

DISTATIS: The Analysis of Multiple Distance Matrices

Hervé Abdi*

The University of Texas at Dallas

Alice J. O’Toole

The University of Texas at Dallas

Dominique Valentin

Université de Bourgogne

Betty Edelman

The University of Texas at Dallas

Abstract

In this paper we present a generalization of classical multidimensional scaling called DISTATIS which is a new method that can be used to compare algorithms when their outputs consist of distance matrices computed on the same set of objects. The method first evaluates the similarity between algorithms using a coefficient called the R_V coefficient. From this analysis, a compromise matrix is computed which represents the best aggregate of the original matrices. In order to evaluate the differences between algorithms, the original distance matrices are then projected onto the compromise. We illustrate this method with a “toy example” in which four different “algorithms” (two computer programs and two sets of human observers) evaluate the similarity among faces.

1. Overview

Numerous face and object recognition algorithms have been developed over the last several decades. Because most of these algorithms perform very well, their evaluation requires a fine grained comparison of their performance. Such an evaluation can be implemented when different algorithms are tested on the same set of objects. In this context, the output of an algorithm is often equivalent to computing the similarity between this set of objects. For example, the current “Face Recognition Grand Challenge” ([21]) evaluates the performance of several face recognition algorithms from the distance matrices they generate on a large set of faces. The comparison of different algorithms can be performed by comparing the distance matrices they create: DISTATIS which is a new technique described in this paper can perform this comparison.

* Corresponding author: Hervé Abdi,
The University of Texas at Dallas,
Program in Cognition and Neuroscience, MS: Gr:4.1,
Richardson, TX, 75083-0688.
www.utd.edu/~herve, email: herve@utd.edu.

1.1. Origin and goal of the method

DISTATIS is a generalization of classical multidimensional scaling (MDS) whose goal is to analyze a *single* distance matrix. By contrast, the goal of DISTATIS is to analyze a *set* of distance matrices. In order to compare distance matrices, DISTATIS combines them into a common structure called a *compromise* and then projects the original distance matrices onto this compromise.

Each data set to be analyzed is called a *study* and corresponds to a distance matrix obtained on a common set of *objects*. For example, these distance matrices may correspond to measurements taken at different times. In this case, the first data set corresponds to the data collected at time $t = 1$, the second one to the data collected at time $t = 2$ and so on. The goal of the analysis, then, is to evaluate if the relative positions of the objects are stable over time. The different measurements, however, do not need to represent time. For example, the data sets can be distance matrices obtained by different methods or algorithms. The goal of the analysis, then, is to evaluate if there is an agreement between the methods.

The name DISTATIS is derived from a technique called STATIS¹ in the French speaking community where it is attributed to Escoufier [9, 10] and l’Hermier des Plantes (see [18], and also, [17, 6, 2]). It is also known as *procrustes matching by congruence coefficients* in the English speaking community (e.g., [3]). STATIS is popular in sensory evaluation (e.g., [22, 24, 5]), ecology ([11, 25]) and also recently in brain imaging ([15]).

1.2. The main idea

The general idea behind DISTATIS is first to transform each distance matrix into a cross-product matrix as it is done for standard MDS. Then, these cross-product matrices are aggregated in order to create a compromise cross-product

¹STATIS is an acronym for “*Structuration des Tableaux à Trois Indices de la Statistique*.”

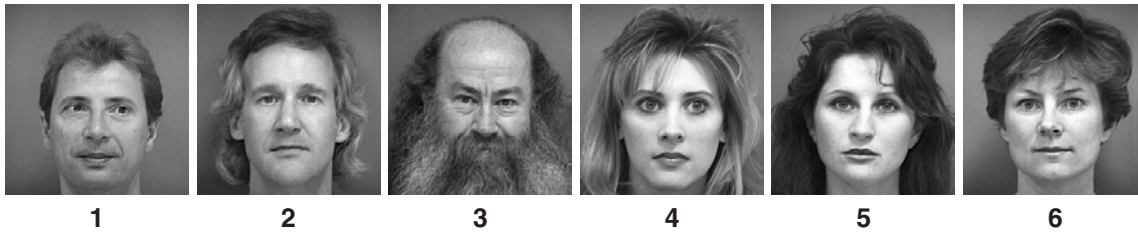


Figure 1: Six faces to be analyzed by different “algorithms.”

matrix which represents their consensus. The compromise matrix is obtained as a weighted average of the individual cross-product matrices. The weights are themselves obtained from the eigen-decomposition of a between studies similarity matrix in which the similarity between two cross-product matrices is evaluated by a coefficient called the R_V coefficient.

The PCA of the compromise gives the position of the objects in the compromise space, and therefore the Euclidean distance between objects in the compromise space is the best approximation of the distance between these objects across the set of studies. The position of the objects for each study can be represented in the compromise space as supplementary² points. Finally, as a byproduct of the weight computation, the studies can be represented as points in a multidimensional space. More formally the different steps involved in DISTATIS are:

1. Transform each distance matrix (*i.e.*, each study) into a between-object cross-product matrix.
2. Analyze the structure of the cross-product matrices.
3. Derive an optimal set of weights for computing the compromise.
4. Compute the compromise as a weighted sum of the individual cross-product matrices.
5. Compute the eigen-decomposition of the compromise matrix.
6. Plot the projections of the observations in the compromise space.
7. Plot the trajectory of the observations as supplementary points in the compromise space.

2. An example

To illustrate DISTATIS we will use the set of faces displayed in Figure 1. Four different “systems” or algorithms are compared, each of them computing a distance matrix between the faces. The first system corresponds to principal component analysis (PCA), and it computes the squared Euclidean distance between faces directly from the pixel values of the images. The second system starts by taking

²Supplementary points are not used to compute the optimum space, but are projected onto this space after it has been defined.

measurements on the faces (see Figure 2), and it computes the squared Euclidean distance between faces from these measures. The third distance matrix is obtained by first asking human observers to rate the faces on several dimensions (*i.e.*, beauty, honesty, empathy, and intelligence) and then computing the squared Euclidean distance from these measures. The fourth distance matrix is obtained from pairwise similarity ratings (on a scale from 1 to 7) collected from human observers. The average similarity rating s was transformed into a distance using the Gaussian transformation (see, *e.g.*, [1, p. 63 ff.]) as $d = \exp\{-s^2\}$.

3. Notations

The data to analyze consist of T studies (*i.e.*, each system or algorithm is a study) with each study producing an $I \times I$ distance matrix denoted $\mathbf{D}_{[t]}$, where I is the number of objects (*e.g.*, faces) and t denotes the study. The data set can be visualized as an $I \times I \times T$ data cube obtained by the concatenation of T distance matrices of order $I \times I$.

Here, we have $T = 4$ studies and $I = 6$ faces. Each study corresponds to a 6×6 distance matrix as shown below. Study 1 (Pixels):

$$\mathbf{D}_{[1]} = \begin{bmatrix} 0 & .112 & .148 & .083 & .186 & .110 \\ .112 & 0 & .152 & .098 & .158 & .134 \\ .146 & .152 & 0 & .202 & .285 & .249 \\ .083 & .098 & .202 & 0 & .131 & .110 \\ .186 & .158 & .285 & .131 & 0 & .155 \\ .110 & .134 & .249 & .110 & .155 & 0 \end{bmatrix}.$$

Study 2 (Measures):

$$\mathbf{D}_{[2]} = \begin{bmatrix} 0 & 0.60 & 1.98 & 0.42 & 0.14 & 0.58 \\ 0.60 & 0 & 2.10 & 0.78 & 0.42 & 1.34 \\ 1.98 & 2.10 & 0 & 2.02 & 1.72 & 2.06 \\ 0.42 & 0.78 & 2.02 & 0 & 0.50 & 0.88 \\ 0.14 & 0.42 & 1.72 & 0.50 & 0 & 0.30 \\ 0.58 & 1.34 & 2.06 & 0.88 & 0.30 & 0 \end{bmatrix}.$$

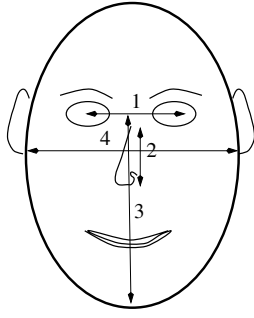


Figure 2: The measures taken on a face.

Study 3 (Ratings):

$$\mathbf{D}_{[3]} = \begin{bmatrix} 0 & 0.54 & 1.39 & 5.78 & 10.28 & 6.77 \\ 0.54 & 0 & 1.06 & 3.80 & 6.83 & 4.71 \\ 1.39 & 1.06 & 0 & 8.01 & 11.03 & 5.72 \\ 5.78 & 3.80 & 8.01 & 0 & 2.58 & 6.09 \\ 10.28 & 6.83 & 11.03 & 2.58 & 0 & 3.53 \\ 6.77 & 4.71 & 5.72 & 6.09 & 3.53 & 0 \end{bmatrix}.$$

Study 4 (Pairwise):

$$\mathbf{D}_{[4]} = \begin{bmatrix} 0 & .014 & .159 & .004 & .001 & .002 \\ .014 & 0 & .018 & .053 & .024 & .004 \\ .159 & .018 & 0 & .271 & .067 & .053 \\ .004 & .053 & .271 & 0 & .001 & .008 \\ .001 & .024 & .067 & .001 & 0 & .007 \\ .002 & .004 & .053 & .008 & .007 & 0 \end{bmatrix}.$$

3.1. Cross-product matrices

We want to compare and analyze distance matrices, and the most versatile tool for analyzing symmetric matrices is the eigen-decomposition. But distance matrices cannot be analyzed directly with the eigen-decomposition and need to be transformed into an equivalent but more convenient form. This transformation corresponds to MDS and transforms a distance matrix into a cross-product matrix³ (in MDS, the eigen-decomposition of this cross-product matrix gives the coordinates of the objects on the dimensions).

Specifically, we start with an $I \times I$ distance matrix \mathbf{D} , and with an $I \times 1$ vector of mass (whose elements are all positive or zero and whose sum is equal to 1) denoted \mathbf{m} and such that

$$\mathbf{m}^T \mathbf{1} = 1. \quad (1)$$

When—as is the case in our example—we want to give the same importance to each of the objects, we give them the

³A cross matrix can be obtained from the product of a matrix by its transpose. Covariance and correlation are well known examples.

same mass and therefore $m_i = \frac{1}{I}$. We then define the centering matrix which is equal to

$$\mathbf{\Xi} = \mathbf{I} - \frac{1}{I} \mathbf{1} \mathbf{1}^T, \quad (2)$$

and the cross-product matrix denoted by $\tilde{\mathbf{S}}$ is obtained as

$$\tilde{\mathbf{S}} = -\frac{1}{2} \mathbf{\Xi} \mathbf{D} \mathbf{\Xi}^T. \quad (3)$$

This cross-product matrix contains the same information as the distance matrix because the original distance matrix can be recovered as:

$$\mathbf{D} = \tilde{\mathbf{s}} \mathbf{1}^T + \mathbf{1} \tilde{\mathbf{s}}^T - 2 \tilde{\mathbf{S}}, \quad (4)$$

where $\tilde{\mathbf{s}}$ is a vector containing the diagonal elements of $\tilde{\mathbf{S}}$ (see, e.g., [13] for a proof and more details).

For example, the first distance matrix is transformed into the following cross-product matrix:

$$\tilde{\mathbf{S}}_{[1]} = -\frac{1}{2} \mathbf{\Xi} \mathbf{D}_{[1]} \mathbf{\Xi}^T = \begin{bmatrix} 0.042 & -0.013 & 0.002 & -0.001 & -0.028 & -0.003 \\ -0.013 & 0.045 & 0.000 & -0.007 & -0.012 & -0.013 \\ 0.002 & 0.000 & 0.108 & -0.027 & -0.044 & -0.039 \\ -0.001 & -0.007 & -0.027 & 0.040 & -0.001 & -0.004 \\ -0.028 & -0.012 & -0.044 & -0.001 & 0.088 & -0.002 \\ -0.003 & -0.013 & -0.039 & -0.004 & -0.002 & 0.062 \end{bmatrix}.$$

Because $\mathbf{D}_{[1]}$ is the squared Euclidean distance between faces, the eigen-decomposition of $\tilde{\mathbf{S}}_{[1]}$, is equivalent to a centered PCA on the face pixel matrices.

In our example, the cross product matrix for each study is expressed in units which are very different from the other studies. In order to compare the studies, we need to normalize these cross-product matrices. There are several possible normalizations; here we normalize the cross-product matrices by dividing each matrix by its first eigenvalue (an idea akin to multiple factor analysis, cf. [8, 2]). The first eigenvalue of $\tilde{\mathbf{S}}_{[1]}$ is equal to $\lambda_1 = .16$, and $\tilde{\mathbf{S}}_{[1]}$ is transformed into a normalized cross-product matrix denoted $\mathbf{S}_{[1]}$ as:

$$\mathbf{S}_{[1]} = \lambda_1^{-1} \times \tilde{\mathbf{S}}_{[1]} \quad (5)$$

$$= \begin{bmatrix} .261 & -.079 & .013 & -.003 & -.174 & -.018 \\ -.079 & .280 & .002 & -.042 & -.077 & -.084 \\ .013 & .002 & .675 & -.168 & -.276 & -.246 \\ -.003 & -.042 & -.168 & .249 & -.009 & -.026 \\ -.174 & -.077 & -.276 & -.009 & .552 & -.015 \\ -.017 & -.084 & -.246 & -.026 & -.015 & .388 \end{bmatrix}.$$

3.2. Vectorized matrices

It is convenient to transform the cross-product matrices into column vectors denoted by $\mathbf{s}_{[t]}$ where $\mathbf{s}_{[t]}$ is an $I^2 \times 1$ vector. This is done by using the vec operation

$$\mathbf{s}_{[t]} = \text{vec} \{ \mathbf{S}_{[t]} \}. \quad (6)$$

For example, the concatenation of the columns cross-product matrix $\mathbf{S}_{[1]}$ gives the 36×1 vector $\mathbf{s}_{[1]}$:

$$\mathbf{s}_{[1]} = \text{vec} \{ \mathbf{S}_{[1]} \} = \begin{bmatrix} .261 \\ -.079 \\ \vdots \\ -.015 \\ .388 \end{bmatrix}.$$

Finally, the complete set of T $\mathbf{s}_{[t]}$ vectors is concatenated into an $I^2 \times T$ matrix called the *complete data matrix* denoted \mathbf{X} . Each column of \mathbf{X} is a $\mathbf{s}_{[t]}$ vector

$$\mathbf{X} = [\mathbf{s}_{[1]}, \dots, \mathbf{s}_{[t]}, \dots, \mathbf{s}_{[T]}]. \quad (7)$$

The four normalized cross-product vectors describing the four algorithms studied are represented by the following 36×4 matrix

$$\mathbf{X} = \begin{bmatrix} 0.261 & 0.124 & 0.220 & 0.070 \\ -0.079 & 0.005 & 1.116 & -0.010 \\ \vdots & \vdots & \vdots & \vdots \\ -0.015 & 0.067 & 0.130 & -0.053 \\ 0.388 & 0.289 & 0.258 & -0.043 \end{bmatrix}. \quad (8)$$

4. Computing the compromise matrix

The *compromise matrix* is a cross-product matrix that gives the best compromise (hence its name) of the cross-product matrices representing each study. It is obtained as a weighted average of these matrices. The first step is to derive an optimal set of weights. As usual, there are several ways of defining “optimal.” Here, by optimal we mean that the studies agreeing the most with the other studies should contribute the most to the compromise. To find a set of weights that fits this definition, we first compute a matrix that expresses the similarity between studies, and next we analyze this matrix *via* its eigen-decomposition. The first eigen-vector obtained from this decomposition provides this set of weights. In addition to giving an optimum set of weights, the eigen-decomposition is also a PCA of the studies, and can be used to analyze their relationships.

4.1. Comparing the studies:

The R_V Coefficient

So, in order to obtain the optimum set of weights for computing the compromise, and to analyze the structure of the relationship between studies, we need to compute a between studies similarity matrix. This similarity should reflect the similarity of the positions of the objects between the studies. Among several possible candidates (see, *e.g.*, [3]), we will look at two coefficients derived from the scalar

product between two matrices. The first coefficient is simply the scalar product between these two matrices. It is computed by first multiplying each element of one matrix by each element of the other matrix and then summing all these products. The second coefficient is derived from the first but it is normalized in order to take values between -1 and $+1$.

Let us start with the between studies scalar product matrix. This matrix, denoted \mathbf{A} , is a $T \times T$ matrix whose term $a_{t,t'}$ gives the cross-product between matrices $\mathbf{S}_{[t]}$ and $\mathbf{S}_{[t']}$. This matrix is computed as:

$$\mathbf{A} = \mathbf{X}^T \mathbf{X} = [a_{t,t'}] = [\mathbf{S}_{[t]}^T \mathbf{S}_{[t']}] . \quad (9)$$

An equivalent definition, more convenient for mathematical analysis but computationally very inefficient, is to compute $a_{t,t'}$ as

$$[a_{t,t'}] = \text{trace} \{ \mathbf{S}_{[t]}^T \mathbf{S}_{[t']} \} . \quad (10)$$

This coefficient is a good measure of the similarity between studies only when they use the same (or comparable) units.

In order to evaluate the similarity of matrices measured with different units, we need to *normalize* the scalar product, this transforms it into a *cosine*. A cosine varies between -1 and $+1$ and is a unit-less number. The $T \times T$ cosine matrix is denoted \mathbf{C} and its elements are denoted $c_{t,t'}$. Its formula is easier to compute if we first denote by \mathbf{N} the diagonal matrix whose terms are the diagonal terms of \mathbf{A} (*i.e.*, $\mathbf{N} = \text{diag} \{ \text{diag} \{ \mathbf{A} \} \}$)⁴. The cosine matrix is then computed as

$$\mathbf{C} = \mathbf{N}^{-\frac{1}{2}} \mathbf{A} \mathbf{N}^{-\frac{1}{2}} = \mathbf{N}^{-\frac{1}{2}} \mathbf{X}^T \mathbf{X} \mathbf{N}^{-\frac{1}{2}} \quad (11)$$

$$= [c_{t,t'}] = \left[\frac{\mathbf{S}_{[t]}^T \mathbf{S}_{[t']}}{\|\mathbf{S}_{[t]}\| \|\mathbf{S}_{[t']}\|} \right] = [\cos(\mathbf{s}_{[t]}, \mathbf{s}_{[t']})] .$$

When all the $\mathbf{s}_{[t]}$ vectors have a norm⁵ equal to 1, the matrices \mathbf{A} and \mathbf{C} are identical (because the denominator of the fraction in Equation 11 is equal to one). The elements of the cosine matrix have been given several names in the literature: R_V , or R_V coefficients⁶ ([23, 17]) *congruence*

⁴The operation *diag* returns a vector of the diagonal elements when its argument is a matrix. When its argument is a vector, the operation *diag* returns a diagonal matrix with the elements of the vector as the diagonal elements. Therefore, when applied twice to a square matrix, the operation *diag* gives a diagonal matrix whose diagonal elements are the diagonal elements of the original matrix.

⁵Incidentally, the normalization described by Equation 5 normalizes the matrix such that its largest eigenvalue is equal to 1. In general, this normalization does not imply that the norm of the vectorized matrix is equal to 1.

⁶ R is the standard notation for a correlation coefficient. It is used here because this coefficient is similar to a correlation coefficient (which is a cosine between *centered* vectors). It is sometimes called *vector correlation coefficient*. This is a potentially misleading appellation because the R_V coefficients is *not*, strictly speaking, a correlation coefficient (because it is not centered).

coefficients ([3], p. 350), or *monotonicity* coefficients ([3], p. 203, see also [4, 7, 16, 26, 27, 28]). In general, in the STATIS framework, the appellation R_V is the most popular one. An equivalent definition of the R_V coefficient is more convenient for analyzing its mathematical properties but is computationally very inefficient. With this definition, the R_V coefficient between $\mathbf{S}_{[t]}$ and $\mathbf{S}_{[t']}$ is computed as

$$R_V = [c_{t,t'}] = \frac{\text{trace} \left\{ \mathbf{S}_{[t]}^T \mathbf{S}_{[t']} \right\}}{\sqrt{\text{trace} \left\{ \mathbf{S}_{[t]}^T \mathbf{S}_{[t]} \right\} \times \text{trace} \left\{ \mathbf{S}_{[t']}^T \mathbf{S}_{[t']} \right\}}} . \quad (12)$$

The elements of \mathbf{A} and \mathbf{C} can also be considered as defining a similarity measure between the $\mathbf{S}_{[t]}$ matrices. Within this framework, the elements of \mathbf{A} or \mathbf{C} are called the *Frobenius*, or *Schur*, or *Hilbert-Schmidt* matrix scalar products (see, e.g., [14, p. 291]).

4.1.1 Back to the example

Using the R_V coefficient, we obtain the following between studies cosine matrix

$$\mathbf{C} = \begin{bmatrix} 1.00 & .77 & .76 & .40 \\ .77 & 1.00 & .41 & .53 \\ .76 & .41 & 1.00 & .30 \\ .40 & .53 & .30 & 1.00 \end{bmatrix} . \quad (13)$$

How to interpret this matrix? As we have already mentioned, a cosine takes on values between -1 and $+1$. A value of $+1$ for the cosine between two studies means that the position of the objects in these two studies is the same up to a rotation or a change of scale. A value of -1 means that the order of the similarities between the objects is exactly reversed for the 2 studies. A value of 0 means that the two studies are unrelated (*i.e.*, they are *orthogonal* to each other). In general, when using DISTATIS, the cosines are positive (because the different studies tend to agree on what they measure on the objects). This property has some interesting consequences described later on (see Section 4.2).

Looking at matrix \mathbf{C} , we can see that, on the whole, the two image-based algorithms are relatively similar to each other (the cosines are relatively large), and that the direct similarity evaluation obtained from human observers differs from the other algorithms.

4.2. Finding the weights: PCA of the cosine matrix

The cosine matrix expresses the similarity structure of the studies. Because it is computed by multiplying a matrix by its transpose (*e.g.*, $\mathbf{X}^T \mathbf{X}$), the cosine matrix is *positive semi-definite*⁷. Therefore, its eigen-decomposition

⁷A positive semi-definite matrix has positive or zeros eigenvalues.

gives the *non centered* PCA of the complete data matrix. Formally, the cosine data matrix has the following eigen-decomposition

$$\mathbf{C} = \mathbf{P} \mathbf{\Theta} \mathbf{P}^T \text{ with } \mathbf{P}^T \mathbf{P} = \mathbf{I} , \quad (14)$$

where \mathbf{P} is the matrix of eigenvectors and $\mathbf{\Theta}$ is the diagonal matrix of the eigenvalues⁸ of \mathbf{C} . For our example, the eigenvectors and eigenvalues of \mathbf{C} are:

$$\mathbf{P} = \begin{bmatrix} .58 & .28 & -.21 & .74 \\ .53 & -.24 & -.64 & -.50 \\ .48 & .56 & .51 & -.44 \\ .40 & -.74 & .53 & .11 \end{bmatrix} \text{ and } \text{diag} \{ \mathbf{\Theta} \} = \begin{bmatrix} 2.62 \\ 0.80 \\ 0.49 \\ 0.09 \end{bmatrix} .$$

An element of a given eigenvector represents the projection of one study on this eigenvector. Thus the T studies can be represented as points in the eigenspace and their similarities analyzed visually. This step corresponds to a PCA of the between-studies space. In general, when we plot the studies in their factor space, we want to give to each component the length corresponding to its eigenvalue (*i.e.*, the inertia of the coordinates of a dimension is equal to the eigenvalue of this dimension, which is the standard procedure in PCA and MDS). For our example, we obtain the following coordinates:

$$\mathbf{G} = \mathbf{P} \times \mathbf{\Theta}^{\frac{1}{2}} = \begin{bmatrix} .93 & .25 & -.14 & .23 \\ .85 & -.22 & -.45 & -.15 \\ .78 & .50 & .36 & -.13 \\ .65 & -.66 & .37 & .03 \end{bmatrix} .$$

As an illustration, Figure 3 displays the projections of the four algorithms onto the first and second eigenvectors of the cosine matrix. We can see on this figure that the four studies are positively correlated with the first eigenvector. This is due to the fact that the elements of the cosine matrix are all positive. Specifically, when the elements of a positive semi-definite matrix are all positive, its first eigenvector has all its elements with the same sign⁹.

The eigenvalues of the cosine matrix give the inertia explained by the eigenvectors. By dividing each eigenvalue of the cosine matrix by the sum of all its eigenvalues, we obtain the *proportion* of the total inertia explained by each eigenvector. For example, here we find that 66% of variance between the algorithms is explained by the first eigenvector, 20% by the second eigenvector, and 12% by the third one.

The first eigenvector explains a large part of the variance between the studies because the matrix is not centered. In this case the first eigenvector represents what is common to the different studies. The more similar a study is to the other studies, the more it will contribute to this eigenvector.

⁸Why not using the usual $\mathbf{\Lambda}$ for the eigenvalues? Because we need this notation later on!

⁹This is the *Perron-Frobenius* theorem ([20, p. 219]).

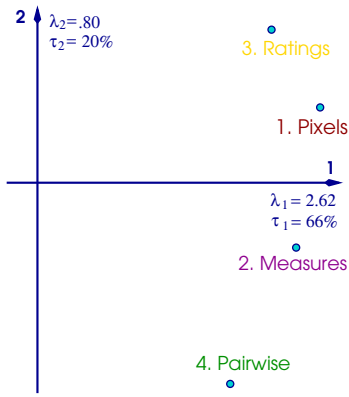


Figure 3: Plot of the between-studies space (*i.e.*, eigen-analysis of the matrix \mathbf{C}).

Or, in other words, studies with larger projections on the first eigenvector are more similar to the other studies than studies with smaller projections. Thus the elements of the first eigenvector give the optimal weights to compute the compromise matrix¹⁰.

4.3. Computing the compromise

The compromise matrix is obtained as a weighted *average* of the study cross-product matrices. The weights are derived from the first column of matrix \mathbf{P} , denoted \mathbf{p}_1 , which is the first eigenvector of the data cosine matrix. Remember that a weighted average is obtained when the sum of the weights is equal to one. Because the sum of the elements of the first eigenvector is *not*, in general, equal to one¹¹, we need to re-scale them to fit this constraint. So the weights are obtained by dividing each element of \mathbf{p}_1 by the sum of the elements of \mathbf{p}_1 . The vector containing these weights is denoted α , and is computed as:

$$\alpha = (\mathbf{1}^T \mathbf{p}_1)^{-1} \times \mathbf{p}_1, \quad (15)$$

[the term $(\mathbf{1}^T \mathbf{p}_1)^{-1}$ just means “take the inverse of the sum of the elements of \mathbf{p}_1 ”]. For our example, we obtain:

$$\alpha = (\mathbf{1}^T \mathbf{p}_1)^{-1} \times \mathbf{p}_1 = [.29 \quad .27 \quad .24 \quad .20]^T. \quad (16)$$

With α_t denoting the weight for the t -th study, the compromise matrix, denoted $\mathbf{S}_{[+]}$, is computed as:

$$\mathbf{S}_{[+]} = \sum_t \alpha_t \mathbf{S}_{[t]}. \quad (17)$$

¹⁰Strictly speaking, this is true only when the elements of the first eigenvector have the same sign.

¹¹The eigenvectors are in general normalized such that the sum of the squared elements of the eigenvectors is equal to one.

In our example, this gives:

$$\mathbf{S}_{[+]} = \begin{bmatrix} .176 & .004 & -.058 & .014 & -.100 & -.036 \\ .004 & .178 & .022 & -.038 & -.068 & -.010 \\ -.058 & .022 & .579 & -.243 & -.186 & -.115 \\ .014 & -.038 & -.243 & .240 & .054 & -.027 \\ -.100 & -.068 & -.186 & .054 & .266 & .034 \\ -.036 & -.010 & -.115 & -.027 & .034 & .243 \end{bmatrix}$$

4.4. How representative is the compromise?

The compromise is the best least-squared aggregate of the original cross-product matrices. But how *good* is this best? To evaluate the quality of the compromise, we need an index of quality. This is given by the first eigenvalue of matrix \mathbf{C} which is denoted ϑ_1 . An alternative index of quality (easier to interpret) is the ratio of the first eigenvalue of \mathbf{C} to the sum of its eigenvalues:

$$\text{Quality of compromise} = \frac{\vartheta_1}{\sum_{\ell} \vartheta_{\ell}} = \frac{\vartheta_1}{\text{trace}\{\Theta\}}. \quad (18)$$

Here the quality of the compromise is evaluated as:

$$\text{Quality of compromise} = \frac{\vartheta_1}{\text{trace}\{\Theta\}} = \frac{2.62}{4} \approx .66. \quad (19)$$

So we can say that the compromise “explains” 66% of the inertia of the original set of data tables. This is a relatively small value and this indicates that the algorithms differ substantially on the information they capture about the faces.

5. Analyzing the compromise

The compromise matrix is itself a cross-product matrix, and therefore its eigen-decomposition amounts to a PCA. From this analysis, we can explore the structure of the set of observations. The eigen-decomposition of the compromise is:

$$\mathbf{S}_{[+]} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T \quad (20)$$

with, in our example:

$$\mathbf{V} = \begin{bmatrix} .017 & .474 & -.451 & -.107 & -.627 \\ .121 & .400 & .256 & .726 & .258 \\ .823 & -.213 & .114 & -.308 & .053 \\ -.388 & .309 & .159 & -.566 & .492 \\ -.348 & -.443 & .549 & .043 & -.462 \\ -.192 & -.527 & -.626 & .211 & .287 \end{bmatrix} \quad (21)$$

and

$$\text{diag}\{\mathbf{\Lambda}\} = [.80 \quad .35 \quad .26 \quad .16 \quad .11]^T. \quad (22)$$

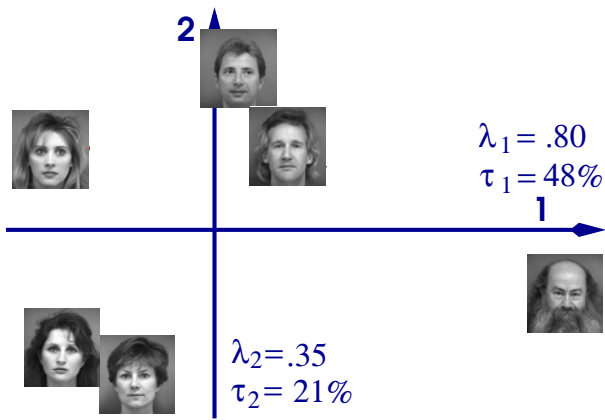


Figure 4: Analysis of the compromise: Plot of the faces in the plane defined by the first two principal components of the compromise matrix.

From Equations 21 and 22 we can compute the compromise factor scores for the faces as:

$$\mathbf{F} = \mathbf{V}\mathbf{\Lambda}^{\frac{1}{2}} \quad (23)$$

$$= \begin{bmatrix} -.015 & .280 & -.228 & -.043 & -.209 \\ .108 & .236 & .129 & .294 & .086 \\ .738 & -.126 & .058 & -.125 & .018 \\ -.348 & .182 & .080 & -.229 & .164 \\ -.312 & -.262 & .277 & .018 & -.155 \\ -.172 & -.311 & -.316 & .086 & .096 \end{bmatrix}.$$

In the \mathbf{F} matrix, each row represents a object (*i.e.*, a face) and each column a component. Figure 4 displays the faces in the space defined by the first two principal components. The first component has an eigenvalue equal to $\lambda_1 = .80$, such a value explains 48% of the inertia of the cross-product matrix. The first component is easily interpreted as the opposition of the male to the female faces (with Face # 3 appearing extremely masculine). This component indicates that the algorithms agree that, on the whole, gender explains most of the differences between faces. The second component, with an eigenvalue of .35, explains 21% of the inertia. The second dimension is more difficult to interpret and seems linked to hair color (*i.e.*, light hair versus dark or no hair).

6. Projecting the studies into the compromise space

Each algorithm provided a cross-product matrix, which was used to create the compromise cross-product matrix. The analysis of the compromise reveals the structure of the face space common to the algorithms. In addition to this common space, we want also to see how each algorithm

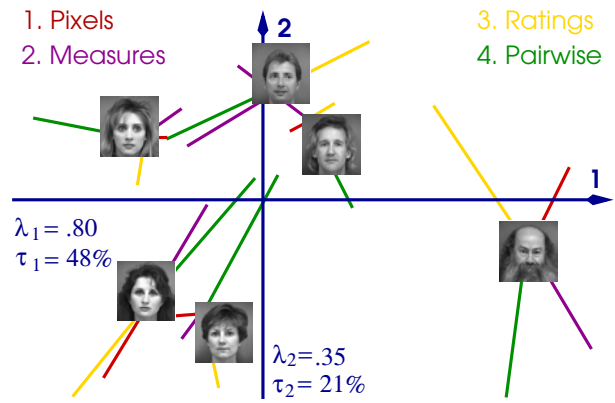


Figure 5: The Compromise: Projection of the algorithm matrices onto the compromise space Algorithm 1 is in red, 2 is in magenta, 3 is in yellow, and 4 is in green.

“interprets” or distorts this space. This can be achieved by projecting the cross-product matrix of each algorithm onto the common space. The mechanics of this projection can be deduced from Equations 20 and 23. Specifically, from

$$\mathbf{S}_{[+]} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \text{ and } \mathbf{F} = \mathbf{V}\mathbf{\Lambda}^{\frac{1}{2}}, \quad (24)$$

and from

$$\mathbf{V}^T\mathbf{V} = \mathbf{I} \text{ (because the columns of } \mathbf{V} \text{ are normalized),}$$

we get that

$$\mathbf{F} = \mathbf{S}_{[+]} \mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}}. \quad (25)$$

Therefore, the matrix $(\mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}})$ transforms the cross-product matrix into its projections¹² onto the compromise space. We can also use this matrix to project the cross-product matrix of each algorithm into the common space. For example, the coordinates of the projections into the compromise space of the first cross-product matrix are computed as:

$$\mathbf{F}_{[1]} = \mathbf{S}_{[1]} (\mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}}) \quad (26)$$

$$= \begin{bmatrix} .07 & .30 & -.44 & -.24 & -.33 \\ .11 & .24 & .22 & .53 & .34 \\ .85 & .11 & .09 & -.44 & .01 \\ -.26 & .19 & .04 & -.31 & .30 \\ -.47 & -.50 & .67 & .18 & -.57 \\ -.30 & -.33 & -.59 & .28 & .25 \end{bmatrix}. \quad (27)$$

Figure 5 shows the first two principal components of the compromise space along with the projections of each of the algorithms. The position of a face in the compromise is

¹²Formally, we say that $(\mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}})$ is a projection operator.

the barycenter of its positions for the four algorithms. In order to facilitate the interpretation, we have drawn lines linking the position of each face for each of the four algorithms to its compromise position. This picture confirms that the algorithms differ substantially. It shows also that some faces are more sensitive to the differences between algorithms (e.g., compare Faces 3 and 4). Specifically, in order to quantify the the sensitivity of faces to the algorithms, we compute the inertia of the projections relative to the compromise position. In our example we obtain the following values: $[\text{.04 } \text{.24 } \text{.79 } \text{.01 } \text{.45 } \text{.63}]$.

6.1. Specific projections are not orthogonal

A word of caution: The projections of the faces on the eigenvectors of the compromise are orthogonal (this is an inherent property of the eigenvectors), but the projections of the same faces for each algorithm are *not* orthogonal. This means, in particular that the coordinates computed for each algorithm cannot be obtained by a standard orthogonal rotation of the compromise.

7. Conclusion

DISTATIS is a very versatile and efficient method that can be used with relatively large data sets. It can give insights both into the performance of the algorithms that we want to analyze, and into the objects that are analyzed by these algorithms. As a statistical technique, DISTATIS is part of the Procrustes analysis family. The relationship between these techniques is explored in more details in [19, 29, 12].

References

- [1] Abdi, H., Valentin, D., Edelman, B. (1999). *Neural networks*. Thousand Oaks: Sage.
- [2] Abdi, H. (2003). Multivariate analysis. In M. Lewis-Beck, A. Bryman, & T. Futing (Eds): *Encyclopedia for research methods for the social sciences*. Thousand Oaks: Sage.
- [3] Borg, I., Groenen, P., (1997). *Modern multidimensional scaling*. New York: Springer Verlag.
- [4] Brokken F.B. (1983). Orthogonal Procrustes rotation maximizing congruence. *Psychometrika*, 48, 343–349
- [5] Chaya, C., Perez-Hugalde, C., Judez, L., Wee, C.S., Guinard, J.X. (2003). Use of the STATIS method to analyze time-intensity profiling data. *Food Quality and Preference*, 15, 3–12.
- [6] Dazy, F., le Barzic, J. F., Saporta, G., Lavallard F. (1996). *L'analyse des données évolutives*. Paris: Technip.
- [7] Cliff, N. (1968). The “idealized individual” interpretation of individual differences in multidimensional scaling. *Psychometrika*, 33, 225–232.
- [8] Escoufier B., Pagès, J. (1998) *Analyses factorielles simples et multiples*. Paris: Dunod.
- [9] Escoufier, Y. (1980). L'analyse conjointe de plusieurs matrices de données. In Jolivet, M. (Ed.) *Biométrie et Temps*. Paris: Société Française de Biométrie. pp. 59–76.
- [10] Escoufier, Y. (1985). Objectifs et procédures de L'analyse conjointe de plusieurs tableaux. *Statistiques et Analyse des Données*, 10, 1–10.
- [11] Gaertner J.C., Chessel, D., Bertrand D. (1998). Stability of spatial structures of demersal assemblages: a multitable approach. *Aquatic Living Resources*, 11, 11–85.
- [12] Gower, J.C., Dijksterhuis, G.B. (2004). *Procrustes problems*. Oxford: OUP.
- [13] Greenacre, M. (1984). *Correspondence analysis*. London: Academic Press.
- [14] Horn, R.A., Johnson, C.R. (1985) *Matrix analysis*. Cambridge: Cambridge University Press.
- [15] Kherif, F., Poline, J.P., Mériaux, S., Benali, H., Flandin, G., Brett M., (2003) Group analysis in functional neuroimaging: selecting subjects using similarity measures. *NeuroImage*, 20, 2197–2208.
- [16] Korth, B., Tucker, L.R., (1976). Procrustes matching by congruence coefficients. *Psychometrika*, 41, 531–535.
- [17] Lavit, C. (1988). *Analyse conjointe de tableaux quantitatifs*. Paris: Masson.
- [18] l'Hermier des Plantes H. (1976). *Structuration des tableaux à trois indices de la statistique*. Thèse de troisième cycle. Université de Montpellier.
- [19] Meyners, M., Kunert, J., Qannari, E.M. (2000). Comparing generalized procrustes analysis and STATIS. *Food Quality and Preference*, 11, 77–83.
- [20] Ortega, J. (1987). *Matrix theory: A second course*. New York: Plenum Press.
- [21] Phillips, P.J., Flynn, P.J., Scruggs, T., Bowyer, K.W., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W. (in press). Overview of the face recognition grand challenge. *IEEE Conference on Computer Vision and Pattern Recognition*.
- [22] Qannari, E.M., Wakeling, I., MacFie, H.J.H. (1995). A hierarchy of models for analyzing sensory data. *Food Quality and Preference*, 6, 309–314.
- [23] Robert, P., Escoufier, Y. (1976) A unifying tool for linear multivariate statistical methods: the RV-coefficient. *Applied Statistics*, 25, 257–265.
- [24] Schlich, P. (1996). Defining and validating assessor compromises about product distances and attribute correlations. In T. Næs, & E. Risvik (Eds.), *Multivariate analysis of data in sensory sciences*. New York: Elsevier.
- [25] Stanimirova, I, Walczaka, B., Massarta, D.L., Simeonov, V., Saby, C.A., di Crescenzo E. (2004) STATIS, a three-way method for data analysis. Application to environmental data. *Chemometrics and Intelligent Laboratory Systems*, 73, 219–233
- [26] Tucker, L.R., Messick, S. (1963). An individual difference model for multidimensional scaling *Psychometrika*, 28, 333–367.
- [27] Tucker, L.R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31, 279–311
- [28] Tucker, L.R. (1972). Relations between multidimensional scaling and three-mode factor analysis. *Psychometrika*, 37, 3–27.
- [29] Vivien, M. (2002). *Approches PLS linéaires et non-linéaires pour la modélisation de multi-tableaux*. Unpublished doctoral dissertation, Université de Montpellier, France. (available from <http://www.myrtille.org/>).