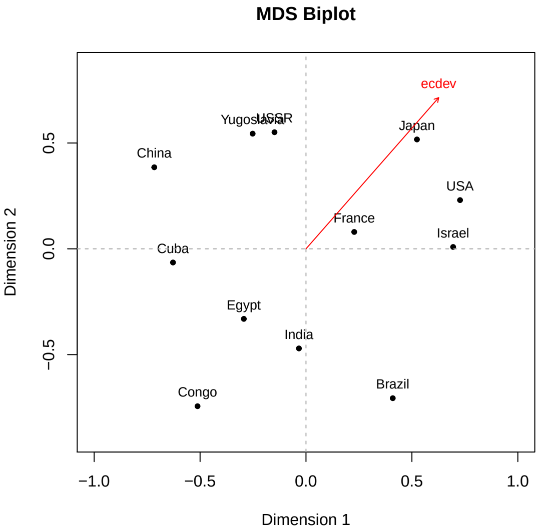
Example: Country Perceptions Data

Two-dimensional nonmetric configuration



Example: Country Perceptions Data

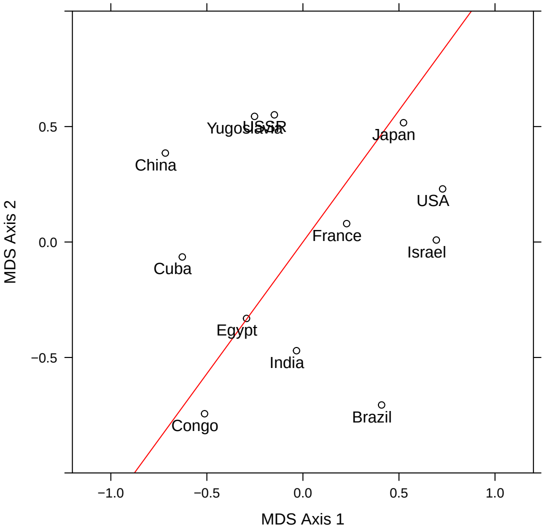
MDS biplot with economic development information embedded



Example: Country Perceptions Data

- Economic development equation:
  - $\text{EconDev}_i = 5.33 + 3.60\text{Dim}_{1i} + 4.10\text{Dim}_{2i} + e_i$
  - $R^2 = 0.931$
  - $\text{Slope} = \frac{4.10}{3.60} = 1.14$
- We can embed this estimated dimension into the configuration by drawing a line with a slope of 1.14 that runs through the centroid of the space

Example: Country Perceptions Data



Cluster Analysis

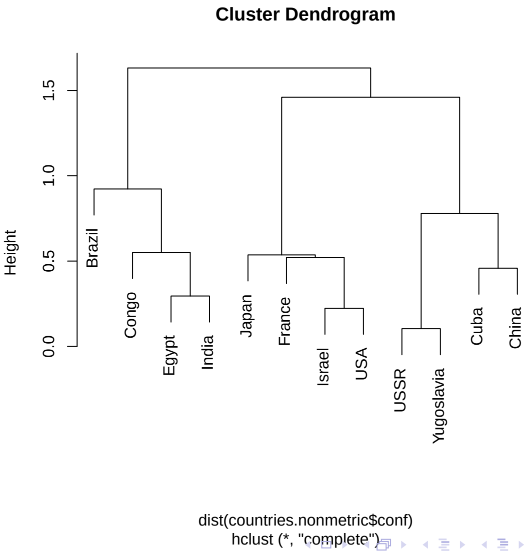
- Cluster analysis is a family of methods for creating taxonomies of objects
- Divides objects into a set of (usually) mutually exclusive categories
- Objects that are close to each other are placed into a common cluster; more distant objects are placed into different clusters
- There are many varieties of cluster analysis, but we'll "focus" (to be generous) on hierarchical clustering

Hierarchical Cluster Analysis

- Begin with each object in a separate cluster
- Proceed through $k-1$ steps, creating a new cluster on each step
- On each step, join together the two closest clusters
- The location of each new cluster is (usually) the mean location of the objects contained in the cluster
- At the $k-1$ step, all objects are joined in a common cluster
- Diagram called the "dendrogram" traces the steps of the clustering process
- Really, this is also pretty darn subjective, but it might help you convince others that your interpretation is "correct"
- Can use the "hclust" command in the base "stats" package in R to do this
  - Results are purely visual

Example: Country Perceptions Data

```
plot(hclust(dist(countries.nonmetric$conf)))
```

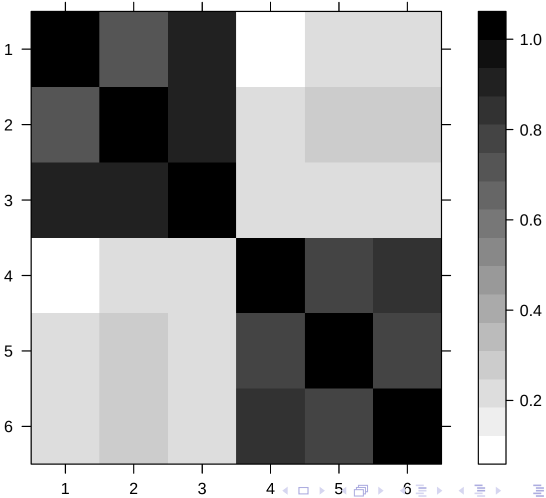


Degenerate Solutions

- Definition: Stress approaches zero (in nonmetric MDS) even though the MDS distances do not represent the data properly
- How do diagnose it?
  - Can usually tell by looking at MDS configuration plot and Shepard diagram
  - Usually, there will be a small number of clusters of object points that are very close to (or on top of) each other
- Why?
  - Most common reason: highly related clusters of objects that are very weakly or unrelated to the objects in the other cluster(s)
- Solutions:
  - Give up
  - Try estimating metric MDS, since we need the procedure to take even small dissimilarities seriously
  - Fancy: use a particular form for the monotonic function relating distances to dissimilarities

Example: Simulated Data from SVD Discussion

Really high correlations between (1, 2, 3) and (4, 5, 6), very weak correlations between pairs of variables across sets



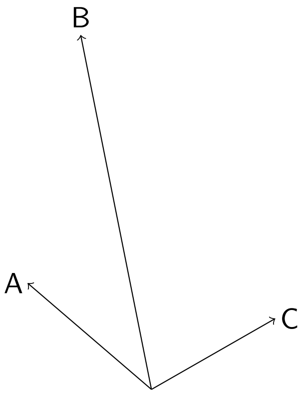
Obtaining Proximities/Dissimilarities

- Obvious choice: actual distances (in space or time units)
- Direct similarity judgements (e.g., ratings, paired comparisons/triads)
  - "How similar are object X and Y?"
  - "Which of the following two objects are most alike?"
  - "Which of the following 3 objects do you like the most? The least"
- Correlations
- Calculate distances between objects using a common, arithmetical distance measure:
  - Euclidean distance: $\delta_{ij} = \sqrt{\sum_{a=1}^{m}(x_{ia} - x_{ja})^2}$
  - "City block" distance: $\delta_{ij} = \sum_{a=1}^{m} |x_{ia} - x_{ja}|$
  - i and j denote objects of interest, a denotes a given attribute (of which there could be many)

Correlations as Input Dissimilarities

- Correlations are measures of angular separation between vectors
- MDS seeks to model inter-object distances that correspond as closely as possible to the input dissimilarities between objects
  - In geometric terms, MDS models the distances between variable vector terminal points, where correlations represent angular separation between variable vectors
- Thus, when using correlations as input data, MDS tends to "fan" the points in a circular shape
- There is nothing inherently wrong with this – MDS could still be accurately accounting for the input dissimilarities/proximities very well
- However, the circular structure is an artifact of using correlations

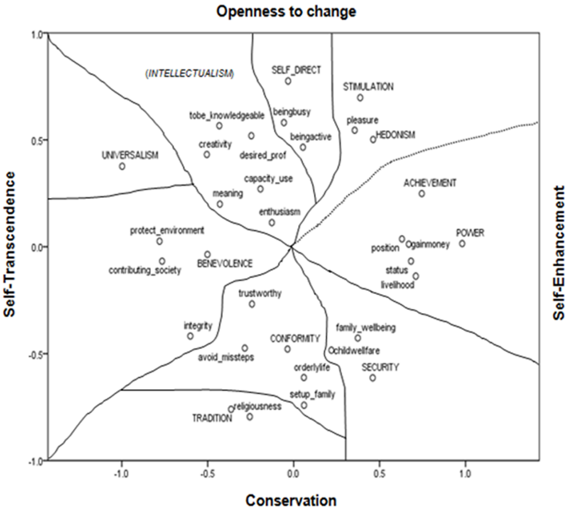
Hypothetical Example



- True distances:
  - dist(A, C) < dist(A, B) < dist(B, C)
- Incorrect distances inferred from magnitude of correlations:
  - dist(A, B) < dist(B, C) < dist(A, C)

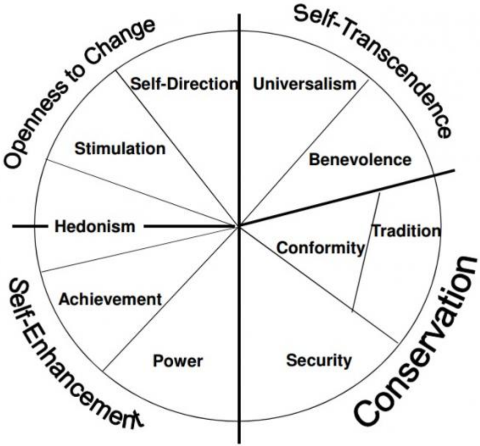
Example: Schwartz Value Circumplex

Schwartz and colleagues usually find solutions that look like this:



Example: Schwartz Value Circumplex

Which enables them to peddle a theory based on cleaned up version of the results that looks like this:



Steps to Conducting MDS Analysis

1. Convert data matrix to distances if not starting with square, symmetric matrix of dissimilarities
   - The "dist()" function in R will compute Euclidean distances between all pairs of row objects (and other types of distances if you'd like)
2. Determine level of measurement and submit dissimilarities matrix to either metric or nonmetric MDS algorithm
   - Will likely want to re-estimate model with different dimensionality to determine how many dimensions are necessary to "best" represent structure in the data
   - "Best": 1) produces a geometric representation that makes substantive sense, 2) low Stress

Steps to Conducting MDS Analysis

3. Consider model fit and basic diagnostics
   - Will want to report Stress value in paper
   - Should also consider Shepard plot and correlation (Pearson or Spearman) between input dissimilarities and output distances
4. Plot configuration and interpret
   - Look for clusters of points and interesting “directions”
   - Might also consider regressing an external variable into the MDS configuration or doing a cluster analysis

Procrustes Rotation

- Procrustes was an evil inn keeper and son of Poseidon
  - Invited passers-by to spend the night...in his special bed!
  - He would stretch short individuals to fit, amputate those who were too long
- Procrustes rotations are designed to rotate one matrix into maximal conformity to a target matrix
- Can use this procedure to compare configurations of points, something mere correlations can't get us
- Procedure:
  1. Center the columns of $\mathbf{X}$ (the target configuration) and $\mathbf{X}^*$ (the matrix to be rotated) so they sum to zero (most MDS software does this automatically)
  2. Calculate the product matrix, $\mathbf{X}'\mathbf{X}^*$ and its singular value decomposition: $\mathbf{X}'\mathbf{X}^* = \mathbf{UDV}'$
     - The optimal rotation matrix is $\mathbf{T} = \mathbf{VU}'$
     - The optimal dilation factor is $s = (tr\mathbf{X}'\mathbf{X}^*\mathbf{T}/(tr\mathbf{X}'\mathbf{X}^*)^{\mathbf{3}}$
     - The optimal translation vector is $t = n^{-1}(\mathbf{X} - s\mathbf{X}^*\mathbf{T})'1$