DISTATIS
How to analyze multiple distance matrices

Hervé Abdi[1] & Dominique Valentin

1 Overview

1.1 Origin and goal of the method

DISTATIS is a generalization of classical multidimensional scaling (MDS see the corresponding entry for more details on this method) proposed by Abdi *et al.,* (2005). Its goal is to analyze several distance matrices computed on the same set of objects. The name DISTATIS is derived from a technique called STATIS whose goal is to analyze multiple data sets (see the corresponding entry for more details on this method, and Abdi, 2003). DISTATIS first evaluates the similarity between distance matrices. From this analysis, a compromise matrix is computed which represents the best aggregate of the original matrices. The original distance matrices are then projected onto the compromise.

[1]In: Neil Salkind (Ed.) (2007). *Encyclopedia of Measurement and Statistics.* Thousand Oaks (CA): Sage.
Address correspondence to: Hervé Abdi
Program in Cognition and Neurosciences, MS: Gr.4.1,
The University of Texas at Dallas,
Richardson, TX 75083–0688, USA
E-mail: herve@utdallas.edu http://www.utd.edu/~herve

1.2 When to use it

The data sets to analyze are distance matrices obtained on the same set of objects. These distance matrices may correspond to measurements taken at different times. In this case, the first matrix corresponds to the distances collected at time $t = 1$, the second one to the distances 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 matrices, however, do not need to represent time. For example, the distance matrices can be derived from different methods. The goal of the analysis, then, is to evaluate if there is an agreement between the methods.

1.3 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 a standard MDS. Then, these cross-product matrices are aggregated to create a compromise cross-product matrix which represents their consensus. The compromise matrix is obtained as a weighted average of individual cross-product matrices. The PCA of the compromise gives the position of the objects in the compromise space. The position of the object for each study can be represented in the compromise space as supplementary points. Finally, as a byproduct of the weight computation, the studies can be represented as points in a multidimensional 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), it computes the squared Euclidean distance between faces directly from the pixel values of the images. The second system starts by taking measurements on the faces (see Figure 2), and 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 Shepard's transformation: $d = \exp\{-s^2\}$.

3 Notations

The raw data consist of T data sets and we will refer to each data set as a *study*. Each study is an $I \times I$ distance matrix denoted $\mathbf{D}_{[t]}$, where I is the number of objects and t denotes the study.

Here, we have $T = 4$ studies. 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):

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

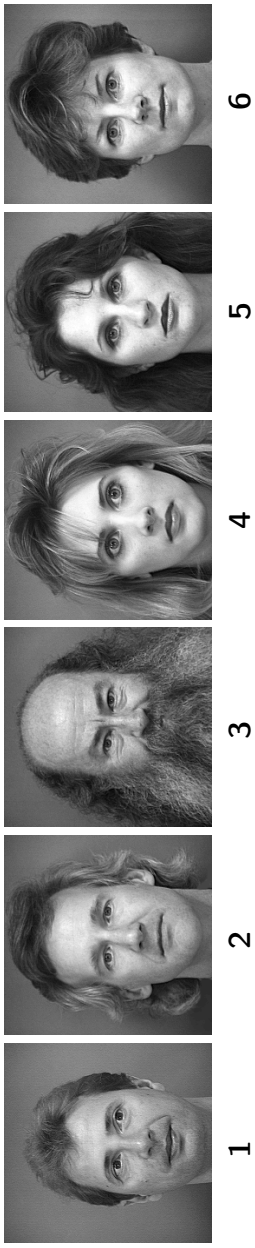

Figure 1: Six faces to be analyzed by different "algorithms."

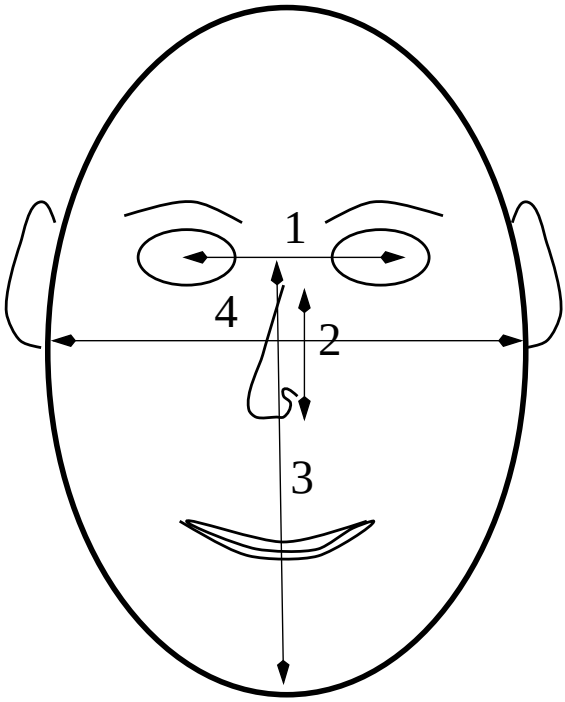

Figure 2: The measures taken on a face.

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):

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

Distance matrices cannot be analyzed directly and need to be transformed. This step corresponds to MDS (see the MDS entry for more details) and transforms a distance matrix into a cross-product matrix

We start with an $I \times I$ distance matrix $\mathbf{D}$, 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

$$\underset{1\times}{\mathbf{m}^T}\ \underset{I\times 1}{\mathbf{1}} = 1\ . \tag{1}$$

If all observations have the same mass (as in here) $m_i = \frac{1}{I}$. We then define the centering matrix which is equal to

$$\underset{I\times I}{\mathbf{\Xi}} = \underset{I\times I}{\mathbf{I}} - \underset{I\times 1\times I}{\mathbf{1}\ \mathbf{m}^T}\ , \tag{2}$$

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

$$\widetilde{\mathbf{S}} = -\tfrac{1}{2}\mathbf{\Xi}\mathbf{D}\mathbf{\Xi}^T\ . \tag{3}$$

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

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

In order to compare the studies, we need to normalize the cross-product matrices representing them. 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.* Escofier & Pagès 1998, see also entry in this encyclopedia). The first eigenvalue of matrix $\widetilde{\mathbf{S}}_{[1]}$ is equal to $\lambda_1 = .16$, and matrix $\widetilde{\mathbf{S}}_{[1]}$ is transformed into a normalized cross-

product matrix denoted $\mathbf{S}_{[1]}$ as:

$$\mathbf{S}_{[1]} = \lambda_1^{-1} \times \widetilde{\mathbf{S}}_{[1]} \tag{4}$$

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

4 Computing the compromise matrix

The *compromise matrix* is a cross-product matrix that gives the best compromise of the studies. It is obtained as a weighted average of the study cross-product matrices. The weights are chosen so that studies agreeing the most with other studies will have the larger weights. To find these weights we need to analyze the relationships between the studies.

The *compromise matrix* is a cross-product matrix that gives the best compromise 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. The principle to find this set of weight is similar to that describe for STATIS and involves the following steps

4.1 Comparing the studies

To analyze the similarity structure of the studies we start by creating a *between study cosine matrix* denoted $\mathbf{C}$. This is a $T \times T$ matrix whose generic term $c_{t,t'}$ gives the cosine between studies t and t'. This cosine, also known as the R_V-coefficient (see the corresponding entry for more details on this coefficient), is defined as

$$R_V = \left[c_{t,t'}\right] = \frac{\mathsf{trace}\left\{\mathbf{S}_{[t]}^{\mathsf{T}}\mathbf{S}_{[t']}\right\}}{\sqrt{\mathsf{trace}\left\{\mathbf{S}_{[t]}^{\mathsf{T}}\mathbf{S}_{[t]}\right\} \times \mathsf{trace}\left\{\mathbf{S}_{[t']}^{\mathsf{T}}\mathbf{S}_{[t']}\right\}}}. \tag{5}$$

Using this formula we get the following matrix **C**:

$$\mathbf{C} = \begin{bmatrix} 1.00 & .77 & .76 & .40 \\ .77 & 1.00 & .41 & .53 \\ .76 & .41 & 1.00 & .30 \\ .40 & .53 & .30 & 1.00 \end{bmatrix}. \tag{6}$$

4.2 PCA of the cosine matrix

The cosine matrix has the following eigendecomposition

$$\mathbf{C} = \mathbf{P}\boldsymbol{\Theta}\mathbf{P}^{\mathsf{T}} \text{ with } \mathbf{P}^{\mathsf{T}}\mathbf{P} = \mathbf{I}\,, \tag{7}$$

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

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

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:

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

As an illustration, Figure 3 on the following page displays the projections of the four algorithms onto the first and second eigenvectors of the cosine matrix.

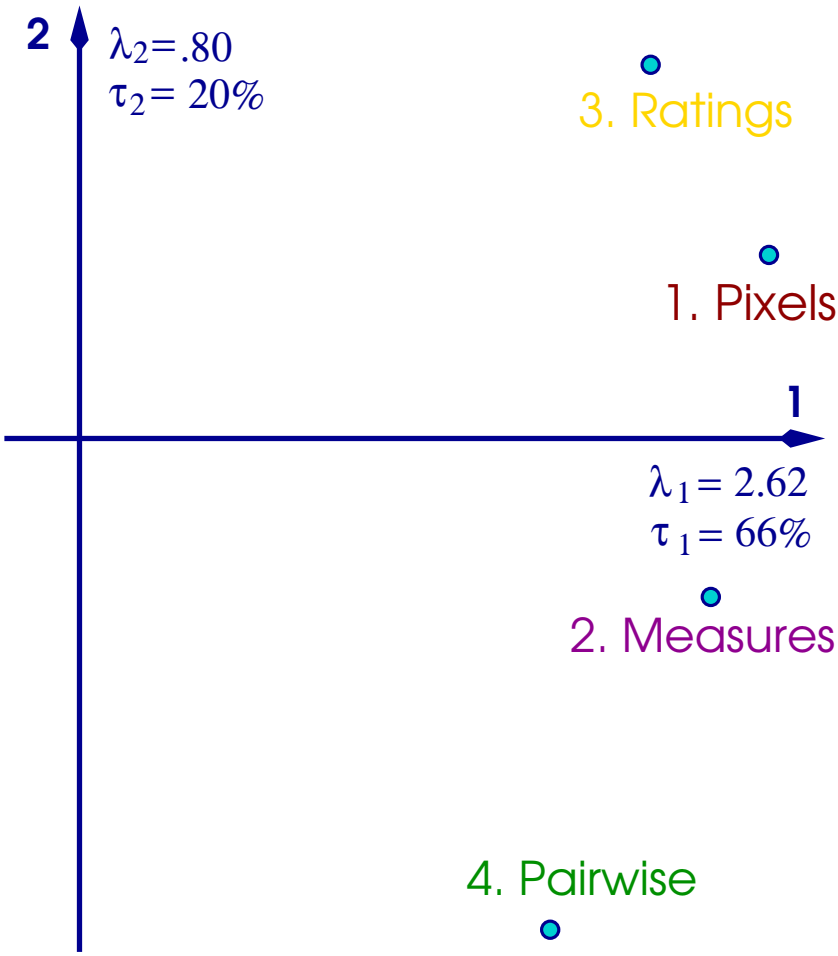

Figure 3: Plot of the between-studies space (*i.e.*, eigen-analysis of the matrix $\mathbf{C}$).

Because the matrix is not centered 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. 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 matrix.

4.3 Computing the compromise

As for STATIS the weights are obtained by dividing each element of $\mathbf{p}_1$ by their sum. The vector containing these weights is denoted $\boldsymbol{\alpha}$, for our example, we obtain:

$$\boldsymbol{\alpha} = \begin{bmatrix} .29 & .27 & .24 & .20 \end{bmatrix}^{\mathsf{T}} . \tag{8}$$

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

$$\mathbf{S}_{[+]} = \sum_{t}^{T} \alpha_t \mathbf{S}_{[t]} . \tag{9}$$

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?

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:

$$\text{Quality of compromise} = \frac{\vartheta_1}{\sum_{\ell} \vartheta_\ell} = \frac{\vartheta_1}{\text{trace}\{\boldsymbol{\Theta}\}} . \tag{10}$$

Here the quality of the compromise is evaluated as:

$$\text{Quality of compromise} = \frac{\vartheta_1}{\text{trace}\{\boldsymbol{\Theta}\}} = \frac{2.62}{4} \approx .66 . \tag{11}$$

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.

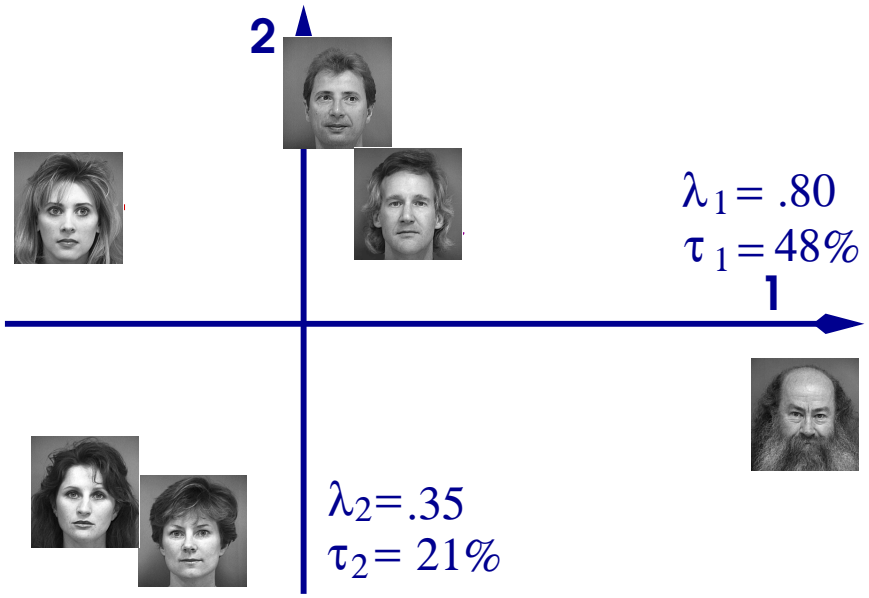

Figure 4: Analysis of the compromise: Plot of the faces in the plane defined by the first two principal components of the compromise matrix.

5 Analyzing the compromise

The eigen-decomposition of the compromise is:

$$\mathbf{S}_{[+]} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \tag{12}$$

with, in our example:

$$\mathbf{Q} = \begin{bmatrix} .017 & .474 & -.451 & -.107 & -.627 \\ .121 & .400 & .256 & .726 & .258 \\ .823 & -.213 & .114 & -.308 & .053 \\ -.388 & .309 & .159 & -.566 & .492 \\ -.348 & -.443 & .549 & .043 & -.462 \\ -.192 & -.527 & -.626 & .211 & .287 \end{bmatrix} \tag{13}$$

and

$$\text{diag}\{\mathbf{\Lambda}\} = \begin{bmatrix} .80 & .35 & .26 & .16 & .11 \end{bmatrix}^T. \tag{14}$$

From Equations 13 and 14 we can compute the compromise factor scores for the faces as:

$$\mathbf{F} = \mathbf{Q}\boldsymbol{\Lambda}^{\frac{1}{2}} \tag{15}$$

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

In the **F** matrix, each row represents an objects (*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: The second component, with an eigenvalue of .35, explains 21% of the inertia. The first component is easily interpreted as the opposition of the male to the female faces (with Face # 3 appearing extremely masculine). 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 "interprets" or distorts this space. This can be achieved by projecting the cross-product matrix of each algorithm onto the common space. This operation is performed by computing a projection matrix which transforms the scalar product matrix into loadings. The projection matrix is deduced from the combination of Equations 12 and 15 which gives

$$\mathbf{F} = \mathbf{S}_{[+]}\mathbf{Q}\boldsymbol{\Lambda}^{-\frac{1}{2}}. \tag{16}$$

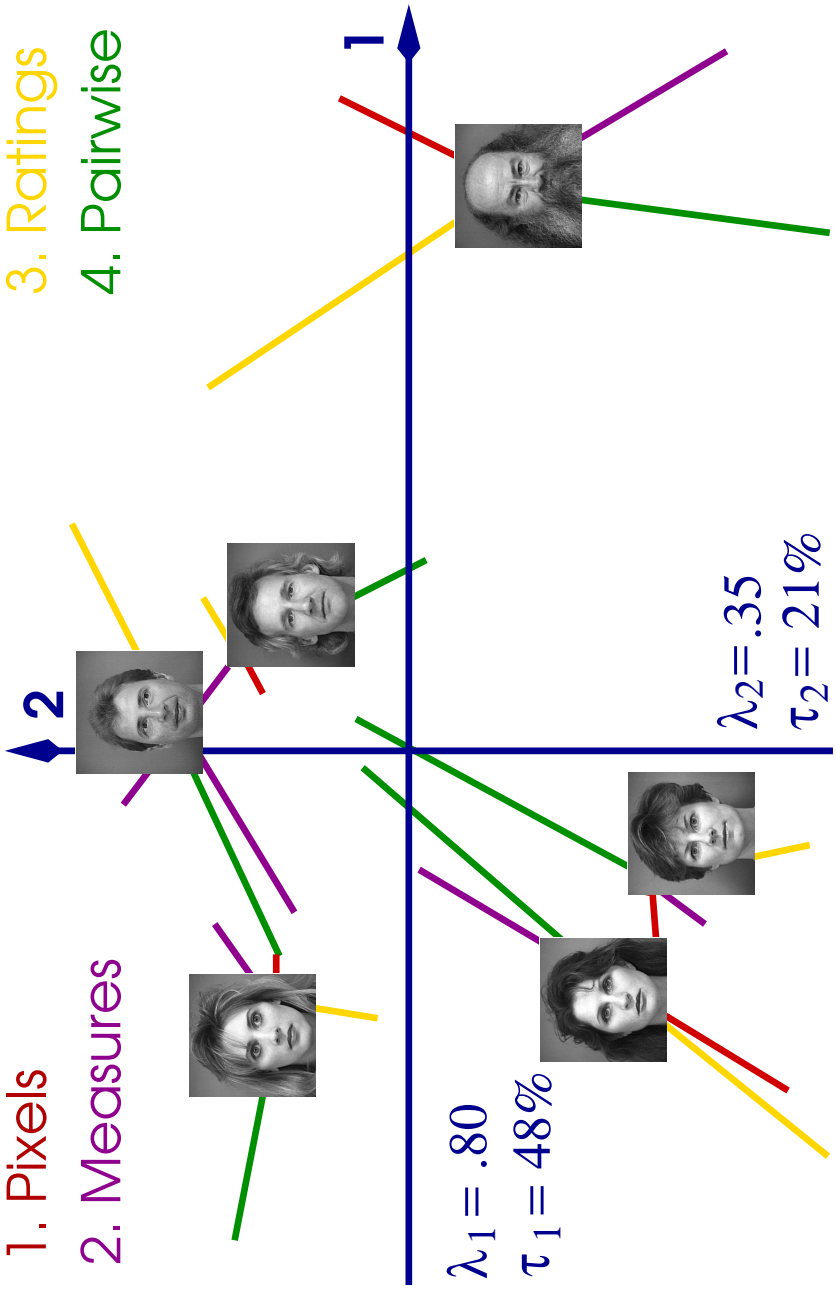

Figure 5: The Compromise: Projection of the algorithm matrices onto the compromise space Algorithm 1 is in red, 2 is in magenta, 3 is in yellow, and 4 is in green.

This shows that the projection matrix is equal to $\left(\mathbf{Q}\boldsymbol{\Lambda}^{-\frac{1}{2}}\right)$. It is used to project the scalar product matrix of each study onto the common space. For example, the coordinates of the projections for the first study are obtained by first computing the matrix

$$\mathbf{Q}\boldsymbol{\Lambda}^{-\frac{1}{2}} = \begin{bmatrix} -0.02 & 0.80 & -0.89 & -0.26 & -1.88 \\ 0.13 & 0.68 & 0.51 & 1.79 & 0.77 \\ 0.92 & -0.36 & 0.23 & -0.76 & 0.16 \\ -0.43 & 0.52 & 0.31 & -1.40 & 1.48 \\ -0.39 & -0.75 & 1.09 & 0.11 & -1.39 \\ -0.21 & -0.89 & -1.24 & 0.52 & 0.86 \end{bmatrix}, \tag{17}$$

and then using this matrix to obtain the coordinates of the projection as:

$$\mathbf{F}_{[1]} = \mathbf{S}_{[1]}\left(\mathbf{Q}\boldsymbol{\Lambda}^{-\frac{1}{2}}\right) \tag{18}$$

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

The same procedure is used to compute the matrices of the projections on to the compromise space for the other algorithms.

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 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).

References

[1] 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.

[2] Abdi, H., Valentin, D., O'Toole, A.J., Edelman, B. (2005). DISTATIS: The analysis of multiple distance matrices. *Proceedings of the IEEE Computer Society: International Conference on Computer Vision and Pattern Recognition*. (San Diego, CA, USA). pp. 42-47.

[3] Escofier B., Pagès, J. (1998) *Analyses factorielles simples et multiples*. Paris: Dunod.