CovSTATIS: The basis of multi-table techniques for group and individual connectivity analyses

Jenny Rieck[1] & Derek Beaton[1]

[1] Rotman Research Institute, Baycrest Health Sciences

Author Note

An author note

Abstract

Some abstract

Keywords: keywords

Word count: X

CovSTATIS: The basis of multi-table techniques for group and individual connectivity analyses

Introduction

Methods

CovSTATIS

Here we primarily describe CovSTATIS, followed briefly by an explanation of CovSTATIS for distance matrices (DiSTATIS). We then discuss an extension of both techniques called "K+1 CovSTATIS" and "K+1 DiSTATIS", respectively, where there is an external or additional "target" table (the "$K+1$" amongst K tables).

Notation. Our notation rougly matches that of (CITE), but introduces slight differences and new information for our particular explanation. Bold uppercase letters denotes matrices (e.g., $\mathbf{R}$), bold lowercase letters denote vectors (e.g., $\mathbf{r}$), and italic lowercase letters denote specific elements (e.g., r). Upper case italic letters denote cardinality, size, or length (e.g., I). For example, $\mathbf{R}$ is an $I \times I$ matrix, which has I rows and I columns. Lower case subscript italic denotes a specific index (e.g., $_i$). A generic element of $\mathbf{R}$ at the $_i$th row and $_j$th column is $x_{i,j}$. Common letters of varying type faces, for example $\mathbf{R}$, $\mathbf{r}$, $r_{i,j}$, come from the same data struture. Vectors are assumed to be column vectors unless otherwise specified. Two matrices side-by-side denotes standard matrix multiplication (e.g., $\mathbf{RZ}$), where $\odot$ denotes element-wise (Hadamard) multiplication where $\oslash$ denotes element-wise (Hadamard) division. Superscript ^{T} denotes the transpose operation, superscript $^{-1}$ denotes standard matrix inversion, and superscript $^+$ denotes the Moore-Penrose pseudo-inverse. The diagonal operation, diag{}, transforms a vector into a diagonal matrix, or extracts the diagonal of a matrix and produces a vector. The vectorize operation, vec{}, transforms a matrix into a column vector.

The following letters have reserved and specific meanings. $\mathbf{I}$ denotes the identity

matrix. $\mathbf{R}$ denotes a generic but individual covariance (or correlation) matrix, where $\mathbf{R}_{[k]}$ denotes the $_k$th covariance matrix in a set of K covariance matrices. $\mathbf{S}$ denotes a generic but individual cross-product matrix, where $\mathbf{S}_{[k]}$ denotes the $_k$th covariance matrix in a set of K covariance matrices. $\mathbf{D}$ denotes a generic but individual distance matrix, where $\mathbf{D}_{[k]}$ denotes the $_k$th covariance matrix in a set of K covariance matrices. $\mathbf{L}$ is an external ("K+1") covariance matrix of exactly the same size as any of the $\mathbf{R}_{[k]}$ matrices. We discuss the properties of these tables when needed.

CovSTATIS. CovSTATIS is a particular form of cross-product STATIS that works for covariance (or correlation or equivalent) matrices. By definition, all individual connectivity matrices must meet the following conditions: (1) square (same number of rows and columns), (2) symmetric (upper and lower off-diagonal triangles are identical), and (3) that after a single preprocessing step (double centering), each matrix is positive semi-definite (strictly non-negative eigenvalues) just as in standard PCA and related techniques (CITE).

For simplicity let us assume the use of correlation matrices (which are covariance matrices of normalized column vectors). First, each $\mathbf{R}_{[k]}$ must be converted to a cross-product matrix through double centering. We denote a centering matrix as $\mathbf{\Xi} = \mathbf{I} - \mathbf{1}(I^{-1})\mathbf{1}^T$. We then double center each $\mathbf{R}_{[k]}$ to convert them to cross-product matrices as

$$\mathbf{S}_{[k]} = \frac{1}{2}\mathbf{\Xi}\mathbf{R}_{[k]}\mathbf{\Xi}. \tag{1}$$

Next, each $I \times I$ $\mathbf{R}_{[k]}$ matrix is normalized to account for different variance per each of the K tables. In general there are three strategies: no normalization (if the assumed variance per table is roughly equal), to normalize each table such that the sum of all squared elements equals 1, or a dividing each table by its first eigenvalue (i.e., "MFA normalization"). While there are other normalization strategies, these are the most typically used; see (CITE, CITE) for more details on normalization strategies per table in multi-table PCA-like analyses. Thus

we assume that each $\mathbf{S}_{[k]}$ is a normalized cross product matrix. MFA normalization is the preferred normalization approach.

We then must compute a similarity between all pairs of matrices in order to obtain "α weights" for each table. We first vectorize all $\mathbf{S}_{[k]}$ matrices and store each column vector in a new matrix as

$$\mathbf{Z} = [\text{vec}\{\mathbf{S}_{[1]}, \ldots, \mathbf{S}_{[\text{k}]}, \ldots, \mathbf{S}_{[\text{K}]}\}] \tag{2}$$

where $\mathbf{Z}$ is of size $I^2 \times K$ and $\mathbf{C} = \mathbf{Z}^T\mathbf{Z}$. We can obtain the α weights in one of two ways: (1) through the eigenvalue decomposition (EVD) of $\mathbf{C}$ as $\mathbf{C} = \mathbf{V}\boldsymbol{\Theta}\mathbf{V}^T$ or alternatively through the singular value decomposition (SVD) of $\mathbf{Z}$:

$$\mathbf{Z} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^T. \tag{3}$$

The $\mathbf{V}$ in both the EVD or SVD approaches are equivalent, and $\boldsymbol{\Delta}^2 = \boldsymbol{\Theta}$. We compute the α weights from the first vector of $\mathbf{V}$. To note, all elements in the first vector of $\mathbf{V}$ have the same sign and reflect overall similarity; the large the value the more similar that table is to all other tables. We compute the *alpha* weights as

$$\boldsymbol{\alpha} = \mathbf{v}_1 \times (\mathbf{v}_1^T\mathbf{1})^{-1} \tag{4}$$

Alternatively we can compute similarity between the normalized (i.e., Z-scored) column vectors of $\mathbf{Z}$, which is equivalent to computing the Rv coefficient (CITE) between each table. This Rv-similarity is the preferred approach to compute the α weights.

We then compute the compromise cross-product matrix as

$$\mathbf{S}_{[+]} = \sum_{i=1}^{K} \alpha_k \mathbf{S}_{[k]}. \tag{5}$$

Finally we decompose $\mathbf{S}_{[+]}$ with the EVD as follows

$$\mathbf{S}_{[+]} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T. \tag{6}$$

[Algorithm goes here] 1. Double center each

2. Normalize

3. Check SSPSD

4. Vectorize to obtain Z

5. Get α from svd(Z)

6. Compute S+ compromise

7. eigen(S+)

8. Compute compromise component scores

9. Compute partial component scores

Results

Discussion

88 ## References